Ministère de l'Enseignement Supérieur et de la recherche Scientifique

BADJI MOKHTAR-ANNABA UNIVERSITY
NIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار – عنابة

Faculté des sciences de l'ingénieur
Département d'informatique

THESE

Présentée en vue d'obtention du diplôme de ***DOCTORAT en Informatique***

Sécurisation évolutionnaire du transfert d'images

Option
Informatique

Par
SOUICI Ismahane

Directeur de Thèse
SERIDI Hamid Pr. Univ. 08 mai 1945 Guelma

Devant le jury

Président :	**KHADIR Tarek**	Pr. Univ. Badji Mokhtar Annaba
Examinateurs :		
	FARAH Nadir	Pr. Univ. Badji Mokhtar Annaba
	MEROUANI Hayet	MC. Univ. Badji Mokhtar Annaba
	KHOLADI Khiredine	Pr. Univ. Mentouri Constantine
	CHIKHI Salim	Pr. Univ. Mentouri Constantine

Année universitaire : 2012/2013

Remerciements

En premier lieu, je tiens à exprimer ma profonde gratitude à Monsieur SERIDI Hamid, Professeur à l'université 08 mai 1945 de Guelma, pour avoir dirigé ma recherche depuis le Magister et pour la confiance et l'intérêt qu'il m'a témoignés tout au long de l'élaboration de cette thèse. Je lui suis très reconnaissante pour le suivi régulier et formateur reçu durant ces quatre années de Doctorat, et bien avant, durant les trois années de Magister. Je le remercie pour la démarche scientifique rigoureuse et l'esprit d'auto-critique qu'il a su m'inculquer. J'espère avoir été à la hauteur de ces espérances.

J'adresse mes remerciements les plus sincères à Monsieur KHADIR Tarek, Professeur à l'université Badji Mokhtar d'Annaba, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Mes vifs remerciements vont aux membres du jury qui ont accepté de prendre de leur temps pour examiner mon travail : Monsieur FARAH Nadir, Professeur à l'université Badji Mokhtar d'Annaba, Madame MEROUANI Hayet, Maître de Conférences à l'université Badji Mokhtar d'Annaba, Monsieur KHOLLADI Khiredine, Professeur à l'université Mentouri de Constantine et Monsieur CHIKHI Salim, Professeur à l'Université Mentouri de Constantine.

Au cours de ces quatre années de recherche, j'ai eu à effectuer un stage de courte de durée en 2010, au laboratoire CReSTIC de l'université de Reims Champagne Ardenne, France, sous la direction de Monsieur AKDAG Herman, Professeur à l'université de Reims Champagne Ardenne, France. Qu'il trouve ici, l'expression de mes remerciements les plus sincères, pour son chaleureux accueil. Je tiens à remercier, aussi, Monsieur Cyril DE RUNZ, Maitre de Conférences à l'université de Reims Champagne Ardenne, France pour sa disponibilité, ainsi que pour ses discussions fructueuses que j'ai eues avec lui.

Merci à celles et ceux qui auront à lire tout ou une partie de ce manuscrit et qui y trouveront un intérêt quelconque.

Enfin, un grand MERCI à ma famille et, en particulier, mes parents pour le soutien et les encouragements qu'ils ont su m'apporter pour arriver au terme de cette thèse.
J'espère les honorer avec ce travail.

الملخص

في يومنا هذا أصبحت شبكات المعلوماتية معقدة و التنصتات غير القانونية عليها قائمة. لهذا أصبح من الضروري حماية المعلومات الحساسة المنقولة عبر هذه الشبكات, أين يعد علم التشفير واحد من الحلول الجد فعالة, الذي و على الرغم من كل التطورات التي سجلت بهذا المجال ماتزال بعض النقائص مسجلة خاصة ما يتعلق بحجم المفتاح المستعمل و مقاومة الهجمات المتطورة.

في هذا السياق ينصب موضوع بحثنا للدكتوراه حيث نسعى إلى تطوير خوارزمات تشفير باستعمال métaheuristiques التي تعد من الطرق التي توفر حلولا تقريبية و تفتح مجالات تصميمية لطرق حل مهمة لمشاكل التحسين.
هذه الأخيرة تنقسم إلى قسمين : metaheuristiques التي تستعمل مجموعة حلول métaheuristiques التي تستعمل حلا واحدا. وبما أن métaheuristiques توظف طابع العشوائية في معظم مراحل عملها و الذي يعد عنصرا مهم إستعماله في مجال التشفير لتعقيد مهمة المهاجمين, لهذا وقع إختيارنا على هذه الطرق لحل مشكل تشفير المعلومات النصية و الصور.

إذن وضمن الصنف الأول الذي يشمل métaheuristiques التي تستعمل مجموعة حلول قمنا بتطوير ثلاث خوارزمات تعتمد الطريقة التطورية المستوحاة من نظرية "دارون" و هذا حسب نوعين مختلفين من التشفير (التشفير المعتمد على الوضعيات و التشفير المعتمد على الظهور), بالإضافة إلى خوارزم رابع و الذي يستعمل métaheuristique مستوحاة من comportement الحقيقي للنمل.
بالنسبة للصنف الثاني والذي يضم métaheuristique التي تستعمل حلا واحدا, توصلنا إلى تكييف طريقة البحث الطابوه إستعمالها في حل المشكل لمدروس و هكذا قمنا بتطويرخوارزم تشفير طابوه خامس.

أخيرا قمنا بتجريب ومقارنة كل الخوارزمات المقترحة فيما بينها وبالنسبة لعدد من مراجع التشفير حيث أظهروا خصائص جيدة سواءا فيما يتعلق بدرجة الغموض أو مقاومة الهجمات الأكثر تطورا. قطة أخرى جد مهمة توصلنا إليها عبر عملنا هذا هي استحداث خوارزمات non déterministes , نقطة تساهم بدرجة كبيرة في زيادة قوة مثل هذه الخوارزمات.

Résumé

Aujourd'hui, les réseaux informatiques sont complexes et les écoutes illégales possibles. Il se pose donc un réel problème quant à la sécurité lors de la transmission de données. Pour des raisons éthiques, le transfert des données délicates ne peut se faire avec un tel risque et doit donc se protéger. La protection la plus adaptée pour ce type de communication réside dans la cryptographie. Cependant et malgré toutes ses évolutions et ses mises en œuvre, elle est toujours entravée par quelques défauts citant en particulier la taille des clés et la résistance contre les attaques avancées.

Ainsi, notre travail de thèse porte sur le développement de nouveaux algorithmes cryptographiques par exploitation des métaheuristiques constituant une partie importante des méthodes approchées et ouvrant des voies très intéressante en matière de conception de méthodes heuristiques pour l'optimisation. Ces dernières se scindent en deux catégories : les métaheuristiques à population de solutions et les métaheuristiques à une seule solution ou encore dites à trajectoire. Et vu que les métaheuristiques exploitent un aspect pseudo-parallèle durant différentes étapes de leurs schémas opératoires, chose qui est très intéressante à exploiter dans le domaine cryptographique pour compliquer de façon considérable la tâche des cryptanalystes, donc notre choix s'est porté sur l'utilisation de telles méthodes pour résoudre le problème de cryptage de données texte et images.

En effet, et pour la première catégorie de métaheuristiques à population, nous avons développé trois algorithmes de chiffrement exploitant les principes évolutionnaires inspirés de la théorie darwinienne, mais opérants suivant deux modes de chiffrement différents (chiffrement à base de positions et un chiffrement à base d'occurrences) ; et un quatrième algorithme utilisant une métaheuristique inspirée du comportement réel des fourmis (ACO).

Pour la deuxième catégorie qui est celle de métaheuristiques à trajectoire, nous avant arrivé à adapter la méthode de recherche taboue et l'exploiter pour résoudre notre problème étudié et ainsi développer un cinquième algorithme de cryptage tabou.

Les algorithmes proposés ont été évalués et comparés entre eux et avec les principaux algorithmes standards de cryptage où ils ont montré de bonnes caractéristiques relatives soit à leur degré confusionnel ou à leur résistibilité aux attaques les plus avancées. L'autre point crucial résultant de nos travaux de thèse était l'innovation d'algorithmes de chiffrement non déterministes, chose qui augmente considérablement leur robustesse.

Mots clés : Cryptage, attaques avancées, métaheuristiques, algorithmes évolutionnaires, recherche tabou, algorithme de colonies de fourmis.

Abstract

Today, computer networks are complex and the illegal wiretapping is possible. This raises a real problem of security when transmitting data. For ethical reasons, the transfer of delicate data can not be done with such a risk and must protect themselves. Protection best suited for this type of communication lies in cryptography. However, despite all its developments and implementations, it is still hampered by some shortcomings citing in particular the key size and resistance against advanced attacks.

Thus, the object of our thesis is the development of new cryptographic algorithms by exploitation of metaheuristics constituting a significant portion of the approximate methods and opening very interesting channels in the design of heuristic methods for optimization. The latter fall into two categories: metaheuristics of solutions population and metaheuristics of one solution or also tell trajectory metaheuristics. And since metaheuristics exploit a pseudo-parallel aspect during different stages of their operating patterns, something that is very interesting to use in the cryptographic domain to significantly complicate the task of the cryptanalysts, so our choice fell on the use of such methods to solve the problem of the encryption of text and images data.

Indeed, for the first class of metaheuristics of population, we developed three encryption algorithms exploiting the evolutionary principles inspired by the Darwinian theory but operating in two different encryption modes (positions based encryption and occurrences based encryption) and a fourth algorithm using a metaheuristic inspired by the ants real behavior (ACO).

For the second category which is trajectory metaheuristics, we arrived to adapt the method of tabu search and use it to solve the problem at hand. Thus, a fifth taboo encryption algorithm was developed.

The proposed algorithms were evaluated and compared among themselves and with principal encryption standard algorithms where they showed good characteristics for either their confusional degree or their resistibility against the most advanced attacks. The other crucial point arising from our thesis work was the innovation of non-deterministic encryption algorithms, something that greatly increases their robustness.

Table des matières

Chapitre III : CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

Chapitre IV : CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

Chapitre V : CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Dès que les hommes ont appris à communiquer, ils ont trouvé des moyens d'assurer la confidentialité d'une partie de leurs communications : l'origine de la cryptographie remonte sans doute aux origines de l'homme. En effet, le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire de les rendre inintelligibles sans une action spécifique.

Du bâton nommé « scytale » au Vie siècle avant JC, en passant par le carré de Polybe ou encore le code de César, on assista au développement plus ou moins ingénieux de techniques de chiffrement expérimentales dont la sécurité reposait essentiellement dans la confiance que leur accordaient leurs utilisateurs. Après la Première Guerre mondiale a lieu une première révolution technologique.

Mais ce n'est qu'à l'avènement de l'informatique et d'Internet que la cryptographie prend tout son sens. Les efforts conjoints d'IBM et de la NSA conduisent à l'élaboration du DES (Data Encryption Standard), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XXe siècle. À l'ère d'Internet, le nombre d'applications civiles de chiffrement (banques, télécommunications, cartes bleues…) explose. Le besoin d'apporter une sécurité accrue dans les transactions électroniques fait naître les notions de signature et authentification électroniques. La première technique de chiffrement à clef publique sûre (intimement liée à ces notions) apparaît : le RSA.

Donc, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information. C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à ramener et à modéliser le problème de cryptage comme un problème d'optimisation.

En effet, l'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solutions algorithmiques efficaces valables pour toutes les données.

Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles que les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens.

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques*. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, les chromosomes et les mécanismes d'intensification et de diversification pour la métaheuristique des algorithmes évolutionnaires), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Les métaheuristiques sont représentées essentiellement par les *méthodes de voisinage* comme le recuit simulé et la recherche tabou, et les *algorithmes évolutifs* comme les algorithmes génétiques et les stratégies d'évolution. Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi et hormis cette introduction et la conclusion générale qui reprennent les travaux de l'ensemble des chapitres et quelques perspectives majeures pour la poursuite de ce travail, le manuscrit est divisé en deux grandes parties.

Un état de l'art aussi complet que possible relatif soit au problème étudié qui est celui de cryptage, soit aux approches de résolutions exploitées qui sont les métaheuristiques, fera l'objet des deux premiers chapitres formant la première partie. En effet, le premier chapitre traite la cryptographie depuis sa première apparition jusqu'à nos jours en présentant un bref aperçu historique permettant de comprendre la distinction entre les trois disciplines (cryptologie, cryptographie et cryptanalyse), les fonctionnalités de base de la cryptographie (confidentialité, authentification, intégrité et la non-répudiation) et leurs différentes catégories (la cryptographie symétrique, asymétrique, hybride et quantique) tout en citant les fameux algorithmes appartenant aux principales catégories.
De son tour, le deuxième chapitre présente une introduction aux métaheuristiques tout en citant les concepts de base ainsi qu'une classification des métaheuristiques illustrée d'exemples d'algorithmes de chacune des classes.

Dans la deuxième partie de notre recherche, nous proposons un nouvel axe de recherche que représente l'application des métaheuristiques pour résoudre le problème de cryptage de données textes et images. Ainsi, les deux catégories de métaheuristiques ont été exploitées : métaheuristiques à population et les métaheuristiques à trajectoire.
Dans le cadre de la première, nous avons choisi d'exploiter deux métaheuristiques qui sont les algorithmes évolutionnaires (AEs) et les algorithmes de colonies de fourmis. Le principal avantage de ces méthodes heuristiques vient de leur capacité à traiter le problème de cryptage en ne possédant qu'un minimum d'informations sur celui-ci. Chose qui est primordiale dans ce problème pour augmenter la confusion des algorithmes développés.

Ainsi, le troisième chapitre résume l'application d'AEs où les différentes étapes partant du codage sont explicitées. En effet, pour un problème à résoudre, il est nécessaire d'adapter la représentation des individus aux objectifs recherchés. Cette adaptation permet de faire converger l'AE plus ou moins rapidement et de tenir compte naturellement des contraintes de la modélisation du problème. Dans le cas présent, l'espace de recherche que nous cherchons à optimiser est l'ensemble représenté par les solutions possibles du problème de cryptage et par une fonction d'évaluation de chaque solution.

Le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, les algorithmes ont été groupés en deux catégories distinctes suivant les modes de chiffrement utilisé implémentant différents codages : chiffrement à base de position ou chiffrement à base d'occurrences.

D'une manière globale, nous allons définir un individu de la population comme une donnée chiffrée possible. Cet individu doit pouvoir être évalué numériquement par la fonction d'évaluation proposée. Cette fonction de fitness calcule la qualité d'une donnée chiffrée en fonction du besoin de confusion. Nous montrerons aussi l'application de notre méthode sur quelques exemples d'images de différentes tailles. Une interprétation et une discussion des résultats obtenus seront abordées pour pouvoir, par la suite, comparer les nouveaux algorithmes proposés de cryptage évolutionnaire où l'algorithme opérant suivant le mode de chiffrement par occurrences a montré une efficacité en termes de temps de calcul, de pouvoir de confusion, de résistibilité aux attaques et de longueur de clé meilleure que celles des algorithmes opérants suivant le mode de chiffrement par positions.

Cette conclusion a été démontrée par l'application d'une deuxième méthode de résolution exploitant les algorithmes de colonies de fourmis illustrée sur un quatrième chapitre où les résultats obtenus été très satisfaisants.

Au niveau du cinquième chapitre, un autre algorithme de cryptage a été proposé s'inscrivant cette fois-ci, sous la deuxième catégorie de métaheuristiques dans le but d'explorer l'espace de recherche par exploitation de la notion de voisinage. Il s'agit d'un algorithme utilisant le principe d'une recherche tabou pour pouvoir choisir la configuration représentant la meilleure solution au problème de cryptage de données texte ou images.

Dans chacun des chapitres décrivant nos algorithmes de cryptage proposés, nous présentons des exemples et résultats expérimentaux sur des données tests. Nous avons aussi effectué des bilans en faisant ressortir les avantages et faiblesses des méthodes développées.

CHAPITRE I

CRYPTOGRAPHIE

I.1. Introduction

Depuis les temps historiques les plus reculés, l'homme a perçu le besoin de cacher, de dissimuler ou faire mystère des informations personnelles ou confidentielles, et cela bien avant l'ère informatique afin de les rendre inintelligibles à des lecteurs indésirables. C'est pourquoi, les codes ont existé.

Les origines de la cryptographie semblent remonter à plus de 4000 ans en Egypte. Plusieurs indications archéologiques tendent à montrer que les « écritures secrètes » sont en fait anciennes que l'invention de l'écriture elle-même. Polybius développa un système de codage des lettres de l'alphabet consistant à remplacer chaque lettre de l'alphabet par deux nombres, donnant la ligne et la colonne où se trouve cette lettre dans une matrice. Jules César utilisait une simple méthode de substitution de lettres pour communiquer secrètement avec ses généraux : c'est un chiffre par décalage,…etc.

C'est cependant au cours de la seconde guerre mondiale que la cryptographie s'inscrit véritablement comme élément central des stratégies militaires. Le cas le plus connu est certainement l'histoire entourant le décodage du code Enigma par les Polonais et les Britanniques. Une conjonction d'espionnage classique et d'efforts de mathématiciens polonais permet de déduire la clé utilisée et ainsi de décoder les messages encodés avec Enigma. On trouve apparemment bien moins de détails sur les efforts cryptographiques durant la guerre froide, probablement parce que ces informations sont encore « Top Secret ».

On en est maintenant à l'époque moderne où le champ d'application de la cryptologie s'est élargi et a trouvé un regain d'actualité avec toutes les applications nouvelles suscitées par l'utilisation de l'Internet. La révolution d'Internet et l'utilisation de plus en plus d'informations massives sous forme numérique facilitent les communications et rendent de ce fait plus fragiles les informations que l'on détient, c'est pourquoi il devient nécessaire de protéger le contenu de certains messages des inévitables curieux. En effet, les réseaux ouverts créent des brèches de sécurité et il est plus aisé à un adversaire d'accéder aux informations. Dans une communication à distance, des questions cruciales se posent et leurs réponses s'imposent ; comment être sur :

- que l'on parle à la bonne personne (authenticité),
- que nos propos ne sont pas altérés (intégrité),
- que la conversation n'est pas espionnée (confidentialité),
- de l'identité de l'émetteur (non répudiation) ?

Indéniablement, avec l'essor fulgurant des nouvelles technologies, le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements.

Alors que la cryptographie consiste à sécuriser les données, tandis que, la cryptanalyse est l'étude des informations cryptées afin d'en découvrir le secret. Grâce à la cryptanalyse, les militaires ont pu mener leurs guerres en découvrant les correspondances de leurs ennemis et en contrôlant les réseaux de communications mais d'autre part si elle est utilisée par une personne mal intentionnée il peut générer des dégâts onéreux aux entreprises ou aux sociétés.

I.2. Les Fondements de la Cryptographie

I.2.1. Terminologie

Comme toute science, la cryptographie possède son propre langage. Dans ce qui suit les mots clés de domaine cryptographique.

I.2.1.1. Cryptographie

Le terme cryptographie vient en effet des deux mots grecs : *Kruptus* qu'on peut traduire comme secret et *Graphein* pour écriture. Ainsi la cryptographie est l'art de dissimuler une information écrite en clair (plain text) en *cryptogramme* (cipher text) pour qu'elle soit incompréhensible que par son destinataire légitime par le biais d'une clé appelé « clé de chiffrement » (processus de *chiffrement*). Pour rendre l'information à nouveau intelligible par le biais d'une clé appelé « clé de déchiffrement » le processus inverse est appliqué (processus de *déchiffrement*).

On distingue généralement deux types de clefs :

- **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

Le schéma suivant illustre le processus cryptographique :

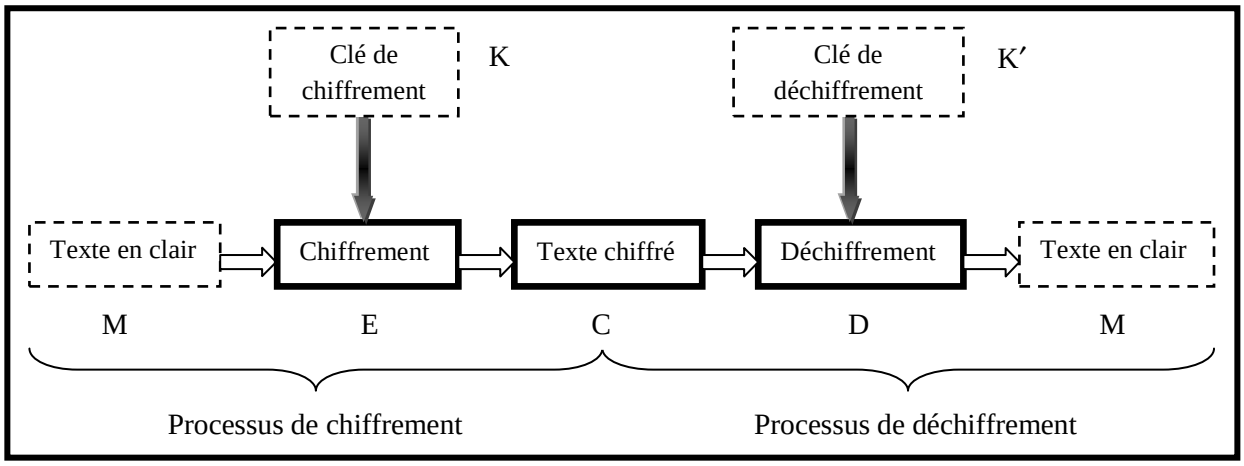


Figure I.1. *Processus cryptographique.*

I.2.1.2. Cryptanalyse

a. Définition

C'est l'art d'étude des crypto systèmes en cherchant leurs failles et leurs vulnérabilités afin de retrouver des messages clairs correspondant à des messages chiffrés sans avoir à connaître les clés utilisées dans le chiffrement. Lorsque tous les éléments de la méthode utilisée pour coder des messages sont repérés, on dit qu'on a **cassé** ou **brisé** le système cryptographique utilisé. Plus un système est difficile à briser, plus il est sûr.
La personne qui pratique la cryptanalyse est appelée : cryptanalyste. Il tente à décrypter le message chiffré pour découvrir son secret. On a distingué entre le verbe « décrypter » et « déchiffrer » puisque ce dernier est réservé pour le déchiffrement par le destinataire légitime.

b. Types de cryptanalyse

On distingue plusieurs types d'attaques :

- ***Attaque sur texte chiffré seul (ciphertext-only)*** : l'attaquant a seulement la possibilité d'intercepter un ou plusieurs messages chiffrés. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- ***Attaque à texte clair connu (known-plaintext attack*)** : se base sur la connaissance d'une partie du texte en clair pour déduire le reste du message. La tâche est de retrouver la clef utilisée pour chiffrer ce message.
- ***Attaque à texte clair choisi (chosen-plaintext attack)*** : se base sur la possibilité de choisir un texte clair et d'obtenir son chiffrement et en ayant la possibilité de générer les versions chiffrées de messages clair avec un algorithme considérer comme une **boîte noire** tel que les algorithmes à clé publique puisque l'algorithme est public.
- ***Attaque à texte chiffré choisi (chosen-ciphertext attack)*** : le cryptanalyste possède des messages chiffrés et essaye de les déchiffrer de son choix. Sa tâche est de retrouver la clef.

c. Familles d'attaques cryptanalytiques

Il existe plusieurs familles d'attaques cryptanalytiques, les plus connues sont les suivantes :

- **L'analyse fréquentielle** : examine les répétitions des lettres du message chiffré afin de trouver la clé. Cette technique est découverte par Al-Kindi au IXe siècle [www1] contre les chiffrements mono-alphabétiques. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est basée sur le fait que, dans chaque langue, certaines lettres ou combinaisons de lettres apparaissent avec une certaine fréquence.
- **L'attaque par dictionnaire** : le mot testé est pris dans une liste prédéfinis contenant les mots de passe les plus courants et aussi des variantes de ceux-ci. Ces listes sont généralement dans toutes les langues les plus utilisées, elles contiennent des mots existants ou des mots diminutifs (par exemple « powa » pour « power » ou « K7 » pour « cassette »). Elle est souvent couplée à l'attaque par force brute.
- **L'attaque par force brute** : s'appuie sur le cassage d'un mot de passe en testant tous les mots de passe possibles. C'est le seul moyen de récupérer la clé dans les algorithmes les plus modernes et encore inviolés comme AES.
- **La cryptanalyse linéaire** : c'est une attaque à texte clair inventée par le japonais Mitsuru Matsui [www1]. L'idée est de trouver des approximations linéaires entre les bits de sortie, les bits d'entrée et les bits de la clé. Si certaines de ces approximations apparaissent avec une probabilité suffisante, on a alors démontré que la correspondance

entre entrée et sortie n'est pas purement aléatoire. Le nombre de clés à envisager pour déchiffrer le message est restreint.

- **La cryptanalyse différentielle** : découverte par deux cryptologies israéliennes : Bihan et Shamir [www1]. Elle consiste à comparer les sorties de l'algorithme quand on lui met en entrée deux messages ayant une différence fixe. On étudie comme variant les sorties si les deux messages ne diffèrent que par un seul bit. Si en déplaçant ce bit à l'intérieur des messages, certains bits des sorties restent inchangés, on a alors trouvé une faille dans l'algorithme et celui-ci est attaquable.

I.2.1.3. Cryptologie

La cryptologie du grec kruptos « secret, caché » et logos « discours » qui embrasse à la fois la cryptographie et la cryptanalyse. Elle se partage entre la cryptographie, qui inclut la conception des mécanismes destinés à assurer les fonctions suivantes : confidentialité, intégrité, authentification et traçabilité (non répudiation), et la cryptanalyse dont le but est de déjouer les protections ainsi mises en place.

I.2.1.4. Fonction de hachage

Une fonction de hachage est une fonction mathématique qui à partir d'un message (d'une donnée) génère une autre chaîne, l'empreinte (généralement plus courte). Cette empreinte est très sensible au texte initial (une petite modification du texte provoque une grande modification d'empreinte).

I.2.1.5. Signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message. Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur.

I.2.1.6. Les certificats

Pour assurer l'intégrité des clés publiques, ces dernières sont publiées avec un certificat. Un certificat (ou certificat de clés publiques) est une structure de données qui est uniquement signée par une autorité certifiée.

CA, Certification Autorité, est une autorité en qui les utilisateurs peuvent faire confiance. Il contient une série de valeurs, comme le nom du certificat et son utilisation, des informations identifiants le propriétaire et la clé publique, la clé publique elle-même, la date d'expiration et le nom de l'organisme de certificats. Le CA utilise sa clé privée pour signer le certificat et assure ainsi une sécurité supplémentaire. Si le récepteur connaît la clé publique du CA, il peut vérifier que le certificat provient vraiment de l'autorité concernée et, ainsi, de s'assurer que le certificat contient des informations viables et une clé publique valide.

I.2.2. Fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur fiabilité et confidentialité au travers d'un canal peu sûr (téléphone, réseau informatique ou autre). Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions en plus de la fiabilité et la confidentialité. Il s'agit de garantir l'intégrité et l'authenticité des données échangées. Les postulats de sécurité associés à la cryptographie sont :

a. Intégrité

Vérifier l'intégrité des données consiste à déterminer si les données, ressources, traitements ou services n'ont pas été altérées durant la communication de manière fortuite ou intentionnelle.

b. Confidentialité

La confidentialité est le maintien du secret des informations. Elle consiste à rendre l'information discrète ou inintelligible à d'autres personnes que les seuls acteurs de la transaction. La confidentialité peut être vue comme la «protection des données contre une divulgation non autorisée» [Gher, 2004].

c. Authentification

L'authentification consiste à assurer l'identité d'un utilisateur c.à.d. de garantir à chacun de correspondant que son partenaire est bien celui qu'il croit être.
On distingue deux types d'authentification :

- ***Le contrôle d'accès :*** C'est l'opération permettant d'être certain de l'identité d'une personne utilisateur pour permettre l'accès à des ressources uniquement pour la personne autorisée (par exemple l'utilisation d'un mot de passe pour un disque dur).
- ***Authentification de l'origine des données :*** Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services, l'authentification et l'intégrité.

d. La non répudiation (traçabilité)

La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction ; l'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information.

I.2.3. Problèmes de la cryptographie

Quelque soit le cryptosystème utilisé pour le chiffrement, ceci reste toujours cassable un jour ou l'autre. La sécurité d'un cryptosystème repose en fait sur la complexité des algorithmes définis et sur les puissances de calcul disponibles pour une attaque. Ainsi la solution adoptée actuellement est de faire « *retarder le travail des cryptanalystes* », c.à.d. faire en sorte que la durée nécessaire pour déchiffrer un code soit supérieure à la durée de validité des données.

I.3. Algorithmes cryptographiques

Au cours d'un échange visant à communiquer de façon secrète deux protagonistes, appelé ici l'émetteur et le récepteur à travers un canal peu sûr, l'information à transmettre sera donc chiffrée par un procédé de chiffrement en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de en utilisant une clé de déchiffrement sans laquelle son procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique : *algorithme cryptographique* (ou *chiffre***)**. La sécurité des données chiffrées est entièrement dépendante de deux facteurs : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un *cryptosystème***.**

I.3.1. Le principe de Kerckhoffs

Pour briser un cryptosystème, un opposant cherche à obtenir deux éléments d'information :

1. Quel est le type de système de codage utilisé ? et,
2. Quelle est la clé d'encodage utilisée ?

Bien entendu, son travail est simplifié (mais certainement pas terminé) s'il connait le type de système utilisé. Avec le temps cette information finit par circuler. Cette hypothèse de travail est appelée le principe de Kerckhoffs. Ce principe consiste à affirmer que la sécurité d'un système de chiffrement ne devrait pas être fondée sur le secret de la procédure utilisée, mais essentiellement sur le secret de la clé.

Auguste Kerckhoffs écrit en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire », où il disait [Kerc, 1883] :

« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants:

1) le système doit être matériellement, sinon mathématiquement, indéchiffrable.
2) il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.
3) la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
4) il faut qu'il soit applicable à la correspondance télégraphique.
5) il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
6) enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer ».

Les points 2 et 3 sont les axiomes fondamentaux de la cryptographie suivant lesquels l'attaquant possède tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé spécifique pour le chiffrement. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ce secret sera découvert ou volé.

I.3.2. Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplait (P, C, K, E, D) satisfaisant les points suivants :

1) P est un ensemble fini de blocs de textes clairs possibles.
2) C est un ensemble fini de blocs de textes chiffrés possibles.
3) K est un ensemble fini de clefs possibles.
4) Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k (e_k (x)) = x$ pour tout texte clair $x \in P$.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$ qui doit être transmise préalablement loin des yeux d'Oscar (à travers un canal de communication sûr). Supposant qu'Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne : $x = x_1 x_2 \ldots x_n$ avec : $n \in Z$, $n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera : $y = y_1 y_2 \ldots y_n$.

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1 x_2 \ldots x_n$. Le procédé de communication est illustré sur la Figure I.2.

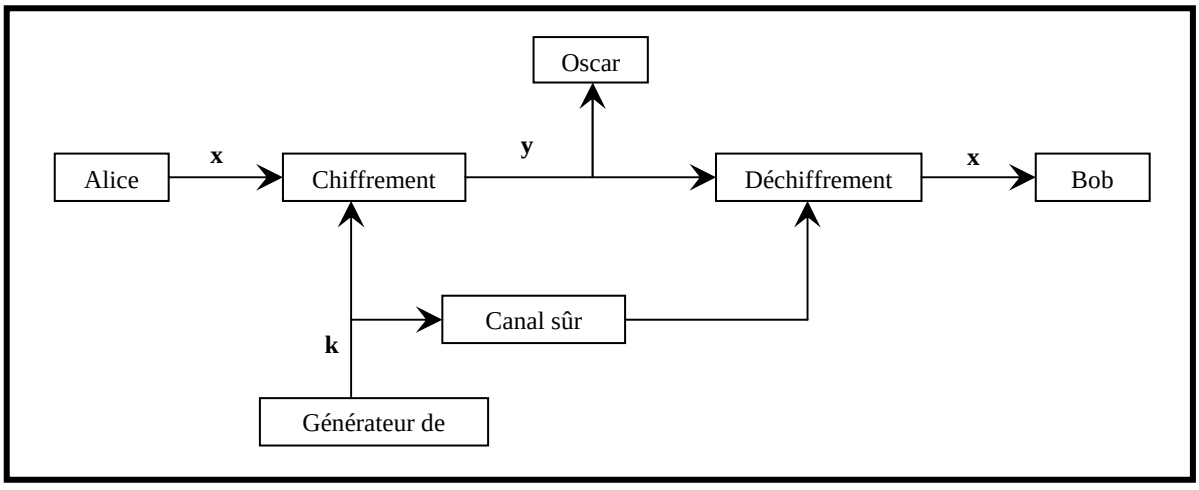


Figure I.2. *Le procédé de communication.*

I.3.3. Classes de cryptographie

Le schéma suivant illustre les différentes classes de la cryptographie :

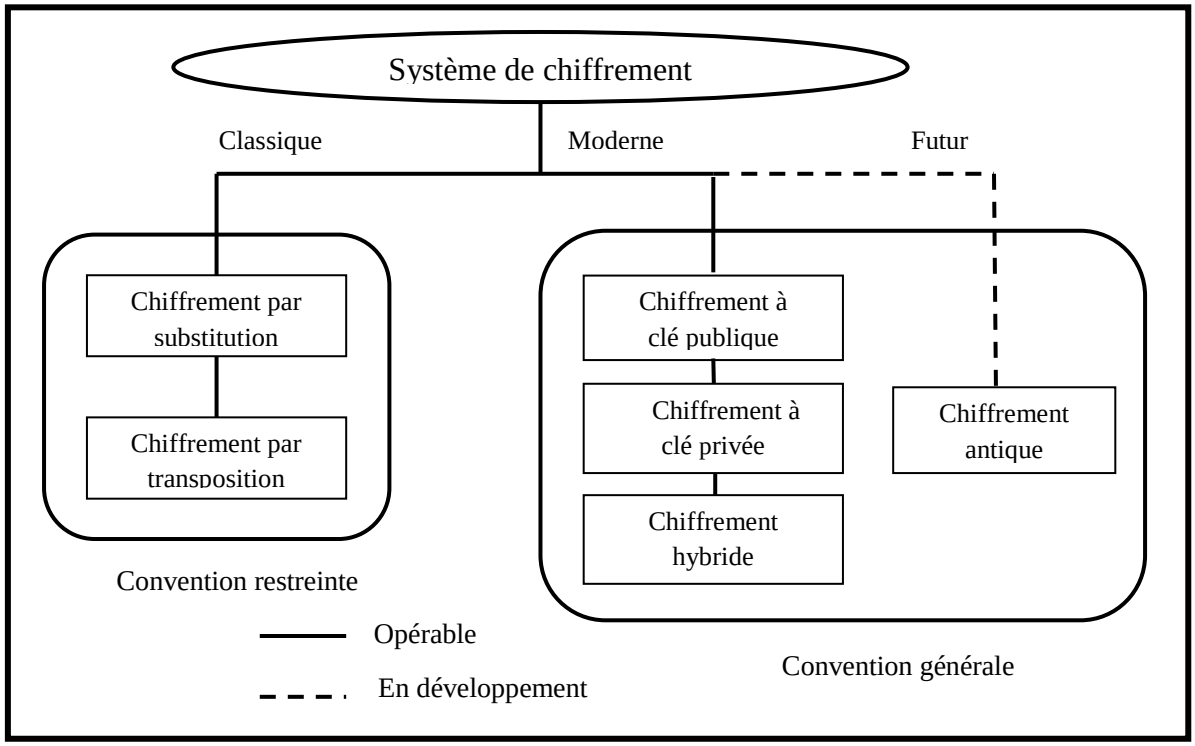


Figure I.3. *Les classes de la cryptographie*

I.3.3.1. La cryptographie classique

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble et ils sont trop simples pour offrir la moindre sécurité. Pour cacher la substance d'un texte, ils utilisent la substitution de caractères par d'autres ou les transposer dans des ordres différents. De ce fait, la confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide. On appelle généralement cette classe de méthodes : le chiffrement **à usage restreint**.

a. Cryptographie par substitution

La substitution signifie que chaque lettre (ou groupe de lettres) est substituée par une (ou groupe) lettre(s), chiffre(s) ou symbole(s). Le déchiffrement consiste à effectuer la substitution inverse. Selon la façon de substituer, on quatre catégories :

- **Substitution simple (mono-alphabétique)**

Le codage par substitution mono-alphabétique (ou encore les alphabets désordonnés) est le plus simple à imaginer. Chaque lettre dans le message clair est remplacée dans le message chiffré par une autre lettre différente unique pour toutes les occurrences de celle-ci.

Dans la littérature, plusieurs algorithmes ont été proposés, entre autres, nous citons : le chiffre de César, le chiffre Atbash, le carré de Polybe, etc.

- **Substitution poly-alphabétique**

Au lieu de remplacer une lettre par une même autre lettre dans tout le message comme dans la substitution simple, elle est remplacée périodiquement par différentes lettres. L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles.

- **Substitutions homophoniques**

Au lieu d'associer un seul caractère crypté à un caractère en clair, on dispose d'un ensemble de possibilités de substitution de caractères dans lequel on choisit aléatoirement. Par exemple : C=>S, K ; G=>G, J ; Q=>K ; S=>S, Z ; PH=>F ; …etc.

- **Substitution par polygrammes**

Au lieu de substituer des caractères, on substitue par exemple des digrammes : groupe de deux caractères. Pour se faire, deux moyen sont utilisés : soit par table (Chiffre de Playfair) ou par transformation mathématique (Chiffre de Hill).

b. Cryptographie par transposition

Elle consiste à permuter les lettres du message à chiffrer entre elles, afin de le rendre inintelligible. Plusieurs variations de transposition sont utilisées, parmi eux on trouve :

- **Transposition simple (à base matricielle)**

Elle consiste à écrire le texte en clair dans une matrice de ***n*** colonnes (une lettre dans chaque case), et ensuite de construire le texte chiffré en prenant les lettres à partir de cette matrice colonne par colonne. La clé dans ce cas est le nombre ***n***.

- **Transposition avec substitution simple**

L'idée dans ce cas est de combiner la transposition avec une substitution simple. Il s'agit ainsi de chiffrer le message clair par une méthode de substitution simple, et en suite d'en appliquer une transposition. Une autre astuce est souvent utilisée qui consiste à appliquer une fonction de permutation sur l'ordre d'arrangement des colonnes. On cite à titre d'exemple : le chiffre de DELASTELLE.

c. Conclusion

Les nouvelles techniques de communications (moyens de transports rapides, journaux, télégraphe, télégraphie sans fil) donne une nouvelle impulsion à la cryptologie. L'interception devient simple et le décryptement des informations devient vital. La cryptologie entre dans son ère moderne et les algorithmes classiques ne sont plus efficaces.

I.3.3.2. La cryptographie moderne

La cryptographie entre dans son ère moderne avec l'utilisation intensive des ordinateurs à partir des années septante. Vue la nécessité croissante de sécuriser les données dans tous les domaines (économique, industriel, informatique,...etc.) …. La cryptographie est appelée à devenir une technique de plus en plus fondamentale pour la protection des informations possédées et/ou échangées par des individus ou des organisations. Dans la cryptographie moderne, on utilise aussi des problèmes mathématiques que l'on ne sait pas (encore) résoudre, par exemple factoriser des grands nombres (chiffre RSA).

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ la *cryptographie à clef secrète,* ou encore appelée *symétrique,*
- ✓ la *cryptographie à clef publique,* dite également *asymétrique*.

a. La cryptographie symétrique

- **Principe**

Le cryptage à clé privée ou symétrique (ou encore dit conventionnel) utilise la même clé pour chiffrer et déchiffrer un message. Autrement dit, les clefs de chiffrement et de déchiffrement sont identiques ou on peut facilement calculer l'une à partir de l'autre. La connaissance de cette clef est cruciale pour la confidentialité des informations échangées. L'emploi d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Ces systèmes sont vulnérables parce qu'il est généralement possible de découvrir la clé à partir des messages codés. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1}. En excepte le système à masque jetable qui est hors porté de cette attaque.

Les algorithmes symétriques sont de deux types :

- ✓ Les algorithmes de *chiffrement en continu*, qui agissent sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé *chiffrement en flux* (Stream cipher).
- ✓ Les algorithmes de *chiffrement par blocs* (Bloc cipher), qui opèrent sur le texte en clair par groupes de bits appelés blocs.

Plusieurs algorithmes de chiffrement symétriques ont été définis. Nous présentons, ci-dessous, les plus connues de ces méthodes.

- **Quelques algorithmes de chiffrement symétrique**

A ce niveau, on va présenter, plus ou moins en détails, les plus fameux des algorithmes de chiffrement s'inscrivant sous ce mode.

1) Masque jetable (one-time pad)

Ce chiffre appelé aussi chiffre de Vernam ou encore le chiffrement parfait; est un algorithme de cryptographie inventé par Gilbert Vernam en 1917. Ce chiffrement est le seul qui soit théoriquement impossible à casser puisque la sécurité de ce système repose sur la génération complètement aléatoire de la clé [www2]. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront équiprobables. Malgré cela, il présente d'importantes difficultés de mise en œuvre pratique, il ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

Il consiste à combiner le message en clair avec une clé présentant les caractéristiques suivantes :

- choisir une clé K_M aussi longue que le texte à chiffrer.
- utiliser une clé constituée de caractères choisis aléatoirement.
- ne jamais utiliser 2 fois la même clé (d'où le nom de masque jetable).
- pour chiffrer un message faire le ou exclusif du message et de la clé : $C=M \oplus K_M$.
- pour déchiffrer un message l'opération est la même : $M= C \oplus K_M = M \oplus K_M \oplus K_M$.

2) DES (Data Encryption Standard)

Jusqu'aux années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence de besoins civils, le NBS (National Bureau of Standards) lança le 15 mai 1973 un appel d'offres dans le Federal Register (l'équivalent du Journal Officiel américain) [Stin, 1996] pour la création d'un système cryptographique qui doit :

- Reposer sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement.
- Etre facile à implémenter, logiciellment et matériellement et être très rapide aussi.
- Avoir un haut niveau de sécurité, lié uniquement à la clé et non à sa confidentialité.

Ce cryptosystème permet de chiffrer des messages de 64 bits avec une clef de 56 bits. De ce fait, c'est un système de chiffrement par blocs. Pour chiffrer un texte, il faut d'abord le découper en blocs de 64 bits puis appliquer le chiffrement sur chacun des blocs. Ainsi, les données en entrée de cet algorithme (texte clair et les données en sortie (texte chiffré) seront des bocs de 64 bits.
Sa clé est une chaîne binaire de 64 bits, mais en fait seuls 56 bits servent réellement à définir la clé. Les bits 8, 16, 24, 32, 40, 48, 56, 64 sont des bits de parité. Le $8^{ème}$ bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Ceci permet d'éviter les erreurs de transmission. Il y a donc pour DES 2^{56} clés possibles, soit environ 72 milliards possibilités. Malgré ça, c'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, alors qu'elle a été largement suffisante au moment de sa conception.

Nous donnons ci-dessous une idée simple et intuitive du fonctionnement du DES qui est un algorithme relativement simple puisqu'il combine des permutations et des substitutions. Il se déroule en quatre étapes [www3] :

1. **Préparation-Diversification de la clé :** le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K 16 sous-clésK_1,…, K_{16} à 48 bits.
   La figure ci-dessous montre comment obtenir à partir d'une clé de 64 bits 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES.

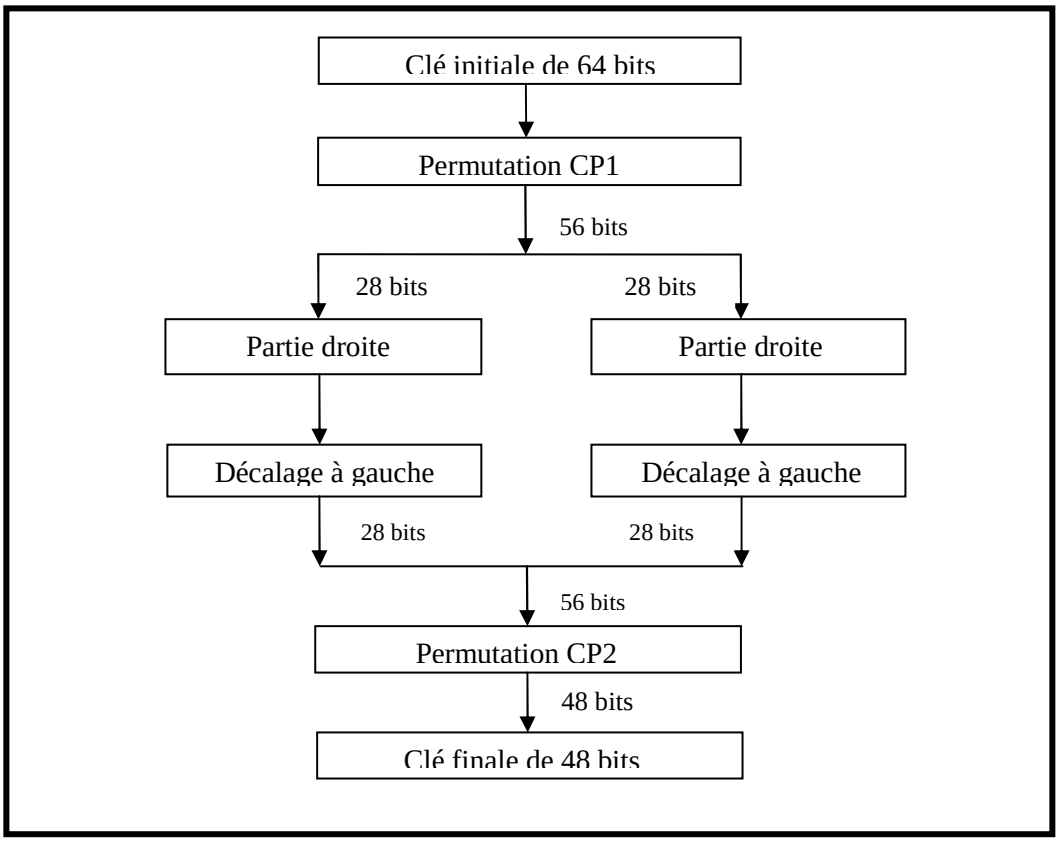


Figure I.4. *Génération des clés.*

La première étape consiste à faire une permutation noté CP1 dont la matrice est présentée ci-dessous, puis éliminer les bits de parité afin d'obtenir une clé d'une longueur de 56 bits. Elle est composée de deux matrice G_i et D_i chacune de 28 bits. Ces deux blocs subissent ensuite une rotation à gauche. Et enfin, ces derniers sont regroupé en un bloc de 56 bits ensuite une permutation CP2 a eu place pour fournir en sortie une clé de 48 bits. Des itérations de l'algorithme permettent de donner les 16 clés K_1, …, K_{16}.

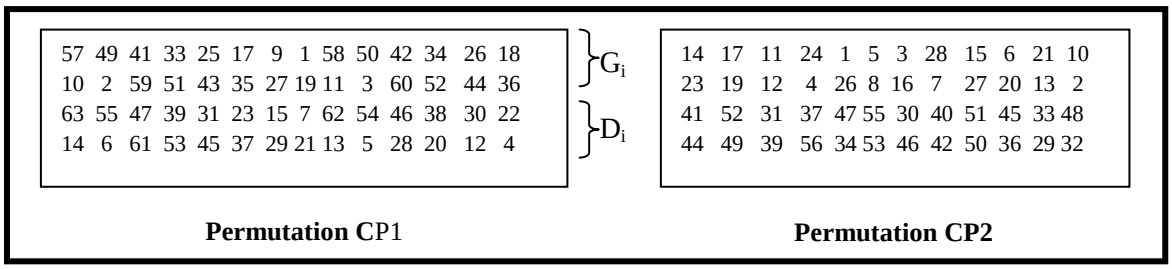


Figure I.5. *Permutation CP1 et CP2.*

2. **Permutation initiale :** pour chaque bloc de 64 bits x du texte, on calcule une permutation finie y=PI(x). y est représenté sous la forme : y = G_0 D_0 , G_{16} étant les 32 bits à gauche de y, D_0 les 32 bits à droite (voir Figure I.6). Quant à leurs significations, par exemple, la permutation initiale (IP) signifie que le $58^{ème}$ bit de la chaine à chiffrer, *x*, est le premier bit de IP(*x*),…

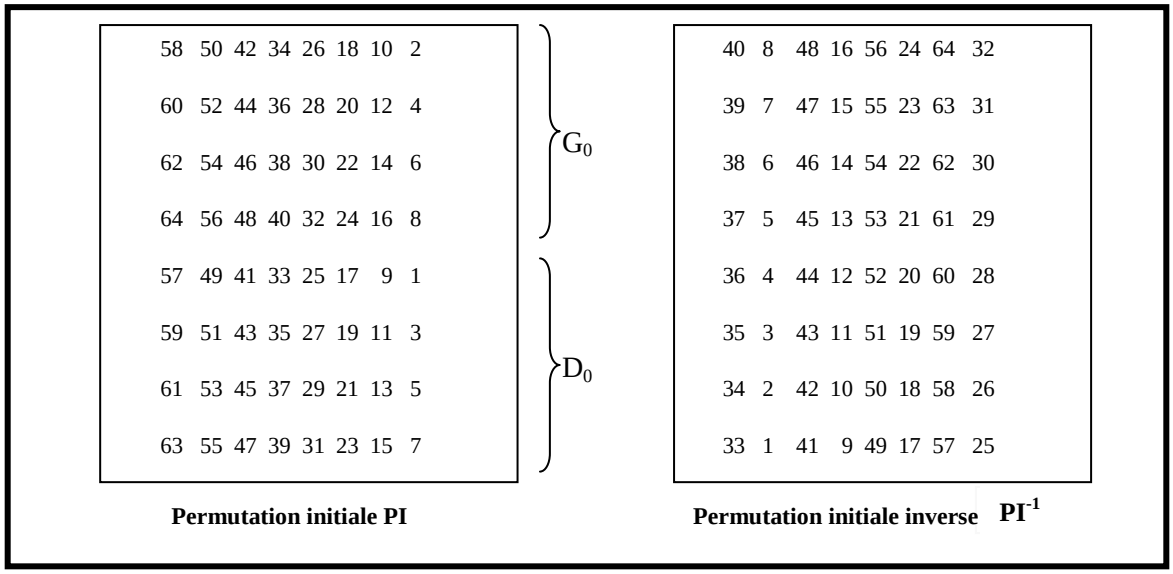


Figure I.6. *La permutation initiale et son inverse.*

3. **Itérations (Rondes) :** On applique 16 rondes d'une même fonction. A partir de $G_{i-1}D_{i-1}$ (pour i de 1 à 16), on calcule G_iD_i en posant :
   - $G_i = D_{i-1}$
   - $D_i = G_{i-1} \oplus f(D_{i-1},K_i)$

D'après la formule qui correspond au calcul de D_i, on constate que la fonction *f* utilise deux arguments ayant des tailles différentes : D_{i-1} de 32 bits et k_i de 48 bits. Ainsi, D_{i-1} sera étendu en 48 bits grâce à une matrice appelée table d'expansion (notée E), dont les 48 bits sont mélangés et 16 d'entre eux sont dupliqués (voire Figure I.7).

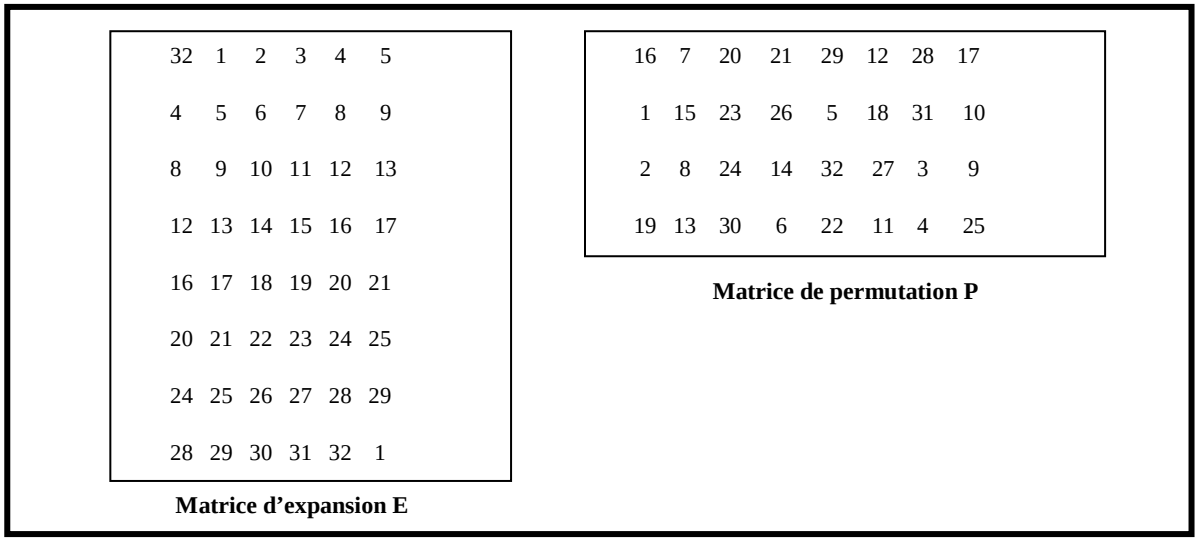


Figure I.7. *Matrice d'expansion E et de permutation P.*

L'opération qui suit consiste à calculer un « ou exclusif » entre le résultat obtenu jusqu'à maintenant, codé sur 48 bits, et G_{i-1} codé sur 32 bits. Donc, le résultat final de *f* doit être codé sur 32 bits au lieu de 48 bits. Pour cela, la chaine de $48 = 8 \times 6$ bits sera transformée en une chaîne de $32 = 8 \times 4$ bits en utilisant des dispositifs appelés *boites-S*. Elles sont au

nombre de huit, où chacune calcule un bloc de 4 bits à partir d'un bloc de 6 bits. Enfin, on applique une permutation à ce 32 bits pour obtenir la valeur finale de *f* (voir la Figure I.7).
L'ensemble de ces résultats en sortie de P est soumis à un « ou exclusif » avec le G_0 de départ (comme il est indiqué dans la Figure I.9) pour donner D_1, tandis que le D_0 initial donne G_1, et ainsi de suite pour les 16 itérations comme il est mentionné auparavant.
La succession d'opérations constituant la fonction ***f*** est représentée à travers la Figure I.8**.**

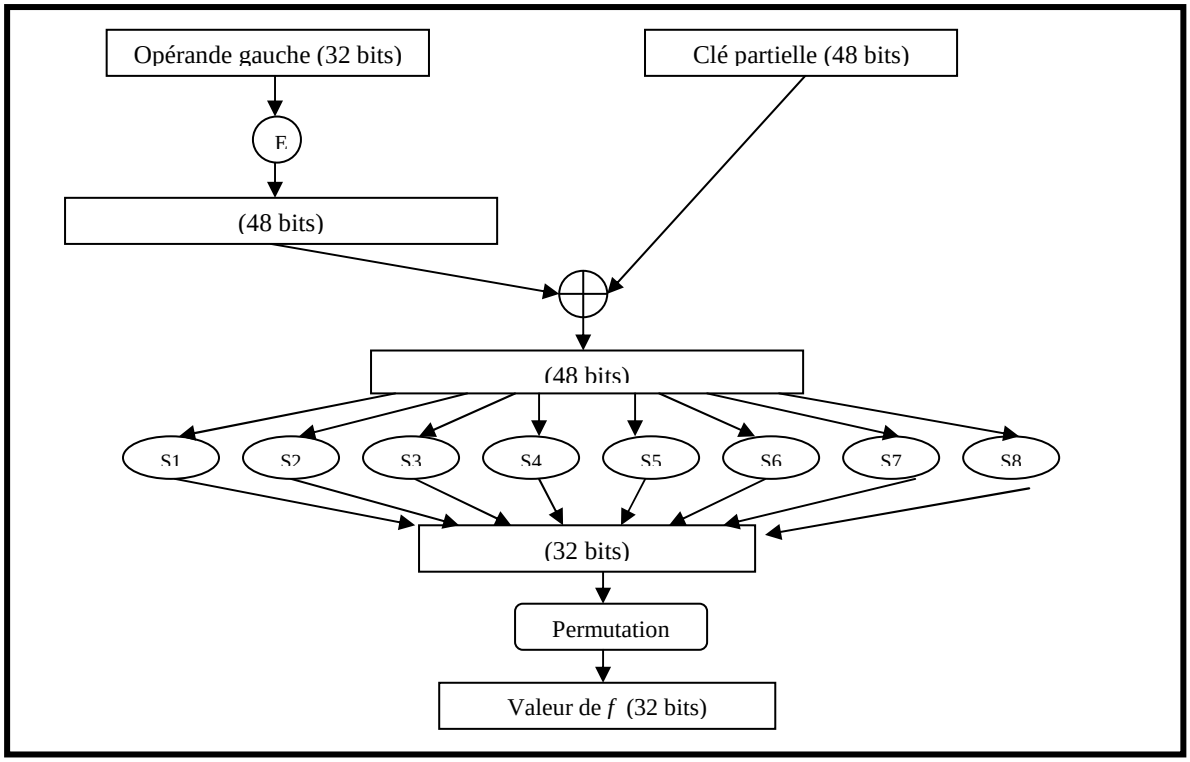


Figure I.8. *Schéma de la fonction f.*

4. **Permutation initiale inverse :** A la fin des itérations, les deux blocs G_{16} et D_{16} sont récoltés, puis soumis à la permutation initiale inverse : $Z = PI^{-1}(G_{16}\ D_{16})$. Le résultat en sortie est un texte codé de 64 bits.
   Cette description peut être résumée par le schéma général illustré par Figure I.9, où on a seulement représenté quelques-unes des 16 étapes.

Remarque : le déchiffrement suit le même algorithme avec la même clef *K*. Seules les sous-clés sont appliquées dans le sens inverse en commençant par la clé k_{16} jusqu'à la clé k_1.

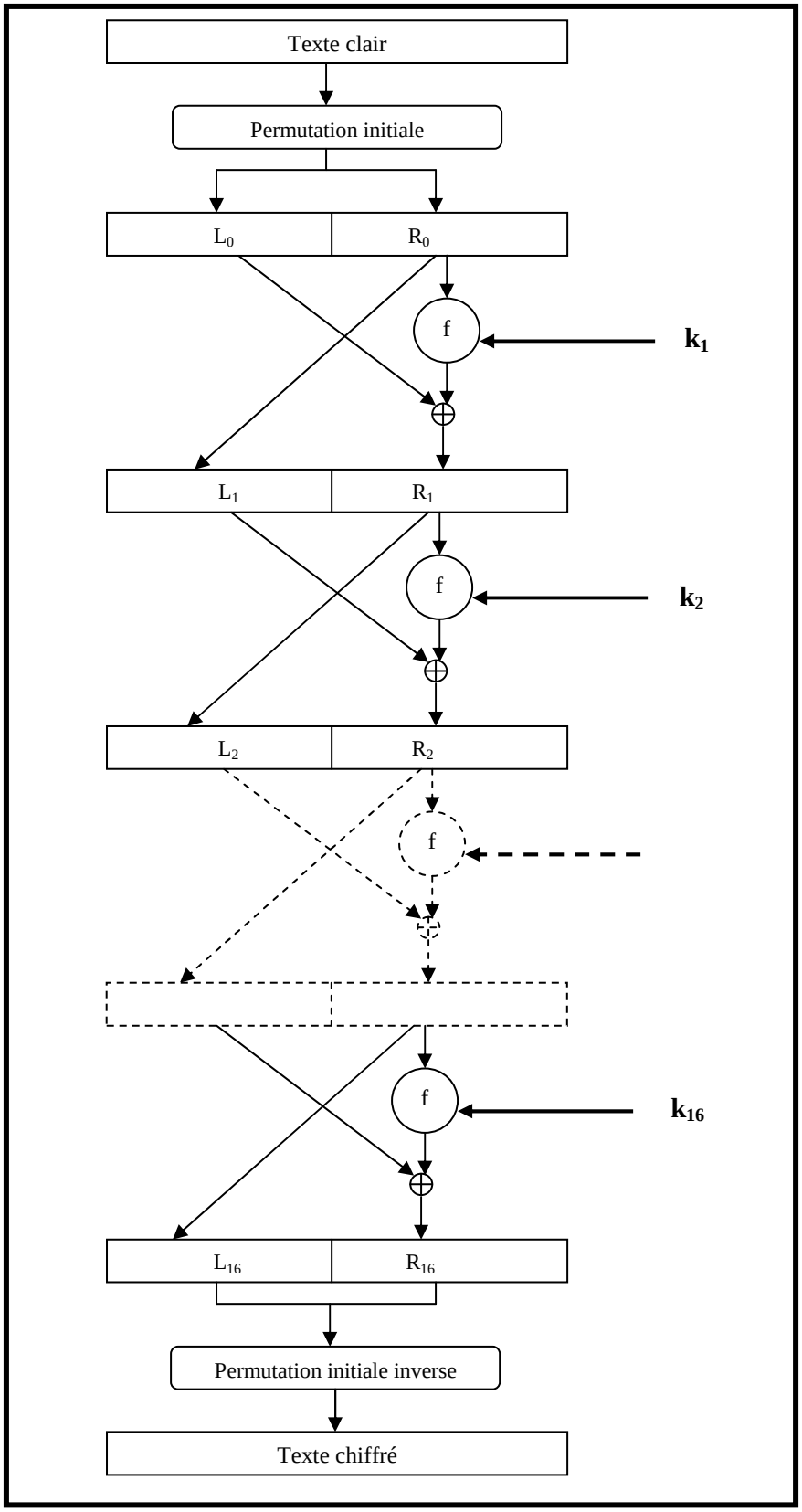


Figure I.9. *Schéma général de DES.*

Le DES a fait l'objet de très nombreuses attaques. On peut citer quelques une :

- **Cryptanalyse différentielle :** c'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires. Ainsi, un DES à 8 ou à 10 tours peut facilement être cassé, mais le DES complet à 16 tours est resté hors de portée de cette attaque.

- **Cryptanalyse linéaire :** c'est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Elle n'est utilisable que pour un DES restreint à quelques tours, mais le DES réel n'est pas menacé par cette attaque.

- **Recherche exhaustive :** les laboratoires RSA ont lancé en Janvier 1997 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES pour démontrer que la taille des clés DES a devenu insuffisante. Ils ont réussi le 17 Juin 1997. Bien qu'une telle recherche demande des moyens considérables, elle a montré que le DES n'offre plus aujourd'hui une grande sécurité.

3) Triple DES (3DES)

Pour palier l'insuffisance cryptographique observée du cryptosystème DES, dû à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. Triple DES est apparait comme une solution pour remédier les faiblesses de DES. Comme son nom l'indique, le principe du triple DES est d'effectuer 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Ce principe peut être formulé comme suit :

$$\textbf{Triple-DES}_{k1,\,k2} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}^{-1}_{k2} \;\textbf{O}\; \textbf{DES}_{k1}$$

On peut constater que le DES sera retrouvé comme cas particulier de la formule ci-dessus, lorsque $k_1 = k_2$. Le déchiffrement de son tour est formulé par :

$$\textbf{Triple-DES}^{-1}_{k1,\,k2} = \textbf{DES}^{-1}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}^{-1}_{k1}$$

Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc composée de 112 bits. Une autre variante à trois clés DES différentes peut être conçue. D'une façon plus formelle, son principe peut être donné par la formule suivante :

$$\textbf{Triple-DES}_{k1,\,k2,\,k3} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}_{k3}$$

Malgré cela, cette variante reste aussi fragile à une attaque de coût en 2^{112} s'appuyant sur l'un des deux messages intermédiaires.

4) AES (**Advanced Encryption Standard**)

Avec le temps et les progrès de l'informatique, les 2^{56} clés possibles du DES se voient incapables de fournir une barrière infranchissable dû à la faiblesse des clés de 56 bits. Désormais avec des moyens modestes, percer les messages chiffrés par DES en un temps raisonnable, ne pose aucun problème. De ce fait, l'AES a vu le jour. Il est issu d'un appel à candidatures international lancé en janvier 1997 et ayant reçu 15 propositions. Au bout de cette évaluation, ce fut le candidat Rijndael du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen [www4] qui a été choisi par le NIST en Octobre 2000 pour être l'algorithme AES et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. II a été déclaré vainqueur de la deuxième ronde dans laquelle

s'opposaient les 5 candidats finalistes (MARS, RC6, Rijndael, Serpent, TwoFish). L'AES a une taille longue de sa clé allant jusqu'à 256 bits qu'il le permet de résister toujours a la cryptanalyse. L'AES est devenu le nouveau standard du chiffrement symétrique comme le signifie cet acronyme. L'AES est libre d'utilisation comme l'est l'algorithme DES.

Techniquement, le chiffrement AES travaille avec des blocs de 128 bits seulement et la longueur des clés utilisées peut être de 128, 192 ou de 256 bits. Chaque bloc subit une séquence de transformations que nous résumons à travers les points suivants :

1. Addition de la clé secrète et du bloc en question avec un « ou exclusif ».
2. ByteSub : les 128 bits sont répartis en 16 blocs de 8 bits (16 octets), qui sont ensuite placés dans une matrice de 4×4. Chaque octet est transformé par une fonction non linéaire S (S-box) conçu pour résister à la cryptanalyse linéaire et différentielle.
3. ShiftRow: les lignes de cette matrice sont soumises à une rotation vers la droite où l'incrément pour la rotation varie selon le numéro de la ligne. La $2^{ème}$ ligne est décalée d'une colonne, la $3^{ème}$ ligne de 2 colonnes, et la $4^{ème}$ ligne de 3 colonnes.
4. MixColumn: chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne. Cela revient à multiplier la matrice 4×4 par une autre matrice 4×4.
5. AddRoundKey: une clé dite *de tour* est générée à partir de la clé secrète par un sous-algorithme dit *de cadencement*. Cette clé de tour est ajoutée par un « ou exclusif » au dernier bloc obtenu.

Ces différentes opérations, définissant un *tour*, sont répétées plusieurs fois. Cependant dans le dernier tour il n'y a pas d'opération MixColumn. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours. La figure suivante résume le principe de fonctionnement de cet algorithme de chiffrement.

Le déchiffrement consiste en appliquer les opérations inverses dans chacune des étapes (*InvSubBytes*, *InvShiftRows*, *InvMixColumns*). *AddRoundKey* (à cause du XOR) est son propre inverse. On réitère ce processus le nombre de tour-1. Pour le dernier tour on exclu l'opération *InvMixColumns* ; et comme dans le chiffrement pas d'opération *InvMixColumns* dans le dernier tour.

De nombreux travaux de cryptanalyse ont été publiés mais pour l'instant il n'a pas été cassé et la recherche exhaustive demeure la seule solution. En particulier, on cite :

- **Attaques sur des versions simplifiées :** *Niels Ferguson* et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours. En effet, cette dernière attaque repose sur le principe des clés apparentées. Dans une telle attaque, la clé demeure secrète mais l'attaquant peut spécifier des transformations sur la clé et chiffrer des textes à sa guise. Il peut donc légèrement modifier la clé et regarder comment la sortie de l'AES se comporte.

- **Attaques sur la version complète :** plusieurs chercheurs ont mis en évidence des possibilités d'attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL. Le XSL fait appel à une analyse heuristique dont la réussite n'est pas systématique. Elles sont impraticables car le XSL demande au moins 2^{87} opérations voire 2^{100} dans certains cas. Le principe est d'établir les équations (quadratiques / booléennes) qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer et l'AES est toujours considéré comme sûr.

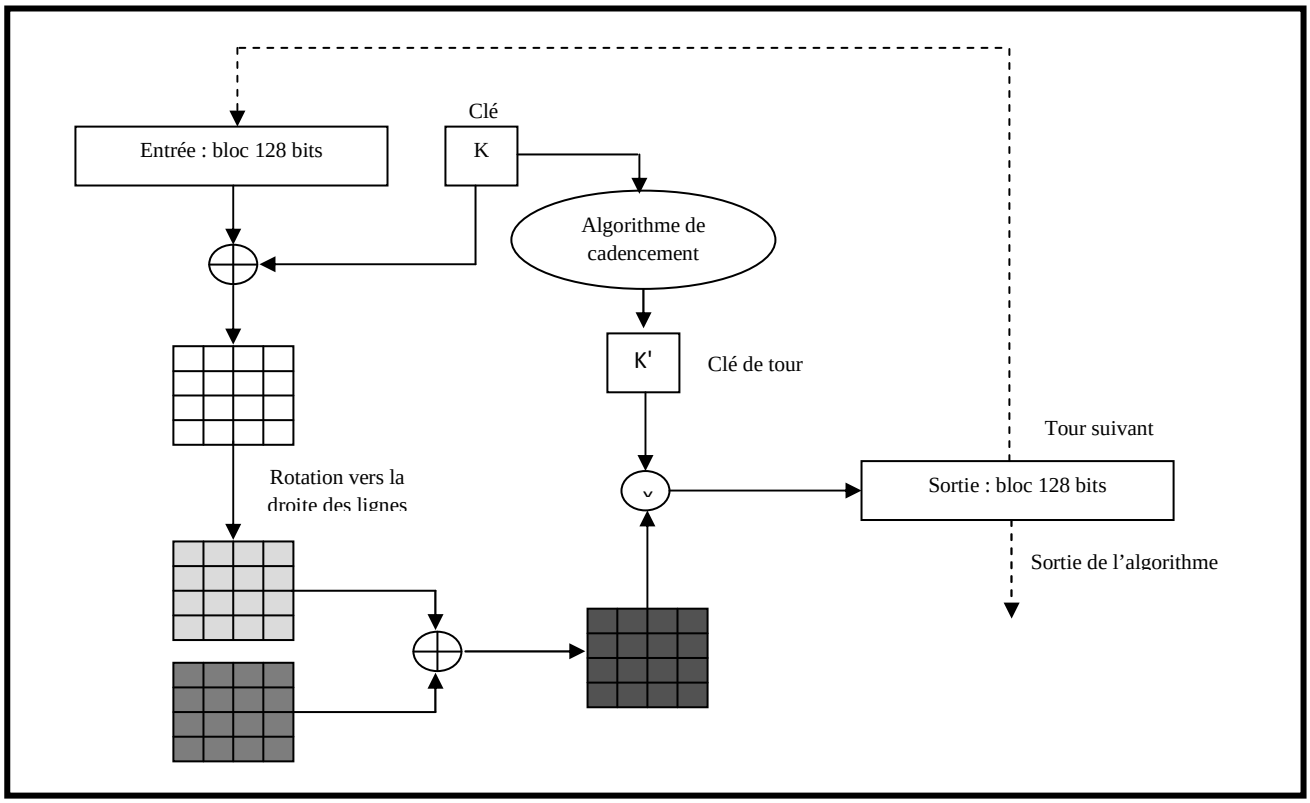


Figure I.10. *Schéma général de l'AES.*

5) IDEA (International Data Encryption Algorithm)

L'**IDEA** qui est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 rondes inventé en 1990 à Zürich par James L. Massey et Xuejia Lai. C'est la taille des clés qui apporte une grande sécurité au système (que l'on n'a pas, à l'heure actuelle, officiellement réussi à "casser").

Chaque bloc est divisé en 4 sous-blocs de 16 bits : X_1, X_2, X_3 et X_4. Ces quatre sous-blocs deviennent les entrées de la première ronde de l'algorithme.
À chaque ronde, la séquence d'évènements est la suivante (voir Figure I.11) :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé ;
5. combinez par OU exclusif les résultats des étapes (1) et (3) ;
6. combinez par OU exclusif les résultats des étapes (2) et (4) ;
7. multipliez le résultat de l'étape (5) avec la cinquième sous-clé ;
8. additionnez les résultats des étapes (6) et (7) ;
9. multipliez le résultat de l'étape (8) par la sixième sous-clé ;
10. additionnez les résultats des étapes (7) et (9) ;
11. combinez par OU exclusif les résultats des étapes (1) et (9) ;
12. combinez par OU exclusif les résultats des étapes (3) et (9) ;
13. combinez par OU exclusif les résultats des étapes (2) et (10) ;
14. combinez par OU exclusif les résultats des étapes (4) et (10).

La sortie de la ronde est constituée des 4 sous-blocs produits par les étapes (11), (13), (12) et (14). Changez les deux blocs intérieurs (sauf lors de la dernière ronde) et cela donne l'entrée de la ronde suivante.
Après la huitième ronde, il y a une transformation finale :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé.

Enfin les 4 sous-blocs sont réassemblés pour former le texte chiffré.

Chaque ronde utilise 6 sous-clés K_i^r de 16 bits : K_1^1, ..., K_6^1, …, K_1^8, …, K_6^8. La transformation finale utilise 4 sous-clés : $K_1^9, K_2^9, K_3^9, K_4^9$; donc un total de 52 sous-clés est utilisé, les premiers 8 sous-clés sont extraites directement à partir de clé, des groupes de 8 clés sont extraits par rotation à gauche de 25 bits de la clé K, ce qui donne 6 rotations en total.
Concernant le processus de déchiffrement on utilise celui de chiffrement avec un seul changement dans la génération des clés. En fait, on utilise K pour générer les sous clés K_i^r ; à partir de ces derniers, d'autres clés $K_i^{'r}$ sont obtenues dans le Tableau I.1 [Omar, 2006] ; ensuite on utilise $K_i^{'r}$ à la place des K_i^r dans l'algorithme de chiffrement IDEA. Dans le Tableau I.1, $-K_i$ dénote l'opposé modulo 2^{16} de K_i. K_i^{-1} dénote l'inverse multiplicative de $K_i \bmod(2^{16}+1)$, se trouvant aussi dans {0, 1, …,216-1}.

Ronde r	$K_1^{'(r)}$	$K_2^{'(r)}$	$K_3^{'(r)}$	$K_4^{'(r)}$	$K_5^{'(r)}$	$K_6^{'(r)}$
r =1	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
2 ≤ r ≤ 8	$(K_1^{10-r})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
r= 9	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	--	--

Tableau I.1. *Les sous clés de déchiffrement $K_i^{'r}$ générées à partir des sous clés K_i^r.*

La méthode de génération des sous-clés de l'IDEA est toujours régulière et donc pourrait être une faiblesse à l'algorithme. Cependant, il est considéré comme étant hautement sécuritaire. En effet, pour résoudre IDEA avec l'attaque en force brute, il faudrait effectuer 2^{128}, donc 10^{38} opérations [www5].

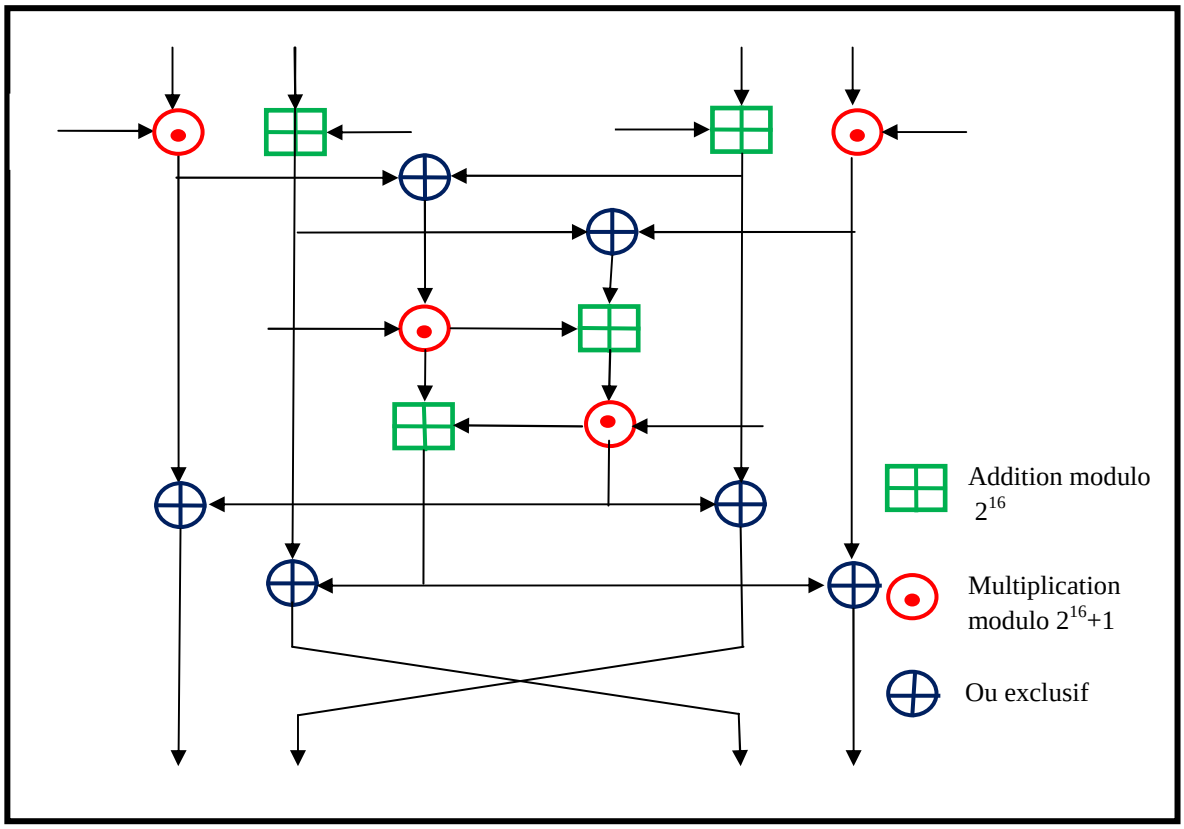


Figure I.11. *Schéma général de l' IDEA.*

- **Conclusion**

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. AES est immunisé contre les attaques et il reste néanmoins le meilleur choix dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause. D'autre part, IDEA est considéré par les spécialistes comme l'un des meilleurs cryptosystèmes à clé privée. Il est utilisé par le PGP pour le chiffrement de données.

b. La cryptographie asymétrique

- **Principe**

La *cryptographie asymétrique* ou encore dite *à clé publique* utilise deux clés différentes pour chaque utilisateur : une est privée et n'est connue que de l'utilisateur et qui est sensé d'être en mesure de faire signature ou déchiffrement. L'autre est publique diffusée en général dans un annuaire et donc accessible par quiconque afin de permettre aux interlocuteurs de mettre en œuvre les opérations réciproques (vérification de signature ou chiffrement de message). Les clés, publique et privée sont mathématiquement liées par l'algorithme de cryptage de telle manière qu'un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de *fonction à sens unique avec trappe*. Sachons qu'une fonction est appelée à *sens unique* si elle est

aisément calculée, mais extrêmement difficile de déduire la fonction inverse. Et elle sera dite *à trappe*, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la *trappe*. L'utilité d'utilisation d'une telle fonction réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas assez facile puisqu'elles s'appuient généralement sur des problèmes mathématiques réputés difficiles.

Ce cryptage présente l'avantage de permettre le placement de signature numérique dans le message et ainsi permettre l'authentification de l'émetteur grâce à la fonction de hachage. Le principal avantage consiste à résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé puisque la clé privée n'est connue que par l'utilisateur. Bien que plus lent que la plupart des cryptages à clé privée il reste toujours préférable pour 3 raisons :

- ✓ plus évolutif pour les systèmes possédant des millions d'utilisateurs.
- ✓ authentification plus flexible.
- ✓ supporte les signatures numériques.

Les systèmes asymétriques les plus connus sont [Mene, 1996] : RSA basé sur la difficulté de la factorisation des grands entiers, El Gamal basé sur la difficulté de résoudre le problème du logarithme discret dans un corps fini et les systèmes sur les courbes elliptiques basés sur la difficulté de certains calculs sur les courbes elliptiques.

- **Quelques algorithmes asymétriques**

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivants :

1) RSA

La méthode de cryptographie RSA a été inventée en 1977. Elle titre son nom des noms de ses trois inventeurs: *R. Rivest, A. Shamir* et *L. Adleman* [www6]. Le RSA est le premier et est encore le système cryptographique à clé publique le plus utilisé de nos jours et toujours considéré comme sûr. Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers ; à l'heure actuelle, il est pratiquement impossible de les reconstituer en un temps raisonnable. Ainsi, la sécurité de RSA semble satisfaisante malgré qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser.

On peut résumer le fonctionnement de ce cryptosystème dans les étapes suivantes :

- **Fonction d'encodage E (publique) :** la clé publique k utilisée pour l'encodage comporte deux entiers: k = (e,n). L'opération de chiffrement se fait au moyen de l'élévation à la puissance **e** modulo n : $E_k(M) = M^e \bmod n$.
- **Fonction de décodage D (privée) :** la clé secrète k′ utilisée pour le déchiffrement est aussi un couple d'entiers : k′(d,n). Pour reconstituer le message initial, la fonction inverse d'encodage est appelée, elle est ainsi : $D_{k'}(M) = M^d \bmod n$.
- **Détermination des clés :**
  - **Détermination de n :** pour calculer n on doit initialement choisir deux entiers premiers *p* et *q* très grands et leurs valeurs sont secrètes connus que part l'utilisateur. Le choix de *p* et *q* affecte grandement le niveau de sécurité de RSA. Pour cela, il faut évidemment se prémunir contre les algorithmes de factorisation dont la complexité dépend essentiellement de la taille du plus petit facteur premier de *n* [Zimm, 2005], donc si possible choisir *p* et *q* de même taille.

- **Détermination de e :** pour calculer e, on calcule premièrement un entier z = (p-1)*(q-1) puis tout simplement choisir un entier e premier avec z.
- **Détermination de d :** pour calculer d, on procède ainsi : e*d≡1[z].

La sécurité de RSA est basée sur l'hypothèse que la fonction E est à sens unique, ce qui rend impossible à décrypter un texte chiffré. Mais à l'aide de la trappe qui est la factorisation n = p*q, il est possible d'en trouver d et donc la clé privée.

Depuis son apparition, plusieurs attaques ont été découvertes contre le RSA. Et même si aucune de ces attaques n'est réellement destructive, elles démontrent toutefois qu'il faut implémenter RSA avec beaucoup de précautions. La fameuse attaque est :

- **Recherche exhaustive :** comme la fonction de chiffrement RSA est déterministe, si l'ensemble des messages possibles est connu et de petite taille, il sera facile de décrypter par une recherche exhaustive [Ster, 2004]. Pour éviter cette attaque, il est indispensable de randomiser les messages avant chiffrement.

2) Chiffrement d'ElGamal

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste c.à.d. un même message M peut avoir plusieurs chiffres différents [Lafo, 2006]. Il est basé sur la difficulté du problème des logarithmes discrets c.à.d. la difficulté de trouver l'unique *a* noté $\log_\alpha ß$ / $0 \leq a \leq p-2$ tel que : $\alpha^a \equiv ß \pmod p$ où *p* est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$. Cependant, et pour éviter les attaques connues, *p* doit être convenablement choisi, et $p-1$ doit avoir un grand facteur premier.

D'une manière formelle, l'algorithme ElGamal peut être résumé comme suit :

Soit *p* un nombre premier tel que le problème du logarithme discret dans Z_p soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et $K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod p\}$
Les valeurs *p*, α et ß sont publiques, et *a* est secret.
Pour k = (*p*, α, *a*, ß), et pour un $k \in Z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$\mathbf{e_k(x, k) = (y_1, y_2)}$$

Où

$$\mathbf{y_1 = \alpha^k \bmod p}$$

Et

$$\mathbf{y_2 = x\, ß^k \bmod p}$$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit comme suit :

$$\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$$

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par $ß^k$, en produisant y_2 ;
- ✓ la valeur α^k est également transmise en tant que partie du texte chiffré ;
- ✓ Bob, qui connaît l'exposant secret *a*, peut calculer $ß^k$ à partir de α^k. Il peut alors « enlever le masque » en divisant y_2 par $ß^k$ et obtenir le texte clair x.

Une attaque possible à ce cryptosystème est celle dite ***man in the middle***. Son principe dans le cas d'ElGamal fonctionnant avec le mode *Diffie-hellman* est illustré sur la figure I.12.

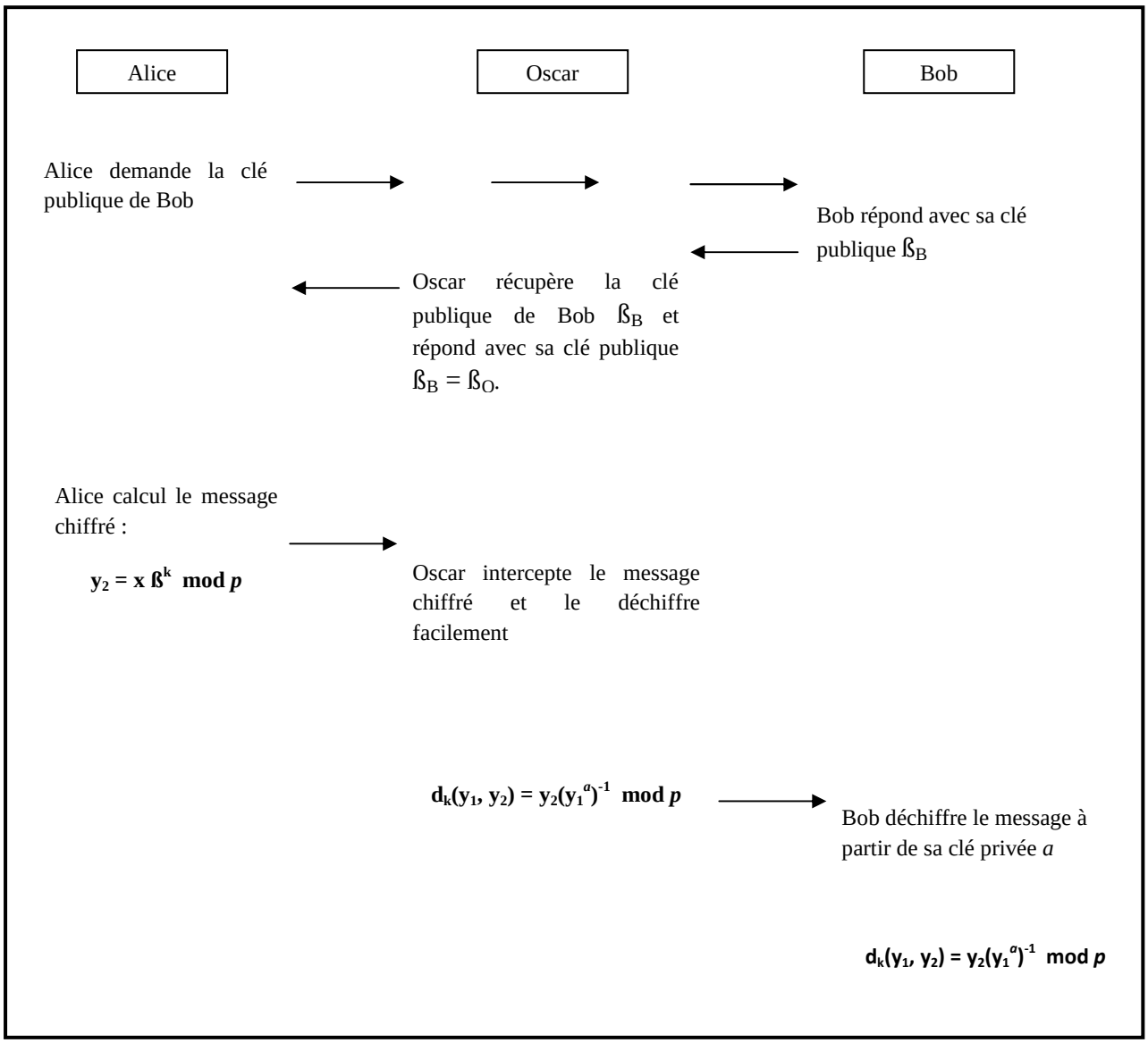


Figure I.12. *Principe de l'attaque « man in the middle ».*

- **Conclusion**

Les calculs faits en 1995 ont ouvert un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent des chiffres plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique serait capable de remettre en cause ce cryptosystème.

De même, casser l'algorithme ElGamal est aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Et pour assurer une bonne sécurisation de cet algorithme, Zimmermann en 2005 [Zimm, 2005] a recommandé l'utilisation des clés d'au moins 1024 bits.

c. Comparaison entre les cryptosystèmes symétriques et asymétriques

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage.

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ?
À partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que l'importance de la taille de clé n'est légitime que dans le cas de la clé secrète, puisque les seules attaques possibles sont les attaques exhaustives [www7]. Mais, dans le cas de la clé publique, la taille de la clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

Méthode	Avantages	Inconvénients
À clefs Secrètes	▪ Rapidité de calcul en général (dépend de la taille de la clé). ▪ Adaptée au cryptage de flux de données.	▪ Moins sécurisé (DES). ▪ Problème de communication de clefs entre émetteur et récepteur. ▪ Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clefs Publiques	▪ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre. ▪ Permet la signature électronique. ▪ Un couple de clés publique/privée suffisant pour 'n' correspondants.	▪ Lente. ▪ Problèmes de gestion de clefs publiques.

Tableau I.2. *Comparaison entre les méthodes de chiffrement symétriques et asymétriques.*

d. La cryptographie hybride

- **Principe**

La cryptographie asymétrique est beaucoup plus lente que la cryptographie symétrique qui brille par sa rapidité. En revanche, cette dernière souffre d'une grave lacune ; assurer une transmission secrète de la clé. Pour palier ce défaut et cumuler les avantages des deux méthodes on a fait recourt à la cryptographie hybride. On code tout d'abord les données avec une clé privée dite *clé de session*, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Comme la clé est courte, on utilise l'algorithme asymétrique puisqu'il prend peu de temps. En revanche, chiffrer l'ensemble du message avec un algorithme asymétrique serait plus lourd. Il suffit ensuite d'envoyer le message chiffré avec une clé privée et accompagné de cette dernière chiffrée avec une clé publique. Le destinataire procède inversement, il commence à déchiffrer la clé symétrique avec sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, via un déchiffrement symétrique pour retrouver le message original. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

- **Quelques algorithme hybride**

1) PGP (**Pretty Good Privacy**)

PGP est inventé par *Philip Zimmermann*, qui dise que tout individu a droit à la confidentialité, notamment les organisations des droits de l'homme dans des pays soumis à la dictature. Il l'a mis à disposition gratuitement sur Internet. Cela lui a valu de sérieux ennuis avec la justice américaine, car les logiciels de cryptage sont considérés comme du matériel de guerre et sont interdits à l'exportation.

Il est souvent utilisé pour garantir l'authentification et le contrôle d'intégrité des fichiers et du courrier électronique. Mais avant de crypter un texte avec PGP, les données doivent tout d'abord compressées dont le but de réduire le temps de transmission, ainsi d'économiser l'espace disque et, surtout, de renforcer la sécurité cryptographique puisque les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement alors que la compression réduit ces modèles dans le texte en clair.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes qui sont [www8] :

✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
✓ PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de déchiffrement se fait également en deux étapes, qui sont [www8] :

✓ PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
✓ PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

Depuis 1978, la recherche universitaire civile a intensément attaqué la cryptographie à clé publique, sans pour autant réussir à la remettre en cause. Mais cela ne fournisse aucune garantie totale sur la sécurité de cette manière de chiffrer, car une attaque menée par le gouvernement, par exemple, qui ne se manque pas de ressources très développées ou même en utilisant quelques nouvelles percées mathématiques classées top-secret, peut tenir à bout ces cryptosystèmes conventionnels utilisés dans PGP.

Tout de même, l'optimisme semble justifié. Les algorithmes de clé publique, les algorithmes de contraction de message, et les chiffres par blocs utilisés dans PGP ont été conçus par les meilleurs cryptographes du monde [Loid, 2005]. Les chiffres de PGP ont subi des analyses de sécurité approfondies et des examens méticuleux de la part des meilleurs cryptographes dans le monde non classé top secret. De plus, et même si les chiffres par blocs utilisés dans PGP possèdent quelques faiblesses, la compression du texte clair utilisée avant le chiffrement réduit de façon considérable ces faiblesses.

2) GPG (**GNU Privacy Guard**)

GPG est la version GNU de PGP. En raison qu'il est sous licence GPL (General Public License) il permet la fourniture du code-source, la libre modification et la libre redistribution de ce code-source. Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

- **Conclusion**

Aujourd'hui, de nombreux gouvernements ont restreint l'usage du PGP, qui a été largement diffusé par son développeur, en pensant qu'un cryptage trop fiable ferait le jeu des terroristes et des trafiquants. Toutefois, il y'a pas mal d'applications qui utilisent encore ce système de chiffrement, telles que les paiements en ligne qui se font grâce au procédé SSL

fonctionnant selon le principe du PGP. De sa part, GPG, qui est un cryptosystème utilisant des algorithmes de chiffrement à clé publiques (DSA, RSA et ElGamal), est largement utilisé dans les communications par messagerie, c.-à-d. pour chiffrer des mails, ainsi que pour signer des données.

I.3.3.3. Cryptographie quantique

a. Principe

La cryptographie à base d'algorithmes aura toujours des faiblesses. Même si la cryptanalyse bloque devant un algorithme, la force brute pourra toujours décrypter n'importe quel code si on lui donne assez de temps. Même avec le plus solide des chiffrages, le contenu du message peut être subtilisé et dupliqué. Les clés, quant à elles, peuvent être volées en chemin ou présenter des faiblesses qui les rendent prévisibles.

Si maintenant on base notre cryptographie, non pas sur un algorithme mathématique, mais sur des lois de la physique quantique, il n'y plus de force brute qui peut briser un code ici. Le cryptage ne se trouve pas dans des formules, mais dans des photons, ainsi tout le monde peut accéder à des formules, mais pas tout le monde peut accéder aux photons sans que le message devienne inutile. De ce fait, l'information n'est plus sécurisée par des subterfuges mathématiques, mais plutôt par des lois fondamentales de physique. Au de-là, la cryptographie quantique a vu le jour.

La figure I.13 [Nave, 2002] présente un exemple relatif à la possibilité soit qu'un photon traverse le filtre ou pas, selon l'orientation de sa polarisation. Sachons qu'un filtre permet de distinguer entre les photons polarisés horizontalement (0°) et verticalement (90°) ; un autre entre les photons polarisés en diagonale (45°, 135°).

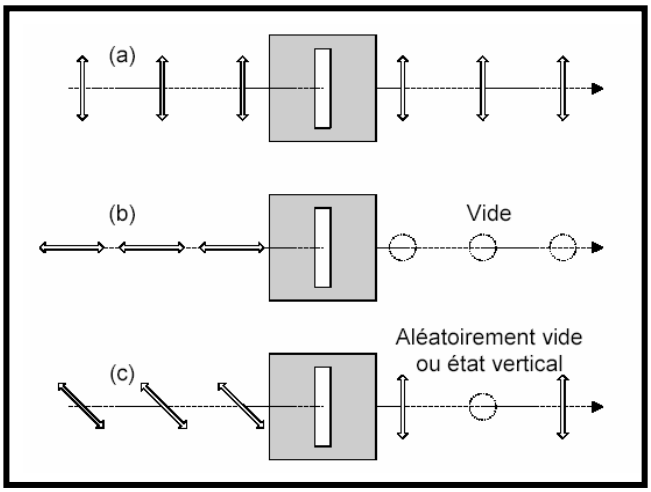


Figure I.13. *Photon unique traversant un filtre ne laissant passer que la lumière polarisée verticalement :*
(a) *Les états polarisés verticalement traversent le filtre sans être absorbés*
(b) *Les états polarisés horizontalement sont tous absorbés*
(c)*Les états polarisés diagonalement sont aléatoirement absorbés ou transmis*

Pour mieux comprendre ce type de cryptographie, on va l'aborder à partir d'un exemple dont le protocole est bien : le protocole BB84. Avant de commencer, on doit mentionner qu'il existe deux modes de polarisations possibles [www9] :

- **Mode 1 :** "0" est codé par un photon d'axe de polarisation 0° et "1" par un photon de polarisation 90°.
- **Mode 2 :** "0" est codé par un photon d'axe de polarisation 45° et "1" par un photon de polarisation 135°.

Pour plus de clarté, ces deux modes peuvent être schématisés comme suit :

Figure I.14. *Les deux modes de polarisation.*

Cet exemple se déroule en plusieurs étapes que l'on va présenter ci-dessous. On a Alice l'expéditrice, Bob le destinataire et Oscar l'éventuel espion [www9]:

1. Alice génère aléatoirement un bit selon un mode (mode 1 ou mode 2) choisi lui aussi aléatoirement, ce que la physique quantique permet, et transmet le photon obtenu à Bob. Elle répète cette opération autant de fois que nécessaire.
2. Pour chaque photon reçu, Bob choisit aléatoirement un mode, c'est-à-dire un polariseur, et note si le photon traverse le polariseur.
3. Ensuite une phase de réconciliation : Alice et Bob communiquent entre eux et pour chaque photon ils comparent si leurs modes coïncident, si c'est le cas, ils ont le même bit, qu'ils conservent, sinon ils ne conservent pas le bit en question. Vue le nombre de photons inutiles, il est donc nécessaire de prévoir un excédent suffisant de photons pour obtenir une clé de longueur donnée. Comme Bob a une chance sur deux de choisir le bon mode, en moyenne on observe ***N*** erreurs pour ***2N*** photons envoyés.
4. Enfin, Alice et Bob contrôlent la sûreté de la clé : parmi les bits conservés ils en choisissent un certain nombre qu'ils comparent publiquement, s'ils ont été espionnés ils obtiendront en moyenne 25% de bits différents (Comme Oscar ne peut pas cloner les photons, il est dans l'obligation de les intercepter, de les mesurer et de renvoyer à Bob un photon similaire à celui qu'il a mesuré. Cependant il a une chance sur deux de choisir le mauvais mode et donc de renvoyer un photon différent de celui qu'Alice avait envoyé à Bob. Puis comme Bob a lui aussi une chance sur deux d'avoir le bon mode, en cas d'espionnage pour chaque photon on a une chance sur 4 (25%) de détecter l'intrusion). Dans ce cas, ils n'utiliseront pas la clé. S'ils n'ont pas de différences, alors ils peuvent conserver la clé pour l'utiliser ultérieurement.

Alice et Bob décident alors de sacrifier une partie de leur clé commune et les comparent publiquement par le canal radio (comme exemple de canal). Dans le Tableau I.3 ils ont quatre bits différents c.à.d. N bits parmi 2N. Ainsi la clé obtenue : 0011.

Photon envoyé par Alice								
Mode choisi par Bob	Mode 1	Mode 2	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								
Après réconciliation								

Tableau I.3. *Exemple sans Oscar.*

Photon envoyé par Alice								
Mode choisi par Oscar	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 2	Mode 2
Résultat mesuré et renvoyée à Bob par Oscar								
Mode choisi par Bob	Mode 1	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								

Tableau I.4. *Exemple avec Oscar.*

Plusieurs cas de figure se présentent à nous dans le Tableau I.4 [Camp, 2010] :

1. Oscar et Bob ont tous les deux le bon mode, alors Oscar mesure bien la bonne polarisation qu'il renvoie à Bob et puis de même pour Bob. Le bit est valable pour la clé et Oscar est passé inaperçue. Exemple : premier photon envoyé.
2. Oscar a le bon mode, mais pas Bob, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon. Oscar passe encore inaperçue, mais sa bonne mesure lui est inutile. Exemple : le deuxième photon envoyé.
3. Oscar a le mauvais mode, mais Bob a le bon mode, alors d'après ce que l'on a vu, Bob a une chance sur deux d'obtenir une mesure compatible avec ce qu'à envoyer Alice. Donc Oscar a une chance sur deux d'être détecté, de plus sa mesure est inutile. Exemple : sixième photon envoyé (avec détection) et dernier photon envoyé (sans détection).
4. Oscar et Bob ont le mauvais mode, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon, Oscar passe donc inaperçue. Exemple : cinquième photon envoyé.

Une fois la réconciliation terminée il ne reste que les bits de la possibilité 1 et 3 qui sont équiprobables. Pour la première possibilité Oscar passe inaperçue, et pour la troisième il a une chance sur deux de se faire détecter. On retrouve bien le 25% ($\frac{1}{2}\frac{1}{2} = \frac{1}{4}$) vu ci-dessus.

b. Conclusion

Le MIT (Massachussets Institute of Technology) a récemment classé la cryptographie quantique parmi les «10 inventions qui changeront le monde». Si on voit peu les avantages à l'heure actuelle, les possibilités pour le futur sont impressionnantes. Le centre canadien IRCA est un leader mondial en recherche quantique, comptant parmi ses membres des chercheurs montréalais au front de cette révolution du monde des télécommunications. Avec le développement du calcul et de la cryptographie quantique, on peut espérer qu'Alice et Bob pourront un jour communiquer en toute quiétude. Entre temps, considérons d'un œil sceptique les soi-disant algorithmes « incassables » qui n'attendent que le prochain superordinateur pour céder comme une coquille d'œuf [Camp, 2010].

I.3.3.4. Algorithme de chiffrement évolutionniste OTL

Omary Fouzia a développé en 2006 un nouvel algorithme de chiffrement en le simulant à un problème d'optimisation dont la résolution est par les algorithmes génétiques. Le but de cet algorithme est de modifier au maximum les fréquences d'apparition des caractères dans le message à chiffrer et d'établir le plus de désordre dans leurs positions. Ces caractères appartiennent à l'ensemble des 256 caractères du code ASCII. L'application de l'algorithme est précédée d'une phase de brouillage du texte initial (M_0) qui peut combiner plusieurs méthodes simples comme les substitutions, les permutations, le chiffrement affiné, etc..., pour obtenir un texte initialement chiffré (M_0').

Pour ce faire, le codage adopté pour représenter les individus, qui sont les différents messages, est résumé à travers la figure ci-dessous :

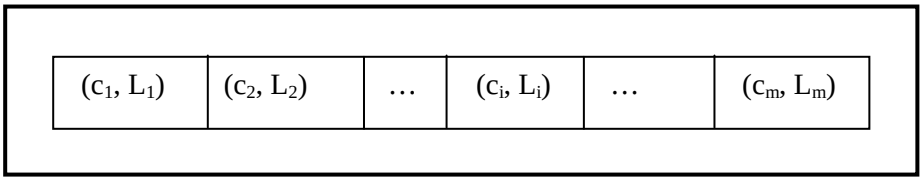


Figure I.15. *Codage des individus.*

Avec :

c_i : un caractère appartenant au message.
L_i : la liste des positions du caractère c_i.
m : le nombre des différents caractères du message.
$L_i \cap L_j = \emptyset, \ \forall i, j \in [1, m]$

L'algorithme OTL cherche à changer itérativement la répartition des listes L_i sur les différents caractères du message (sans modifier le contenu des listes) de telle manière que la différence entre le cardinal de la nouvelle liste affectée à chaque caractère c_i et le cardinal de la liste L_i d'origine soit maximale [Omar, 2006]. Cette variation, en terme de répartition, est assurée grâce aux opérateurs génétiques choisis (MPX pour le croisement avec un taux compris entre 60% et 100%, et une simple permutation pour la mutation avec un taux compris entre 0.1% et 5%). Et pour juger la pertinence des individus construits, la fonction

d'évaluation utilisée est celle donnée ci-dessous. Les meilleurs individus sont ensuite sélectionnés par le biais de la méthode classique de la roulette, en vue de se reproduire.

$$F(X_j) = \sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right|$$

Le processus évolutionnaire est répété jusqu'à la satisfaction d'un critère d'arrêt exprimé à l'aide de la fonction F. Cette dernière est bornée car : $0 \le F(X) \le 2 * l$ (l est la taille du message) pour tout individu X. En fait [Omar, 2006] :

$$\sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right| \le \sum_{i=1}^{m} (card(L_{j_i}) + card(L_i)) \le 2 * l$$

L'opération de déchiffrement, quand à elle, se fait en deux étapes :

1) Tout d'abord, le texte chiffré est représenté suivant le codage proposé en une suite de listes de positions, et c'est grâce à la clé génétique que les caractères vont retrouver leurs listes de position correspondantes dans le message en clair. En effet, la clé, qui peut être d'un usage symétrique ou asymétrique, est une permutation de {1,2,…, m}. Comme résultat, nous obtenons le message M_0'.
2) La deuxième étape, correspond au déchiffrement du message M_0' pour obtenir M_0.

I.4. Conclusion

Dans ce chapitre, nous avons présenté les différentes catégories de cryptographie depuis sa première apparition jusqu'à nos jours. Un état de l'art aussi riche que possible sur les plus célèbres algorithmes de chiffrement, ainsi les fameuses ruses des cryptanalystes ont été exposées. D'après cette étude, un système cryptographique est considéré comme sûr si personne n'a encore mis en défaut sa sécurité. Nous avons vu que ni la cryptographie classique ni symétrique n'a pu s'assurer ce besoin dû aux leurs inconvénients. Pour résoudre cela, les cryptographes ont cherché à déplacer la difficulté ; plutôt que d'utiliser de simples substitutions et de faire reposer la sécurité sur le nombre de clés possibles, ils ont essayé de faire reposer la sécurité sur des difficultés calculatoires. Cela a donné naissance aux algorithmes de cryptographie asymétrique. Cependant il s'avère que l'augmentation constante de la puissance de calcul de nos machines nécessite d'augmenter constamment la difficulté de nos algorithmes. Il se peut en effet qu'une nouvelle technologie anéantisse toute la difficulté d'un algorithme. La cryptographie hybride est aussi un sujet d'attaque puisqu'elle n'est qu'une combinaison de la cryptographie symétrique et asymétrique. Une nouvelle tendance basée sur la cryptographie quantique a aussi été développée. Cette cryptographie est sûre, néanmoins elle est basée sur des principes beaucoup plus théorique et lourds.

Enfin, à partir de la richesse et de la variété des modèles mathématiques exploités dans le domaine de la cryptographie, de nouvelles approches cryptographiques peuvent être proposées par exploitation de méthodes approchées. Le précédent chapitre présentera de telles méthodes de résolution de problèmes : les métaheuristiques.

CHAPITRE II

METAHEURISTIQUES

II.1. Introduction

Un très grand nombre de méthodes existent en RO et en IA pour résoudre différentes sortes de problèmes tels que les problèmes d'optimisation combinatoire. D'une manière très générale, les méthodes de résolution suivent quatre approches différentes pour la recherche d'une solution [JinK, 1999] : l'approche de construction, l'approche de relaxation, l'approche de voisinage et l'approche d'évolution. Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative : par exemple, la méthode de Lin et Kernighan qui resta longtemps le champion des algorithmes pour le problème du voyageur de commerce [LinS, 1973].

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques* [Reev, 1993], [Aart, 1997]. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [Lapo, 1996], [Osma, 1996]. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi, les métaheuristiques sont conçues pour résoudre des problèmes d'optimisation complexes où d'autres méthodes d'optimisation ne sont pas efficaces. Ces méthodes ont fini par être reconnues comme l'une des approches les plus pratiques pour résoudre des problèmes nombreux et complexes, et ceci est particulièrement vrai pour les divers problèmes du monde réel qui sont de nature combinatoire. L'avantage pratique de métaheuristiques réside dans leur efficacité et leur application générale. Dans la littérature et au début des recherches, des heuristiques spécialisées ont été généralement développées pour résoudre des problèmes complexes d'optimisation combinatoire.
D'autre part, avec l'émergence de stratégies des solutions plus générales, y compris les métaheuristiques telles que la recherche taboue, algorithmes génétiques, recuit simulé, le principal défi est devenu l'adaptation des méta-heuristiques à un problème particulier ou une catégorie de problème. Cela nécessite généralement beaucoup moins de travail que de développer une heuristique spécialisée pour une application spécifique, ce qui rend les métaheuristiques un choix attrayant pour la mise en œuvre dans les logiciels à usage général. En outre, une bonne mise en œuvre de métaheuristique est susceptible de fournir des solutions quasi optimales en temps de calcul raisonnables.

Ainsi, une métaheuristique est définie de manière similaire à une heuristique (qui est une méthode conçue pour un problème d'optimisation donné et qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème), mais à un niveau d'abstraction plus élevé (d'après E. Taillard).

II.2. Principes

Les métaheuristiques sont apparues dans les années 80. Ce sont des méthodes d'optimisation, de type stochastique, conçus pour les problèmes difficiles. Des progrès importants ont été réalisés avec l'apparition de nouvelles générations de méthodes approchées puissantes et générales, souvent appelées métaheuristiques [Reev, 1993] [Aart, 1997].

Le terme « métaheuristique » a été initialement utilisé par F. Glover pour distinguer la méthode tabou des heuristiques spécifiques [Glov, 1986]. Notons que ce terme est également utilisé par J-L. Laurière dans son système de résolution Alice [Laur, 1978].

L'origine du mot méta heuristique nous aide à comprendre sa signification. En grec: « Méta » signifie « au-delà », ou « au niveau supérieur », et « Heuristique» veut dire « Trouver ». Ainsi, l'interprétation de ces mots peut signifier que l'on effectue des recherches à un très haut niveau et que la procédure algorithmique regroupe plusieurs heuristiques.

D'après Glover, metaheuristique "se réfère à une stratégie maître qui guide et modifie d'autres heuristiques pour produire des solutions au-delà de ceux qui sont normalement générés dans une quête d'optimalité locale" [Glov, 1997].

Selon [Glov, 1997] « métaheuristiques dans leurs formes modernes sont basées sur une variété d'interprétations de ce qui constitue une recherche intelligente », où le terme de recherche intelligente a été mis en évidence par Pearl [Poire, 1984] (en ce qui concerne heuristiques dans un contexte de l'intelligence artificielle) et [vobs, 1993] (en ce qui concerne le contexte de la recherche opérationnelle). Dans ce sens, on peut aussi considérer la définition suivante: « Une métaheuristique est un processus à générations itératives qui guide

une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter les espaces de recherche en utilisant des stratégies lesarning pour structurer l'information afin de trouver des solutions quasi-optimales » [Osma, 1996]

Pour résumer, la définition suivante semble être la plus appropriée : « Une métaheuristique est un processus maître itératif qui guide et modifie les opérations d'heuristiques subordonnées pour produire efficacement des solutions de haute qualité. Il peut manipuler une solution unique complète (ou incomplète) ou un ensemble de solutions à chaque itération. Les heuristiques subordonnées peuvent être des procédures de niveau élevé (ou faible), ou une recherche locale simple, ou tout simplement une méthode de construction. La famille de métaheuristiques comprend, mais n'est pas limitée à, des procédures de mémoire d'adaptation, de recherche tabou, des systèmes de fourmis, avides de recherche randomisée d'adaptation, de recherche à voisinage variable, des méthodes évolutionnaires, des algorithmes génétiques, de recherche de points, des réseaux neuronaux, de recuit simulé, et leurs hybrides. " [VobS, 1999]

Le but des métaheuristiques est de minimiser ou de maximiser une, ou plusieurs fonctions objectives. Comme exemple, on peut demander de trouver le plus court chemin entre un point A et un point B, sachant que des obstacles sont présents donc il s'agit d'un problème de minimisation ; ou si on veut trouver comment effectuer le plus grand nombre de tâches en un temps limité alors c'est la maximisation du travail à faire.

L'évolution des métaheuristiques se fait de manière itérative. Ceci a l'avantage de permettre l'arrêt de l'algorithme quand on le souhaite, et de récupérer la meilleure solution trouvée jusqu'à présent sans être obligé d'attendre la fin de l'exécution. Un autre point important des métaheuristiques est qu'elles font évoluer des solutions en les améliorant à chaque itération.

Donc, les métaheuristiques sont généralement des algorithmes itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction. Les itérations successives permettent d'évoluer d'une solution peu satisfaisante à la solution la plus adaptée. Les métaheuristiques se comportent, donc, comme des algorithmes de recherche tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution.

Les métaheuristiques sont souvent inspirées de systèmes naturels, qu'ils soient pris en physique -cas du recuit simulé-, en biologie de l'évolution -cas des algorithmes génétiques- ou encore en éthologie -cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires.
Les métaheuristiques désignées par M sont, souvent, des algorithmes utilisant un échantillonnage probabiliste. Elles tentent de trouver l'optimum global (G) d'un problème d'optimisation difficile (avec des discontinuités (D), par exemple), sans être piégées par les optima locaux (L). Ceci est indiqué à la figure suivante :

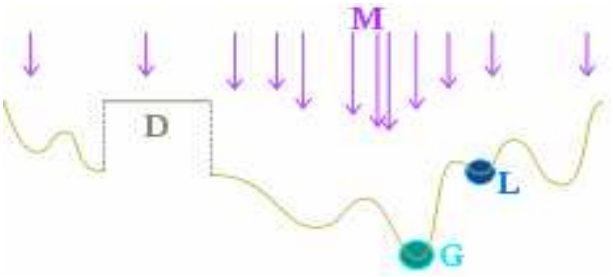


Figure II.1. *Principe général des métaheuristiques.*

II.3. Organisation d'une métaheuristique

D'une manière générale, les métaheuristiques s'articulent autour des principales notions suivantes :

II.3.1. Le Voisinage

Le voisinage d'une solution est un sous-ensemble de solutions qu'il est possible d'atteindre par une série de transformations données.
Formellement parlant [JinK, 1999], soit X l'ensemble des configurations admissibles d'un problème, on appelle *voisinage* toute application $N : X \to 2^X$. On appelle *mécanisme d'exploration* du voisinage toute procédure qui précise comment la recherche passe d'une configuration $s \in X$ à une configuration $s' \in N(s)$. Une configuration *s* est un *optimum local* par rapport au voisinage *N* si $f(s) \leq f(s')$ pour toute configuration $s' \in N(s)$.

Une méthode typique de voisinage débute avec une configuration initiale, et réalise ensuite un processus itératif qui consiste à remplacer la configuration courante par l'un de ses voisins en tenant compte de la fonction de coût. Ce processus s'arrête et retourne la meilleure configuration trouvée quand la condition d'arrêt est réalisée.

Un des avantages de cette stratégie de recherche réside précisément dans la possibilité de contrôler le temps de calcul : la qualité de la solution trouvée tend à s'améliorer progressivement au cours du temps et l'utilisateur est libre d'arrêter l'exécution au moment qu'il aura choisi. Dans la littérature, plusieurs voisinages ainsi que plusieurs stratégies de parcours de ces voisinages ont été définies [Gold, 1989], [Hert, 2003].

II.3.2. Diversification, intensification et apprentissage

La *diversification* (ou *exploration*, synonyme utilisé presque indifféremment dans la littérature des algorithmes évolutionnaires) désigne les processus visant à récolter de l'information sur le problème optimisé. L'*intensification* (ou *exploitation*) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Autrement dit, l'intensification insiste sur la capacité d'examiner en profondeur par une méthode des zones de recherche particulières alors que la diversification met en avant la capacité de découvrir des zones de recherche prometteuses.

La *mémoire* est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi les optima locaux.

Les métaheuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage, ou en mêlant ces notions de façon plus

étroites. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

Les notions d'intensification et de diversifications sont prépondérantes dans la conception des métaheuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherches. En effet, l'application systématique du seul principe d'exploitation ne permet pas une recherche efficace. En effet, l'exploitation conduit à confiner la recherche dans une zone limitée qui finit par s'épuiser. Le cas de l'amélioration itérative rapidement piégée dans un optimum local illustre cruellement ce phénomène. Une autre illustration souvent évoquée est fournie par le problème de convergence prématurée des algorithmes génétiques : du fait de la sélection, la population finit par n'être constituée que d'individus tous similaires. L'une des préoccupations majeures dans les algorithmes génétiques consiste d'ailleurs à préserver le plus longtemps possible une diversité suffisante dans la population. Face à ce type de difficulté, la solution consiste à diriger la poursuite de la recherche vers de nouvelles zones, *i.e.*, à recourir à l'exploration. Les deux notions ne sont, donc, pas contradictoires mais complémentaires, et il existe de nombreuses stratégies mêlant à la fois l'un et l'autre des aspects.

II.4. Méthodes générales et méthodes spécifiques

Contrairement aux algorithmes traditionnels dédiés à un problème spécifique, les métaheuristiques constituent des mécanismes très généraux qui peuvent être adaptés pour traiter de nombreux problèmes différents. Pour expliquer l'efficacité d'un algorithme spécifique, on invoque souvent le fait qu'il utilise des connaissances spécifiques du problème. Pour expliquer l'efficacité d'une métaheuristique, deux types d'arguments opposés sont généralement avancés.

Selon certains, des mécanismes généraux suffisamment puissants ont par eux mêmes la faculté de mener efficacement la recherche sans disposer d'information spécifique du problème considéré (contexte de boîte noire) : c'était notamment le point de vue classique porté sur les algorithmes génétiques [Gold, 1989]. Malheureusement, de même qu'il ne peut pas exister de stratégie avantageuse dans un jeu de hasard comme la roulette, il existe également des limitations théoriques fondamentales qui ruinent les espoirs d'une méthode aveugle dans le cas le plus général. En fait, le théorème « *No Free Lunch (NFL)* » [Wolp, 1997] montre que pour les problèmes de type boîte noire, toutes les méthodes sont équivalentes et font aussi bien, ou plutôt aussi mal, que l'énumération aléatoire.

Selon le point de vue opposé, la puissance d'une métaheuristique est d'abord liée à son aptitude à intégrer des connaissances spécifiques du problème. La connaissance du problème la plus fondamentale réside dans le codage du problème et dans le choix de la fonction de voisinage. Plusieurs auteurs insistent sur l'importance du codage du problème, voir par exemple [Radc, 1995]. Dans le cas du voyageur de commerce, plusieurs types de codages différents ont été proposés et conduisent à des performances très variées [Mich, 1992]. En général, il n'existe pas de codage universellement efficace. Un « bon » codage doit permettre de restreindre l'espace de recherche et d'intégrer des connaissances du problème.

Les métaheuristiques tentent également d'améliorer leur efficacité en incorporant des connaissances supplémentaires dans leurs opérateurs. Plus les opérateurs d'une méthode utilisent des connaissances spécifiques, plus la méthode dispose de moyens potentiels pour conduire efficacement la recherche. En contrepartie, l'intégration de ces connaissances spécifiques (en supposant que ces connaissances soient disponibles) nécessite un effort pour spécialiser ou adapter la méthode. La méthode tabou vise à incorporer le plus possible de connaissances du problème pour atteindre le maximum d'efficacité : l'utilisateur doit notamment définir de façon pertinente le type de caractéristique figurant dans la liste tabou.

De leur coté, les algorithmes génétiques se sont éloignés du modèle standard pour intégrer également des connaissances du problème : codage non binaire, opérateurs spécifiques. Au contraire, le recuit simulé est parfois présenté comme une méthode facile à adapter à un problème et qui ne tente pas d'exploiter de connaissances spécifiques.

En général, une méthode offrant des possibilités d'intégrer des connaissances du problème a plus de chance de produire de bons résultats, mais demande un effort d'adaptation et de spécialisation. Au contraire, une méthode très générale qui prétend n'intégrer aucune connaissance propre ne peut pas être compétitive.

II.5. Classification des métaheuristiques

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents.
Dans la littérature, plusieurs classifications de métaheuristiques ont été proposées. Nous présentons à ce niveau un aperçu des principales classifications.

II.5.1. Les métaheuristiques inspirées et non inspirées d'un phénomène naturel

Une manière intuitive de classifier les métaheuristiques consiste à séparer celles qui sont inspirées d'un phénomène naturel, de celles qui ne le sont pas.

Les algorithmes génétiques ou les algorithmes par colonies de fourmi entrent clairement dans la première catégorie, tandis que la méthode de descente, ou la recherche Tabou, vont dans la seconde.

II.5.2 Les métaheuristiques avec fonction objective statique ou dynamique

Les métaheuristiques peuvent être aussi classées selon la façon dont ils utilisent la fonction objective. Certains algorithmes conservent la fonction objective donnée dans la représentation de problème telle qu'elle est (comme la recherche locale guidée (GLS)) et la modifie lors de la recherche. L'idée de cette approche est d'éviter les minimums locaux en modifiant l'espace de recherche. En conséquence, lors de la recherche la fonction objective est altérée en essayant d'intégrer des informations intégrées au cours du processus de recherche.

II.5.3. Les métaheuristiques avec une ou plusieurs structures de voisinage

La plupart des métaheuristiques travaillent sur une seule structure de voisinage. En d'autres termes, la topologie du paysage de fitness ne change pas en cours de l'algorithme. D'autres métaheuristiques telles que la recherche par voisinage variable utilisent un ensemble de structures de voisinage qui donne la possibilité de diversifier la recherche en échangeant entre les structures de voisinage qui ont des formes différentes.

II.5.4. Les métaheuristiques avec et sans mémoire

Les métaheuristiques utilisent l'historique de leur recherche pour guider l'optimisation aux itérations suivantes. Dans le cas le plus simple, elles se limitent à considérer l'état de la recherche à une itération donnée pour déterminer la prochaine itération : il s'agit alors d'un processus de décision markovien, et on parlera de méthode *sans mémoire*. C'est le cas de la plupart des méthodes de recherche locale (recherche à voisinage variable, recherche locale

itérée, recherche locale stochastique, recherche locale guidée), algorithme d'estimation de distribution, recuit simulé, GRASP.

Beaucoup de métaheuristiques utilisent une mémoire plus évoluée, que ce soit sur le court terme (solutions visitées récemment, par exemple) ou sur le long terme (mémorisation d'un ensemble de paramètres synthétiques décrivant la recherche).

II.5.5. Les métaheuristiques implicite, explicite, directe

En considérant les métaheuristiques comme des méthodes itératives utilisant un échantillonnage de la fonction objective comme base d'apprentissage (définition plus particulièrement adaptée aux métaheuristiques à populations) apparaît le problème du choix de l'échantillonnage.

Dans la très grande majorité des cas, cet échantillonnage se fait sur une base aléatoire, et peut donc être décrit via une distribution de probabilités. Il existe alors trois classes de métaheuristiques, selon l'approche utilisée pour manipuler cette distribution.

La première classe est celle des méthodes *implicites*, où la distribution de probabilité n'est pas connue *a priori*. C'est le cas par exemple des algorithmes génétiques, où le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. L'évolution différentielle, Scatter search, algorithmes à essaim de particules, stratégies d'évolution et algorithmes de colonie de fourmis sont des méthodes implicites.

Par opposition, on peut donc classer les méthodes *explicites*, qui utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution.

Dans cette classification, le recuit simulé occupe une place particulière, puisqu'on peut considérer qu'il échantillonne la fonction objective en utilisant directement celle-ci comme distribution de probabilité (les meilleures solutions ayant une probabilité plus grande d'être tirées). Il n'est donc ni explicite ni implicite, mais plutôt « direct » (exemple : le recuit simulé) [JinK, 1999].

II.5.6. Les métaheuristiques évolutionnaires et non évolutionnaires

On trouve parfois une classification présentant les algorithmes d'optimisations stochastiques comme étant « évolutionnaires » (ou « évolutionnistes ») ou non. L'algorithme sera considéré comme faisant partie de la classe des algorithmes évolutionnaires s'il manipule une population *via* des *opérateurs*, selon un algorithme général donné.

Cette façon de présenter les métaheuristiques dispose d'une nomenclature adaptée : on parlera d'opérateurs pour toute action modifiant l'état d'une ou plusieurs solutions. Un opérateur construisant une nouvelle solution sera dénommé *générateur*, alors qu'un opérateur modifiant une solution existante sera appelé *mutateur*.

Dans cette optique, la structure générale des algorithmes évolutionnaires enchaîne des étapes de *sélection*, de *reproduction* (ou *croisement*), de *mutation* et enfin de *remplacement*. Chaque étape utilise des opérateurs plus ou moins spécifiques.

Quelques algorithmes évolutionnaires : l'algorithme génétique, la programmation génétique, l'algorithme d'évolution différentielle, les stratégies évolutionnaires et l'algorithme d'estimation de distribution [JinK, 1999].

II.5.7. Les métaheuristiques à base de population et les métaheuristiques à trajectoire

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

Les plus connues dans cette classe sont le recuit simulé, la recherche tabou, la recherche à voisinage variable, la méthode GRASP. L'autre approche utilise la notion de population. La métaheuristique manipule un ensemble de solutions en parallèle, à chaque itération. On peut citer les algorithmes génétiques, l'optimisation par essaims particulaires et les algorithmes de colonies de fourmis.

La frontière est parfois floue entre ces deux classes. On peut ainsi considérer qu'un recuit simulé où la température baisse par paliers, a une structure à population. En effet, dans ce cas on manipule un ensemble de points à chaque palier, il s'agit simplement d'une méthode d'échantillonnage particulière.

II.5.7.1. Les métaheuristiques à trajectoire (à solution unique)

Ici, nous résumons les plus connues des métaheuristiques à trajectoire.

a. La méthode de descente

Le principe de la méthode de descente (dite aussi *basic local search*) consiste, à partir d'une solution *S*, à choisir une solution *S'* dans un voisinage de *S* telle que *S' améliore* la recherche (généralement telle que $f(S') < f(S)$). On peut décider soit d'examiner toutes les solutions du voisinage *N* et prendre la meilleure de toutes ces solutions (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.
La méthode de descente est la méthode la plus élémentaire de recherche locale. On peut la formaliser comme suit :

Algorithme II.1 *descente_simple* (solution initiale *S*)

Début
  Répéter
    Choisir *S'* dans N(*S*)
    Si $f(S') < f(S)$ alors $S \leftarrow S'$
  Jusqu'à ce que $f(S') \geq f(S)$, $\forall S' \in$ **S**
Fin

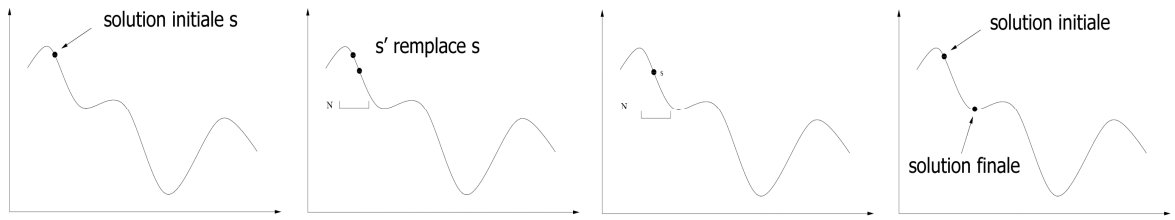


Figure II.2. *Evolution d'une solution dans la méthode de descente.*

On peut varier cette méthode en choisissant à chaque fois la solution S' dans N(S) qui améliore le plus la valeur de f. C'est la méthode de plus grande descente [Cost, 2006].

b. Recherche aléatoire (Random Search : RS)

C'est la plus simple des méthodes stochastiques. Cette méthode consiste à tirer à chaque itération une solution au hasard. La fonction objective *f* est évaluée en ce point. La nouvelle valeur est comparée à la précédente. Si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continu. Sinon on repart du point précédent et on recommence le procédé, jusqu'à ce que les conditions d'arrêt soient atteintes. L'algorithme II.2 [Luke, 2010] présente un pseudo code de la recherche aléatoire dans le cas d'un problème de minimisation.

Algorithme II.2 *Recherche aléatoire*

Début
$S_0 \leftarrow$ solution aléatoire;
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$;
Répéter
S ← solution aléatoire ;
Si ($f(S) < f_{min}$) *Alors*
$f_{min} \leftarrow f(S)$;
$x_{min} \leftarrow S$;
Fin Si
Jusqu'à conditions d'arrêt satisfaites.
Fin

c. Hill Climbing (HC)

On part d'une solution si possible "bonne" (par exemple donnée par une heuristique) et on balaie l'ensemble des voisins de cette solution ; s'il n'existe pas de voisin meilleur que notre solution, cela veut dire qu'un optimum local a été trouvé et on arrête la recherche. Sinon, on choisit le meilleur des voisins et on recommence. Une autre implémentation consiste non pas à passer au meilleur des voisins à chaque étape, mais au premier meilleur voisin trouvé. La convergence vers un optimum local pouvant être très lente, on peut éventuellement fixer un nombre de boucles maximum, si on veut limiter le temps d'exécution [Sea, 2010].

Cette méthode a l'inconvénient de rester bloquée dans un optimum local : une fois un optimum local trouvé, on s'arrête, même si ce n'est pas l'optimum global. Selon le paysage des solutions, l'optimum local peut être très bon ou très mauvais par rapport à l'optimum global. Si la solution de départ est donnée par une heuristique déterministe, l'algorithme sera déterministe. Si elle est tirée au hasard, l'algorithme devient non déterministe et donc plusieurs exécutions différentes sur la même instance pourront donner des solutions différentes et de qualités différentes [Sea, 2010].

La notion de voisinage est primordiale. Si les voisins sont très nombreux, on a de fortes chances de trouver l'optimum global, mais visiter un voisinage peut être très long: on visitera une grande partie de l'espace des solutions. Si le voisinage est très restreint, on risque fort de rester bloqué dans un optimum local de "mauvaise qualité". Le choix de la notion de voisinage est un compromis entre efficacité et qualité.

Le pseudo-code de l'algorithme Hill Climbing est celui présenté par l'algorithme II.3 [Sea, 2010]:

Algorithme II.3 *HILL CLIMBING*

Début
$S_0 \leftarrow$ Solution aléatoire
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$
Répéter
Engendrer un N-échantillon $S_i(x)$ voisinage de S(x) *ET* calculer
$f(S(x)) = \min[f(S_i(x))]$ avec $1 \leq i \leq N$

Si ($f(S) < f_{min}$) *then*
$x_{min} \leftarrow S$
Sinon
Sortir de Répéter
Fin Si
Jusqu'à condition d'arrêt satisfaite.
Fin.

d. Recuit Simulé

Le recuit simulé a été introduit et mis au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985. Il a été repris sous différentes formes par Lawrence Davis en 1987 [Lawr, 1987].
Cet algorithme a été dérivé de l'algorithme de Metropolis, développé par les scientifiques du projet ex-Manhattan : Nicholas Metropolis, Arianna et Marshall Rosenbluth, Augusta et Edward Teller en 1953. Ce dernier permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. Partant d'une solution donnée, en la modifiant, on en obtient une seconde. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

La description des phénomènes physiques et quantiques liés au processus de recuit s'appuie sur la statistique de Boltzmann. Pour qu'un métal ait une qualité optimale, il faut que son état d'énergie soit minimal. Pour améliorer la qualité d'un métal, on le chauffe, puis on le refroidit par paliers en attendant à chaque fois que l'état d'énergie soit stabilisé. Au niveau de l'atome, cela signifie qu'à une température chaude, les atomes bougent beaucoup, et en se

refroidissant, ils trouvent leur place optimale. Plus la température descend, moins les atomes bougent et le métal devient de plus en plus résistant.

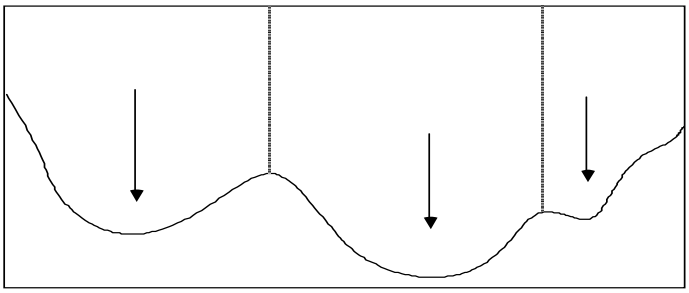


Figure II.3. *Un paysage d'énergie. Suivant la configuration initiale, indiquée par les flèches, la dynamique aboutit à température nulle dans l'un quelconque des minima relatifs, séparés par les barrières indiquées en pointillés. A température élevée, les processus probabilistes permettent au système de sauter les barrières séparant les vallées.*

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. À cette solution correspond une énergie initiale $E = E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T = T_0$ élevée est également choisie. Ce choix est alors totalement arbitraire et va dépendre de la loi de décroissance utilisée.

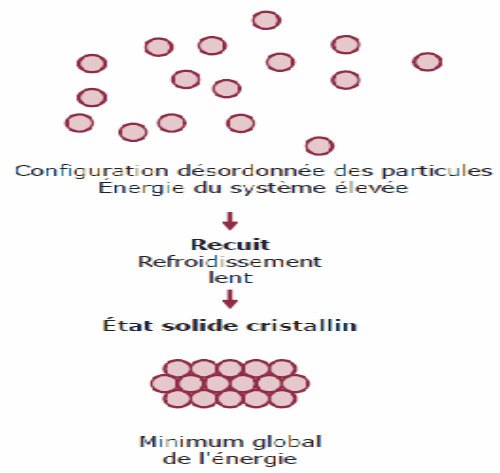


Figure II.4. *Transposition de procédé recuit à la résolution d'un problème d'optimisation.*

À chaque itération de l'algorithme une modification élémentaire de la solution est effectuée. Cette modification entraîne une variation Δ_E de l'énergie du système (toujours calculée à partir du critère que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta_E}{T}}$. Ce choix de l'exponentielle pour la probabilité s'appelle règle de Metropolis.

On itère ensuite selon ce procédé en gardant la température constante.

Algorithme II.4 *Recuit simulé*

```
Début
    Initialisation du temps
    Tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint] Faire
        Initialisation de la température à Tmax
        Tant que [la température est supérieure à 0] et [le temps limite n'est pas atteint] Faire
            Incrémentation du temps
            Modification de la solution courante
            Si [la nouvelle solution est meilleure que l'ancienne] Alors
                la sauvegarder
            Sinon
                Si [le système est stable depuis un nombre défini d'itérations] Alors
                    Diminuer la température
                Fin si
            Fin si
        Fin tant que
    Fin tant que
Fin
```

L'algorithme varie de Hill-Climbing (algorithme 3) dans sa décision du moment de remplacer S, la solution candidate d'origine, avec R, son enfant nouveau peaufiné. Plus précisément, si R est meilleur que S, nous allons toujours remplacer S par R comme d'habitude. Mais si R est pire que S, on peut toujours remplacer S par R avec une certaine probabilité P(t, R, S):

$$P(t, R, S) = e^{\frac{\text{Quality}(R) - \text{Quality}(S)}{t}}$$

Où t> 0. Cette équation est intéressante à deux égards. Notez que la fraction est négative car R est pire que S. Premièrement, si R est bien pire que S, la fraction est plus grande, et donc la probabilité est proche de 0. Si R est très proche de S, la probabilité est proche de 1. Ainsi, si R n'est pas bien pire que S, nous allons toujours choisir R avec une probabilité raisonnable.

Deuxièmement, nous avons un paramètre ajustable t. Si t est proche de 0, la fraction est de nouveau un grand nombre, et donc la probabilité est proche de 0. Si t est élevé, la probabilité est proche de 1. L'idée est d'abord de mettre t comme un grand nombre, ce qui provoque l'algorithme de se déplacer à chaque solution nouvellement créée indépendamment de sa qualité. Nous faisons une marche aléatoire dans l'espace. Puis t diminue lentement, éventuellement à 0, à quel point l'algorithme ne fait rien de plus que de simples Hill-Climbing.

e. Recherche Tabou (Tabu Search : TS)

La recherche avec tabou, ou simplement dite recherche tabou, est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans un article paru en 1986 [Glov, 1986], mais reprenant de nombreuses idées proposées antérieurement dès les années 60 dans les deux articles simplement intitulés « Tabu chearch » [Glov, 1989] proposant la plus part des principes de cette recherche telle qu'elle est décrite actuellement. Maintenant, elle est d'usage très classique en domaine d'optimisation combinatoire pour résoudre les problèmes NP-durs.

L'idée principale de la recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui optimise la fonction objective. Le voisinage choisi est exploré de manière déterministe, il est interdit de reprendre des solutions récemment visitées.

L'un des principes fondamentaux de cette recherche qui la caractérise des autres méta-heuristiques est la construction d'un historique de la recherche itérative ou, ce qui est équivalent, au fait de doter la recherche de mémoire [Glov, 1997]. La recherche tabou examine un échantillonnage de solutions *N(S)* et retient la meilleure solution *S* des solutions obtenues.

A chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche tabou qu'elle est une méthode agressive [www10].

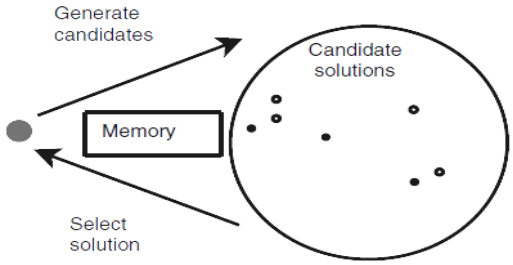
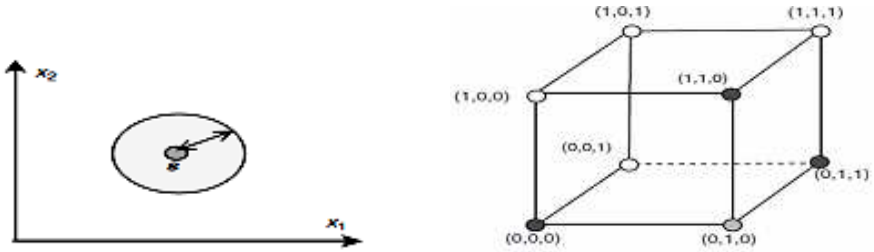


Figure II.5. *Principe de base d'une méta-heuristique à mémoire.*

La recherche tabou est essentiellement axée sur une exploration non triviale de l'ensemble des solutions en utilisant la notion de voisinage. Formellement pour toute solution *s* de *S*, un ensemble de $N(s) \in S$ qu'on appelle ensemble des solutions voisines de *s*.



(a) *(b)*

Figure II.6. *Les types de solutions du voisinage.*
(a) *Le voisinage pour un problème à variables continus : Le cercle symbolise les voisins de la solution s,*
(b) *Le voisinage pour un problème à variables discrètes : Les nœuds du cube sont les solutions et leurs voisins adjacents.*

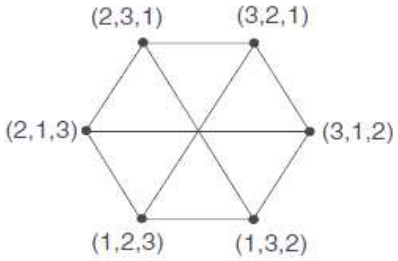


Figure II.7. *Voisinage d'une permutation de taille égale à 3.*
Les voisins de la solution (2.3.1) sont : (3.2.1), (2.1.3), et (1.3.2).

Vue d'un côté pratique l'ensemble N(s) des solutions voisines de s n'est autre que l'ensemble des modifications que l'on peut apporter à cette dernière. On appelle *mouvement* une modification apportée à une solution.

Dans la littérature et dans le cas de la recherche tabou, l'application d'un mouvement *m* à une solution *s* est notée ; d'où l'expression suivante du voisinage :

$$N(s) = \{s' / s' = s \oplus m, m \in M\}$$

La figure II.8 [Kamm, 2006] illustre l'ensemble des mouvements possibles dans le cas d'une solution composée de quatre éléments.

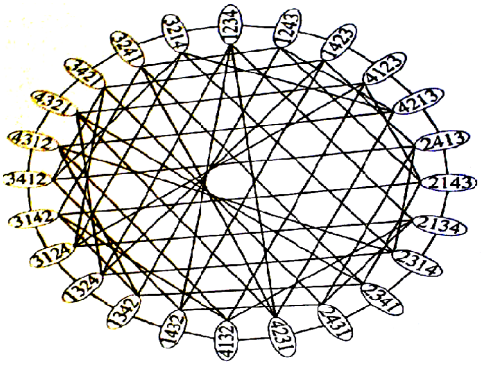


Figure II.8. *Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments.*

La recherche tabou est une méthode itérative qui explore un espace de solutions. Comme toute méthode itérative, il est intéressant d'éviter de revisiter des solutions pour essayer de converger vers la solution optimale le plus rapidement possible et ceci n'est possible qu'à l'aide d'une mémoire caractérisée par une taille définie par l'utilisateur et le problème à résoudre. Cependant, l'utilisation d'une mémoire peut s'avérer dans bien des cas peu efficace, voire mauvaise. Outre le fait de mettre cette idée en pratique est difficile, cela suppose que l'on doit mémoriser chaque solution visitée et de tester à chaque itération et pour chaque solution admissible si cette dernière a été déjà examinée. Les *tables de hachage* permettent de faire cela de manière efficace, mais ceci n'empêche pas la croissance linéaire de la taille de la mémoire avec le nombre des itérations effectuées.

Une table de hachage est définie par un tableau *T* de *J* entiers, avec *J* choisi relativement grand et assez voisin de la capacité de la machine utilisée tout en prenant en considération la taille du problème.

Le but de la table de hachage est :

- ✓ d'interdire le retour aux solutions obtenues pendant les *t* dernières opérations ;
- ✓ ainsi éliminer les cycles de longueurs égales ou inférieures à *t*.

Toutefois, le fait d'interdire un grand nombre de mouvements aura pour conséquence de rendre l'obtention de bons résultats très délicate faute de mouvements disponibles. Si ce nombre diminue, cela augmentera les chances d'une meilleure exploration aux alentours des optimums locaux et d'obtenir ainsi de meilleures solutions. Il ne faut cependant pas trop diminuer ce nombre, car, dans ce cas, il devient très probable de se trouver prisonnier d'un ensemble très restreint de solutions et de les visiter tout le temps de la recherche.

Pour palier à ce compromis du choix de nombre des mouvements à interdire et de bénéficier simultanément des avantages d'un petit nombre, qui permet une visite détaillée du voisinage d'une même solution, et d'un grand nombre qui permet de franchir le voisinage de différentes solutions, ce nombre doit varier au cours du processus itératif. Pour ce faire, plusieurs méthodes existent. Ce nombre peut être tiré au hasard à partir d'un intervalle donné à chaque itération ou après un nombre d'itérations, comme il peut aussi croître ou décroître en fonction des résultats obtenus au cours de la recherche [Tail, 1991], [Tail, 1995] et [Tail, 1999].

Autrement dit, une liste tabou avec trop d'éléments peut devenir très restrictive. En effet, il a été observé que trop de contraintes tabou forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération. Par contre une liste tabou contenant trop peu d'éléments peu s'avérer inutile et mener à des mouvements cycliques [Ayas, 2004].

En plus, dans beaucoup de problèmes, l'interdiction de revisiter des solutions mènerait à des incohérences comme la déconnection de la solution courante de la solution optimale ou bien le blocage de la recherche itérative par cause d'absence de solutions voisines non visitées[Kamm, 2006].

La figure suivante illustre ces derniers cas incohérents :

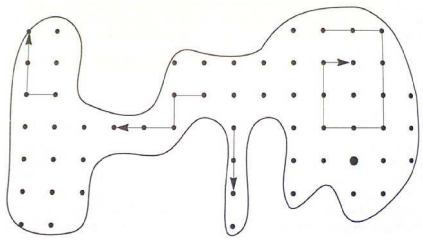


Figure II.9. *Déconnections et blocages dans la recherche tabou.*

Il est à signaler qu'une liste tabou peut être soit statique ou dynamique. Pour la première, la taille de la liste est fixée, en générale au début. Elle dépend de la taille du problème à résoudre et du voisinage choisi. Pour la deuxième, la taille doit changer durant le processus de recherche sans utiliser aucune information sur la mémoire de recherche.

Algorithme II.5 *Recherche Tabou*

Début

Φ Fonction de coût
Variables locales : S solution courante, Liste Taboue L, Meilleure solution M, itération courante K, nombre d'itérations N
Paramétrages : taille de liste taboue, critère d'aspiration
Choix d'une solution initiale S_0
S← S_0
M← S
K ←0
Tant que K< N *faire*
K← K+1
Mise à jour de L
Génération des candidats E par opération de voisinage
C ← Best (E)
Si Φ (S) < Φ(M) *ou* C n'est pas taboue *ou* C vérifie l'aspiration *alors*
S ← C
Sinon
E ← E \ C
Fin tant que
Retourner S

Fin.

f. GRASP

À proprement parler, GRASP (pour *Greedy Randomized Adaptive Search Procedure*) est une méthode hybride, car elle cherche à combiner les avantages des heuristiques gloutonnes, de la recherche aléatoire et des méthodes de voisinage. Un algorithme GRASP répète un processus composé de deux étapes : la construction d'une solution suivie par une descente pour améliorer la solution construite. Durant l'étape de construction, une solution est itérativement construite : chaque itération ajoute un élément dans la solution partielle courante. Pour déterminer l'élément qui sera ajouté, on utilise une liste des meilleurs candidats obtenus avec une fonction gloutonne et on prend *au hasard* un élément dans cette liste. La liste des meilleurs candidats est dynamiquement mise à jour après chaque itération de construction. Cette étape de construction continue jusqu'à ce qu'une solution complète soit obtenue [JinK, 1999].

À partir de cette solution, une descente est appliquée pour améliorer la solution. Une procédure GRASP répète ces deux étapes et retourne à la fin la meilleure solution trouvée. Les deux paramètres de cette méthode sont donc la longueur de la liste de candidats et le nombre d'itérations autorisées.

Algorithme II.6 *GRASP*

```
Début
    x, x*, f* : type Données
    Début
    f* ← infini
    Pour i = 1 à nombre d'itérations faire
        x ← Construction Aléatoire Gloutonne ()
        x ← Recherche Locale(x)
        Si f(x) < f* alors
            x* ← x
            f* ←f(x)
        Fin Si
  Fin Pour
Fin.
```

II.5.7.2. Les métaheuristiques à population

Dans cette section, nous présentons un sommaire des plus connues des métaheuristiques à population.

a. Les algorithmes évolutionnaires

Les *algorithmes évolutionnistes* ou *algorithmes évolutionnaires* (*evolutionary algorithms* ou encore dits *evolutionary computation* en anglais), sont une famille d'algorithmes s'inspirant de la théorie de l'évolution pour résoudre des problèmes divers. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation, elles sont en cela des métaheuristiques, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. On les classe également parmi les méthodes d'intelligence calculatoire.

Selon la génétique et la théorie de l'évolution [www11] :

- ✓ Un enfant hérite son patrimoine génétique pour moitié de sa mère et pour moitié de son père (reproduction sexuée).
- ✓ Les enfants ne sont pas identiques aux parents car des altérations des gènes peuvent se produire (mutations).
- ✓ Parmi les mutations, certaines peuvent être favorables et d'autres défavorables.
- ✓ Il naît beaucoup de descendants : mais seuls les individus les mieux adaptés pourront survivre et transmettre leurs gènes à leur tour à leur descendance.

Dans ce modèle, on observe que [www11] :

- ✓ Le hasard joue un rôle moteur pour produire de nouveaux individus différents de leurs parents.
- ✓ La sélection naturelle effectue le tri entre les variations favorables et les autres.

Ainsi, basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin, ces algorithmes présentent des qualités intéressantes pour la résolution des problèmes d'optimisation. Les individus ou chromosomes d'un algorithme évolutionnaires sont des codages des solutions possibles du problème. Comme dans la nature, ces individus forment une population qui va évoluer dans le temps selon des lois de sélection qui vont favoriser les mieux adaptés à se croiser entre eux en produisant des populations meilleures. L'évolution des individus d'une population à une autre se fait à l'aide de la reproduction. Les individus parents vont se reproduire pour donner des individus enfants qui seront plus performants après avoir subis des opérations génétiques de croisement et de mutation.

Et comme ces reproductions se font avec une part de hasard par analogie à la nature, donc, les parents candidats à la reproduction sont choisis d'une manière probabiliste proportionnelle à leurs aptitudes et l'étape de reproduction est choisie d'une façon totalement aléatoire.

Finalement, passant d'une génération à une autre, les individus forment une progéniture plus performante qui s'approche au mieux de la solution optimale [Kamm, 2006].

La figure suivante résume le principe de résolution de problèmes par algorithmes évolutionnaires.

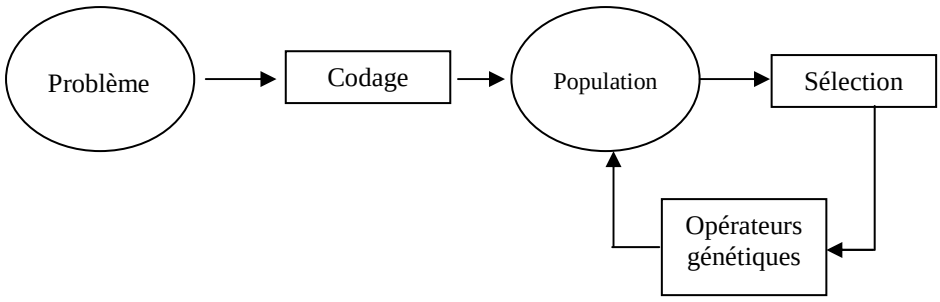


Figure II.10 . *Principe des algorithmes évolutionnaires.*

Ainsi, un algorithme évolutionnaire est construit autour des notions suivantes :

- **Le codage :**

La première étape de la résolution d'un problème à l'aide d'un algorithme génétique est la modélisation appropriée des solutions. Cette modélisation est appelée *codage* et permet de représenter les solutions sous forme de chromosomes.

Le codage se base sur deux notions importantes : le génotype et le phénotype.

- ✓ **Génotype** : représente l'ensemble des valeurs des gènes d'un chromosome.
- ✓ **Phénotype** : c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype.

S'il y a passage immédiat du phénotype au génotype, le codage est dit direct, sinon il est dit indirect et une procédure de passage est indispensable. Le codage d'une solution doit décrire toutes les données du problème et respecter toutes ses contraintes.

- **Population initiale :**

L'espace de recherche étant infini, il est très difficile de répartir plus ou moins équitablement la population initiale sur l'espace de recherche, de sorte que cet espace soit au maximum parcouru. Un certain nombre d'individus doivent être générés aléatoirement s'il n'existe aucune autre méthode d'initialisation.

- **Evaluation de la qualité d'une solution :**

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance (*fitness*) de mesurer cette efficacité pour permettre à l'algorithme de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution. Autrement dit, la fonction de performance *f*, doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre.

- **La sélection :**

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.
Plusieurs modes de sélection ont été définis tels que : sélection par roulette (wheel), sélection par rang, sélection steady-state, élitisme, sélection uniforme, etc.

- **Opérateurs génétiques :**

1) Croisement : C'est un opérateur génétique qui permet à deux chromosomes parents de produire deux chromosomes enfants où de nouvelles séquences de gènes pour les chromosomes enfants sont crées à partir d'une base de configuration des séquences héritées des chromosomes parents. Cet opérateur se produit selon une probabilité *Pc* fixée par l'utilisateur selon le problème à optimiser.

Il existe plusieurs manières d'effectuer un croisement soit par cross-over où l'on a besoin de deux parents qui génèrent à la fin du croisement deux enfants, soit par copie où l'on n'a besoin que d'un seul parent qui nous donnera à la fin du croisement un enfant qui est le parent lui-même (sa copie).

Dans la littérature, il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter [Mesg, 1999]. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que :

- ✓ le croisement à un point.
- ✓ le croisement multipoints.
- ✓ le croisement uniforme.

2) Mutation : permet de changer (permuter) un gène d'un chromosome par un autre d'une manière aléatoire, ce qui est à première vue très ressemblant avec un cross-over. La différence est que le cross-over essaie de converger vers une solution qui lui parait la meilleure (s'intéresse à la qualité), mais que la mutation permet la diversité. En quelque sorte, la mutation sert à éviter une convergence prématurée de l'algorithme. Par exemple, lors de la

recherche d'une solution optimum, la mutation sert à éviter la convergence vers un optimum local.
Cet opérateur est aussi appliqué avec une probabilité P_m. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche aléatoire et conserver le principe de sélection et d'évolution.

Dans la littérature plusieurs opérateurs de mutation ont été définis, tels que : la transposition de deux allèles consécutifs, la transposition de deux allèles quelconques, l'inversion d'allèles, etc.

Les algorithmes évolutionnaires se scindent en grandes familles, dont les principales sont :

1) Programmation évolutionnaire

La programmation évolutionnaire développée par *L.J. Fogel* [Foge, 2003], se base sur l'évolution d'une population d'automates à états finis (Figure II.11) pour résoudre des problèmes de prédiction. Ce modèle évolutionniste accentue l'utilisation de la mutation et n'utilise pas dans sa version originale la recombinaison des individus par croisement [Renn, 2000]. La table de transition des automates est modifiée grâce à des mutations aléatoires uniformes dans l'alphabet discret correspondant. Chaque automate de la population parente génère un enfant par mutation, et les meilleures solutions entre les parents et les enfants sont sélectionnées pour survivre, sachons que l'évaluation de la performance des individus correspond au nombre de symboles prédits correctement.

Par la suite, la programmation évolutionnaire a été développée et son domaine a été élargi par *D.B. Fogel*, afin qu'il puisse travailler dans l'espace réel, où la sélection déterministe est remplacée par un tournoi stochastique.

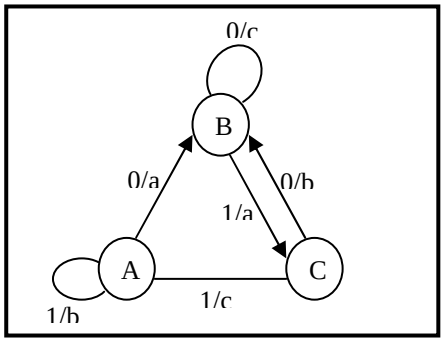


Figure II.11. *Exemple d'un automate à états finis ayant trois états différents S={A,B,C}, un alphabet d'entrée I={0,1}, et un alphabet de sortie O={a,b,c}. Chaque arrête entre deux états indique une transition possible, et la fonction de transition $\delta : S \times I \rightarrow S \times O$ est spécifiée par les labels au niveau des arrêtes ayant la forme i / o, signifiant que $\delta (s_i , i) = (s_j , o)$.*

2) Stratégies d'évolution

Les stratégies d'évolution sont dédiées à l'optimisation des problèmes continus dans l'espace de vecteurs des réels [Magn, 2001]. Les premiers efforts pour la mise en place des stratégies d'évolution ont eu lieu en 1973 à l'université de Berlin par Rechenberg [Rech, 2003] au cours de la résolution d'un problème aérodynamique. C'est avec ces méthodes évolutionnistes que la notion d'auto-adaptativité pour la mutation permettant de contrôler cette fonction de mutation, a été apparue. Une mise en œuvre de ce principe consiste à augmenter l'intensité de la mutation lorsque la proportion de descendants de bonne qualité, c'est à dire le nombre de mutation à succès, dépasse 20% de la population totale [Renn,

2000]. Elle est diminuée dans le cas opposé. Une interprétation possible de cette règle est la suivante : si la proportion de mutation réussie est élevée, l'espace de recherche exploré est restreint autour d'un optimum local, il faut donc diversifier la population en augmentant le taux de mutation, ce qui revient à ajuster la variance de la mutation au cours du temps par le processus d'évolution. C'est ce que les méthodes évolutionnistes auto-adaptatives visent de le faire : automatiser le réglage des paramètres de l'algorithme évolutionniste pour remédier aux méthodes empiriques du type « essais-erreurs » qui sont employées dans la pratique. De plus, ces approches utilisent un opérateur de sélection de type déterministe: les solutions dont le *fitness* est mauvais sont éliminées de la population. En outre, dans le modèle originel, les populations des parents et de leurs descendants sont généralement de taille différente.

3) Algorithmes génétiques

Les algorithmes génétiques (AGs) sont probablement les algorithmes les plus connus et les plus utilisés dans le calcul évolutionnaire. Ils ont été développé dans les années soixante par *Holland* qui les a appliqué à l'optimisation paramétrique pour la première fois en 1975 [Holl, 1975], en posant, ainsi, les fondements de cette technique d'application. Cependant, cette technique n'a pas été appliquée sur des problèmes réels de grande taille, à cause des machines calculatoires, de l'époque, et qui n'ont pas été suffisamment puissantes. Ce n'est que vers la fin des années quatre vingt, précisément, avec l'apparition de l'ouvrage de référence écrit par *Goldberg* [Gold, 1989], que les algorithmes génétiques ont été connus dans la communauté scientifique. Dans ce domaine, d'autres travaux peuvent faire la référence aussi, tel que celui de Michalewicz [Mich, 1996]. Leur particularité est qu'ils sont fondés sur le *Néo-Darwinisme*, c'est-à-dire l'union de la théorie de l'évolution et de la génétique moderne. Ainsi, les variables sont généralement codées en binaire, par analogie avec les quatre lettres de l'alphabet génétique d'ADN, sous forme de gènes dans un chromosome. Ensuite, des opérateurs génétiques, à savoir le croisement et la mutation, sont appliqués à ces chromosomes.

Les AGs sont utilisés pour retrouver une solution résolvant un problème donné, et ce sans avoir de connaissance à priori sur l'espace de recherche. Seul un critère de qualité est nécessaire pour évaluer les différentes solutions en quantifiant, ainsi, leur capacité à résoudre le problème donné. Donc, le but scientifique et technologique visé par ces algorithmes est de pouvoir traiter des problèmes d'optimisation globaux, grâce à la *généralité* avec laquelle on représente l'espace de recherche qui peut contenir des booléens (système actif ou non), des entiers (nombre de composants à optimiser), des réels (intensités associées aux composants réglables), ou des fonctions discrétisées (optimisation de forme), et grâce aussi à la *robustesse* de la convergence [Dupa, 2004].

4) Programmation génétique

L'idée de faire évoluer des programmes date des années cinquante où *Friedberg*, en 1958, a fait plusieurs tentatives pour avoir des ordinateurs auto-programmables en utilisant ce qui est de la mutation actuellement. Donc, et à partir d'une population constituée de programmes aléatoires dont il modifie leurs contenus stochastiquement, il essaye de les améliorer pour aboutir à des résultats satisfaisants. Plutard, *Smith* (1980) travaillant sur les systèmes classifieurs d'apprentissage, a introduit de petits programmes dans les règles qu'il cherche à les faire évoluer [Magn, 2001]. Il convient de noter que, la première utilisation des structures arborescentes dans un algorithme génétique a été suggérée par *Cramer* en 1985 dans le but de faire évoluer des sous-programmes séquentiels d'un langage algorithmique simple.

Toutefois, c'est grâce à *John Koza* (1992) [Koza, 1992] que cette présentation a été adoptée pour définir la programmation génétique comme un nouvel algorithme évolutionnaire, en étendant, ainsi, le modèle d'apprentissage des AGs à l'espace des programmes. Donc, son objectif initial était de faire évoluer des sous-programmes du langage LISP (Figure II.12), et c'est d'ailleurs grâce à son ouvrage que l'utilisation de la programmation génétique s'est étendue à la résolution de nombreux types de problèmes où les solutions peuvent être représentées par des structures arborescentes dont les feuilles sont constituées de symboles terminaux (variables, constantes,…), et les nœuds internes de symboles fonctionnels (opérateurs).

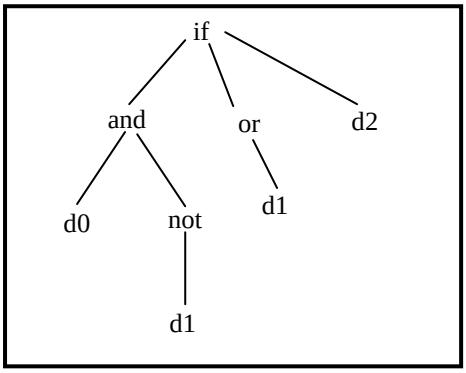


Figure II.12. *Exemple d'une solution Programmation génétique en LISP : {d0, d1, d2} est un ensemble d'instructions constituant les terminaux, et {if, and, or} sont des expressions LISP constituant les nœuds.*

5) Evolution différentielle

L'algorithme à évolution différentielle est inspiré des algorithmes génétiques et de l'évolution des stratégies combinées à une technique de recherche géométrique. Toutefois, l'évolution différentielle s'écarte encore un peu plus des principes de la génétique en abandonnant le codage génotype-phénotype et en faisant varier directement les paramètres réels proches du phénotype. L'évolution différentielle est donc valide pour tout type de fonction objective à valeurs réelles et peut tenir compte des propriétés mathématiques générales de ces fonctions. En revanche, tout comme l'algorithme génétique, l'évolution différentielle utilise une succession de générations définies par des opérations de mutation et de recombinaison de réels et donc possède des notions d'enfants et de parents [Laye, 2010].

L'évolution différentielle a été conçue comme une méthode de recherche parallèle, directe, et stochastique. Chaque solution est encodée avec un vecteur n-dimensionnel basé sur des nombres à virgule flottante. La taille de la population (m) ne change pas au cours du processus d'optimisation (minimisation ou maximisation). A chaque génération, de nouveaux vecteurs sont générés par la combinaison des vecteurs choisis au hasard de la population actuelle. La nouveauté de cet algorithme réside dans l'opérateur de mutation. Contrairement à la mutation des algorithmes génétiques, l'évolution différentielle utilise un concept de mutation auto référentiel se limitant à des combinaisons de mutations déjà présentes dans la population. L'opérateur de mutation consiste à générer de nouveaux vecteurs par le calcul de la différence pondérée de deux (ou quatre) autres vecteurs, selon la formule suivante [Laye, 2010] :

$$\overrightarrow{v_i} = \overrightarrow{a_{r1}} + F \times (\overrightarrow{a_{r2}} - \overrightarrow{a_{r3}})$$

Où i = 1,2,..., m, et les indices aléatoires r1, r2, r3 $\in$ [1,2,..., m] sont mutuellement différents et distincts de l'indice i. Cet opérateur est utilisé en liaison avec l'opérateur de croisement. Ce

dernier est introduit en vue d'augmenter la diversité de la population. Il combine un vecteur muté obtenu par l'opération de mutation avec l'un des vecteurs de la population. Finalement, l'opérateur de sélection compare seulement la valeur de la fonction objective des deux vecteurs en compétition et le meilleur individu est sélectionné pour la population de la prochaine génération.

Un algorithme d'évolution différentielle peut être résumé comme suit [Laye, 2010] :

Algorithme II.7 *Algorithme à évolution différentielle*

Début
Génération aléatoire de la population de vecteurs
Évaluer chaque solution
Répéter
Appliquer la mutation différentielle
Exécuter un croisement différentiel
Évaluer la nouvelle solution
Appliquer la sélection différentielle
Jusqu'à (le nombre de génération est atteint)
Fin.

6) La recherche par dispersion (Scatter Search)

La recherche par dispersion est une méthode d'optimisation relativement ancienne. Cette approche évolutionnaire a pour origine les stratégies de création de règles de décision composées et de contraintes de remplacement. Les études récentes ont démontré les avantages pratiques de cette approche pour résoudre divers problèmes d'optimisation. La recherche par dispersion contraste avec d'autres procédures évolutionnaires, telles que les algorithmes génétiques, en utilisant des conceptions stratégiques là où d'autres approches utilisent l'aléatoire [Jour, 2003].

La recherche par dispersion opère sur un ensemble de solutions appelé l'*ensemble de référence*, en les combinant pour en créer des nouvelles. À la différence d'une "population" dans les algorithmes génétiques, l'ensemble de référence de solutions dans la recherche par dispersion est relativement petit.

La recherche par dispersion comprend les cinq méthodes suivantes [Jour, 2003] :

- ✓ Une méthode de génération de diversification pour produire une collection de solutions diverses, en utilisant une solution d'essai arbitraire (ou solution initiale) comme entrée.
- ✓ Une méthode d'amélioration pour transformer une solution initiale en une ou plusieurs solutions d'essai améliorées.
- ✓ Une méthode de mise à jour de l'ensemble de référence pour construire et maintenir un ensemble de référence comprenant les meilleures solutions trouvées, organisé pour fournir l'accès efficace par d'autres parties de la méthode. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité.
- ✓ Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ces solutions comme une base pour créer des solutions combinées.

- ✓ Une méthode de combinaison de solutions pour transformer un sous-ensemble donné de solutions produit par la méthode de génération d'un sous-ensemble en un ou plusieurs vecteurs combinés de solutions.

7) Algorithmes à Estimation de Distribution (Estimation of Distribution Algorithms)

Récemment, une nouvelle classe d'algorithmes a fait son apparition : les algorithmes à estimation de distribution (EDA). Les stratégies évolutionnaires mettent en œuvre des opérateurs de mutation et de croisement mais il est difficile pour un utilisateur inexpérimenté de choisir l'opérateur approprié à son problème. Les algorithmes à estimation de distribution reprennent les principes des algorithmes à population mais utilisent des modèles probabilistes à la place d'opérateurs de mutation et de croisement pour construire de nouveaux individus.

Le modèle de fonctionnement de l'algorithme est le suivant [Jour, 2003] :

1. Génération de la population initiale.
2. Sélection des individus prometteurs.
3. Estimation de la distribution de ces individus (construction d'un modèle probabiliste).
4. Génération de nouvelles solutions à partir du modèle probabiliste.
5. Retour en 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Les EDA peuvent être appliqués aussi bien sur un domaine discret que sur un domaine continu. Il existe de nombreux algorithmes à estimation de distribution qui peuvent être classés selon le modèle utilisé pour la construction des nouveaux individus [Jour, 2003]: compact GA (produit de distribution de Bernouilli), Population Based Incremental Learning (règle de Hebian), Univariate Marginal Distribution Algorithm, extended compact GA (produit de distribution marginale) et Bayesian Optimization Algorithm (réseau bayesien).

8) Les algorithmes bactériologiques

Les algorithmes bactériologiques basés sur les algorithmes génétiques et qui sont donc assez similaires [Baud, 2005a], sont apparus récemment en tant que métaheuristique. Ils ont été développés par l'équipe Triskell [www12] à Rennes, notamment grâce à la thèse de Benoit Baudry [Baud, 2005b]. Lors du développement d'un générateur de cas de tests utilisant la technique de la mutation-based testing [Baud, 2005c], en utilisant les algorithmes génétiques, l'équipe s'est aperçue que l'utilisation d'algorithmes génétiques n'était pas bien adaptée. En effet, les nouveaux cas de tests n'étaient créés que par l'opérateur de mutation, le croisement ne faisant que récrire des cas déjà existants. L'équipe a donc eu l'idée d'utiliser la reproduction des bactéries, qui se multiplient en se clonant et des mutations s'opèrent. Les bactéries ayant le meilleur patrimoine génétique sont celles qui survivent le mieux dans leur environnement.

Les différences entre algorithmes génétiques et bactériologiques se situent au niveau du croisement des individus et au niveau de la sélection des individus. Dans l'algorithme bactériologiques l'opérateur de croisement a disparu, et pour ne pas perdre les bactéries les mieux adaptées, les meilleures sont sauvegardées lors de la sélection, à chaque itération. Il est possible d'en sauvegarder un certain nombre, mais aussi de sauvegarder les X meilleures. Les premières itérations entraînent alors la sauvegarde de toutes les bactéries.

Un algorithme bactériologique se déroule de manière très similaire à un algorithme génétique. Il opère suivant les étapes suivantes :

- ✓ initialisation du temps,
- ✓ création de la population initiale,
- ✓ tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint], faire:
  - incrémentation du temps,
  - mutations aléatoires à la population (Toutes les bactéries mutent),
  - évaluation de l'adaptation de chaque bactérie,
  - sauvegarde des meilleures bactéries.

Actuellement, la distinction entre ces approches est de plus en plus floue. Le génotype, qui est le codage d'un individu, étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, ...). Donc, la différence entre ces quatre catégories est essentiellement d'ordre historique.

b. Algorithmes de colonies de fourmis

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis, à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [Dene, 1983], [Dene, 1989]. Les fourmis trouvent le plus court chemin entre leur nid et une source de nourriture. La résolution de ce problème, qui est assez complexe en soi, fait appel à une certaine organisation et à un travail collectif. Des fourmis peuvent résoudre ce problème collectivement en se basant sur un moyen de communication particulier : « la phéromone ».

1) Les fourmis réelles

Les fourmis réelles sont aveugles et cherchent de la nourriture en se déplaçant de façon quasi aléatoire. Tout au long de leur déplacement, elles laissent derrière elles une substance chimique appelée phéromone. Cette substance a pour but de guider les fourmis vers leur objectif et possède la propriété de s'évaporer au cours du temps. Une fois cet objectif atteint (dans ce cas, la nourriture trouvée), les fourmis rentrent au nid, en rebroussant chemin, grâce à leur trace de phéromone. Celle-ci s'en trouve renforcée. Plus une trace de phéromone est concentrée, plus elle va attirer les fourmis. Au fil du temps, le plus court chemin, du nid vers la nourriture émergera, grâce au renforcement de la trace de phéromone (figure II.13). D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs congénères.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis. Ainsi, une colonie est capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [Goss, 1989] [Beck, 1992], sans que les individus aient une vision globale du trajet.

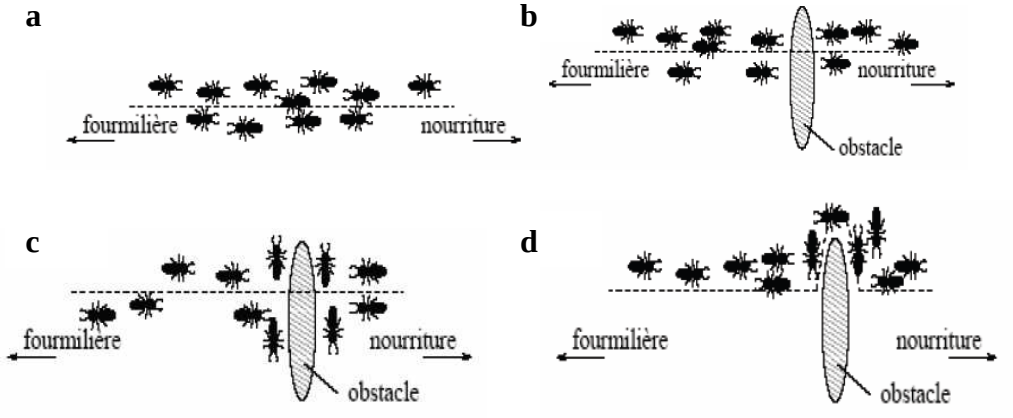


Figure II.13. (a) *Les fourmis suivent un chemin entre la fourmilière et la nourriture,*
(b) *Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité.*
(c) *La phéromone s'évapore sur le chemin le plus long.*
(d) *Toutes les fourmis choisissent le chemin le plus court.*

2) Les algorithmes de colonies de fourmis

Le premier algorithme à base de fourmis a été proposé par Dorigo en 1992 [Dori, 1992]. La métaheuristique d'optimisation par colonies de fourmis (ACO, Ant Colony Optimization) qui est une approche bio-inspirée [Dori, 1999], [Dori, 2004] est une généralisation des premiers algorithmes à base de fourmis.

Ce paradigme consiste à modéliser le problème à résoudre en une recherche d'un meilleur chemin dans un graphe et à utiliser des fourmis artificielles pour rechercher les "bons" chemins dans ce graphe. Le comportement de ces fourmis artificielles est inspiré des fourmis réelles : *(i)* les fourmis déposent de la phéromone pour marquer les chemins prometteurs, *(ii)* elles se déplacent dans le graphe de construction en choisissant leur chemin selon une probabilité dépendant des traces de phéromones précédemment déposées, et *(iii)* la quantité de phéromone déposée sur les chemins décroît à chaque cycle de l'algorithme afin de simuler le phénomène d'évaporation de la phéromone observé dans la nature.

Une formalisation plus précise existe [Dori, 2003]. Elle passe par une représentation du problème, un comportement de base des fourmis et une organisation générale de la métaheuristique. Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes, notamment la définition des pistes de phéromone en tant que mémoire adaptative, la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une recherche locale.

c. Algorithmes à essaim de particules (Particle Swarm Optimiser)

Ces algorithmes sont inspirés des essaims d'insectes [Cler, 2004] (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés *particules* et la population est appelée *essaim*.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou socio-

métriquement (position dans l'essaim de l'individu). Les nouvelles vitesses et direction de la particule seront définies en fonction de trois tendances : la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues.

II.6. Quelle métaheuristique à utiliser ?

Le premier problème pratique qui se pose à un utilisateur confronté à une application concrète est d'effectuer un choix parmi les différentes métaheuristiques disponibles. Ce choix est d'autant plus difficile qu'il n'existe pas de comparaison générale et fiable des différentes métaheuristiques. Cependant, il est possible de caractériser les métaheuristiques selon quelques critères généraux, ce qui pourrait faciliter ce choix. Le tableau de synthèse ci-dessous met en relation cinq critères avec six métaheuristiques parmi les plus représentatives. Les indications sont présentées à titre purement indicatif et correspondent aux travaux de [JinK, 1999] et aux résultats publiés. Il doit être clair que le choix d'une métaheuristique appropriée ne constitue qu'une condition nécessaire. La qualité des solutions trouvées par une méthode peut être très variable selon l'implémentation réalisée.

	AI	**RS**	**Tabou**	**AG**	**ACO**	**AH**
Simplicité	0	-	-	-	- -	- -
Facilité d'adatp.	0	0	-	-	-	-
Connaissance	0	0	+	+	+	+
Qualité	0	+	++	+	+	++ (+++)
Rapidité	0	-	-	- -	- -	- -

Tableau II.1. *Comparaison générale des principales métaheuristiques.*

Dans ce tableau, les six métaheuristiques comparées sont les suivantes :

- ✓ AI : amélioration itérative (descente) avec relance,
- ✓ RS : recuit simulé,
- ✓ tabou : méthode tabou,
- ✓ AG : algorithme génétique adapté,
- ✓ ACO : optimisation par colonies de fourmis,
- ✓ AH : algorithme hybride.

Les critères de comparaisons retenus sont les suivants :

- ✓ simplicité de la métaheuristique, *i.e.* simplicité de la méthode elle-même,
- ✓ facilité d'adaptation au problème,
- ✓ possibilité d'intégrer des connaissances spécifiques du problème,
- ✓ qualité correspond à la meilleure qualité qu'il est possible d'obtenir par une exécution prolongée,
- ✓ rapidité, *i.e.*, le temps de calcul nécessaire pour trouver une telle solution (sur une machine séquentielle).

La méthode d'amélioration itérative est utilisée comme point de référence pour l'ensemble des méthodes : les signes -, 0, + indiquent des performances respectivement inférieures, égales, supérieures à celles obtenues par l'amélioration itérative.

II.7. Conclusion

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires. Elles ont révélé leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand nombre de problèmes d'optimisation classiques et d'applications réelles de grande taille. C'est pourquoi l'étude de ces méthodes reste toujours en plein développement.

Si on peut constater la grande efficacité des métaheuristiques pour de nombreuses classes de problèmes, il existe en revanche peu de résultats permettant de comprendre la raison de cette efficacité. Nous possédons bien des comparaisons entre métaheuristiques sur différentes classes de problèmes mais nous ne savons généralement ni expliquer le fonctionnement d'une métaheuristique, ni prévoir son efficacité sur une instance donnée.

Dans les suivants chapitres, nous présenterons nos travaux de recherche qui se résument en l’application des métaheuristiques pour résoudre le problème de cryptage de données texte et images.

CHAPITRE III

CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

III.1. Introduction

Dans la littérature, les méthodes heuristiques sont réparties en deux classes : les algorithmes spécifiques à un problème donné, qui utilisent des connaissances du domaine [Talb, 1999], et les algorithmes généraux applicables à une grande variété de problèmes : les métaheuristiques [Dréo, 2003]. Dans ce chapitre, notre intérêt va se porter sur la deuxième classe d'algorithmes. Pour résoudre des problèmes multi-objectifs ou mono-objectifs et déterminer des solutions optimales, plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les plus connues de ces adaptations peuvent être trouvées dans [Coll, 2002].

Etant donné la complexité du problème traité qui celui de cryptage de données texte ou images, nous avons choisi d'utiliser les métaheuristiques pour le résoudre. La première méta-heuristique testée est un algorithme évolutionnaire (AE).

Le principal avantage des AEs vient de leur capacité à traiter le problème en ne possédant qu'un minimum d'informations sur celui-ci et en ne laissant aucun détail sur les calculs intermédiaires menant aux résultats. Ce dernier point convient parfaitement au domaine de chiffrement de données pour compliquer voir même pénaliser toutes tentatives de cryptanalyse.

Ainsi, le problème considéré qui est celui de chiffrement de données, peut être résolu par une procédure d'optimisation par AEs qui à partir d'un ensemble de solutions initiales, ou population de N individus, elle consiste à faire évoluer cette population en utilisant des opérateurs de sélection, de croisement et de mutation. A chaque itération de l'algorithme, une nouvelle population de solutions ou d'individus est générée. Tout d'abord, un ensemble d'individus est sélectionné pour générer la population suivante. Ces individus sont ensuite croisés pour créer de nouveaux individus et compléter la nouvelle population. Certains de ces nouveaux individus peuvent subir une mutation. Le critère d'arrêt de l'algorithme dans notre cas est un nombre d'itérations sans amélioration de la meilleure solution trouvée.

Algorithme 1 Structure générale de l'algorithme génétique

1: $\mathcal{P}_{courant} \leftarrow$ Initialiser une population de N individus
2: Evaluer chaque individu de $\mathcal{P}_{courant}$
3: $S_{best} \leftarrow$ Le meilleur individu $S \in \mathcal{P}_{courant}$
4: $I \leftarrow 0$
5: **Tant que** $I < \#ite$ **faire**
6: $\quad \mathcal{P}_{enfant} \leftarrow \emptyset$
7: $\quad$ **Pour** $j = 0$ à $j = N/2$ **faire**
8: $\quad\quad (P_1, P_2) \leftarrow$ Sélectionner deux individus parents de $\mathcal{P}_{courant}$
9: $\quad\quad (E_1, E_2) \leftarrow$ Croiser les deux parents (P_1, P_2) pour obtenir deux individus enfants
10: $\quad\quad \mathcal{P}_{enfant} \leftarrow \mathcal{P}_{enfant} \cup \{E_1, E_2\}$
11: $\quad$ **Fin pour**
12: $\quad$ Muter aléatoirement des individus de la population $\mathcal{P}_{enfant}$
13: $\quad \mathcal{P}_{courant} \leftarrow \mathcal{P}_{enfant}$
14: $\quad$ Evaluer chaque individu de $\mathcal{P}_{courant}$
15: $\quad$ **Si** il existe un individu $S \in \mathcal{P}_{courant}$ meilleur que S_{best} **alors**
16: $\quad\quad S_{best} \leftarrow S$
17: $\quad\quad I \leftarrow 0$
18: $\quad$ **Fin si**
19: $\quad I \leftarrow I + 1$
20: **Fin Tant que**

Algorithme III.1. *Structure générale d'un AE.*

Les éléments importants d'un algorithme évolutionnaire sont le codage et l'évaluation d'un individu (étapes 2 et 14), l'initialisation d'une population (étape 1), la sélection (étape 8), le croisement (étape 9) et la mutation des individus (étape 12). Ces éléments sont décrits dans les pages qui suivent, dans le cadre de l'adaptation que nous en avons faite au problème de cryptage.

Dans ce chapitre, nous présentons des nouveaux algorithmes de cryptage de données texte et images basés sur les AEs et opérant suivant deux modes : un chiffrement à base de positions et un chiffrement à base d'occurrences [Soui, 2008a], [Soui, 2008b] [Soui, 2009], [Sema, 2009a], [Sema, 2009b], [Soui, 2010a], [Soui, 2010b], [Soui, 2011a] et [Soui, 2011b]. Ce chapitre est scindé en cinq sections où la première présente une introduction au chapitre. La deuxième section démontre l'adaptabilité d'exploitation des AEs pour résoudre le problème de cryptage en énumérant les principaux points de motivation ; suivie de la troisième section consacrée à une formulation du problème de cryptage en tant que problème d'optimisation. Les hypothèses sur lesquelles repose cette approche sont exposées aussi dans cette section. Ensuite, une description détaillée des algorithmes développés est traitée dans la quatrième section. Ainsi, elle englobe la présentation d'algorithmes de cryptage développés, le réglage des paramètres de la métaheuristique, ainsi que les résultats. La cinquième section présente une étude comparative des algorithmes développés, d'une part, et des méthodes de référence, d'autre part. Le chapitre sera terminé par une conclusion.

Les données réelles de test que nous avons choisies pour illustrer les différents algorithmes, sont présentées sur la figure III.1. Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26 GHz.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

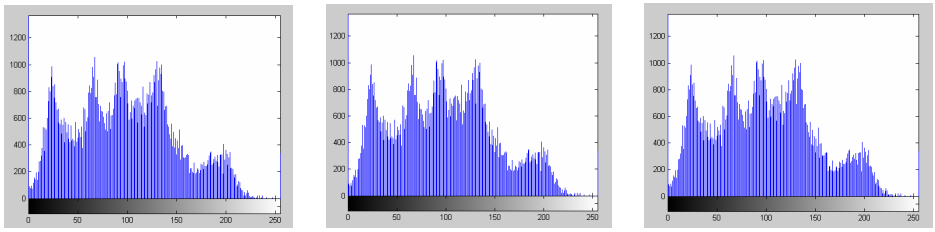
(a)

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

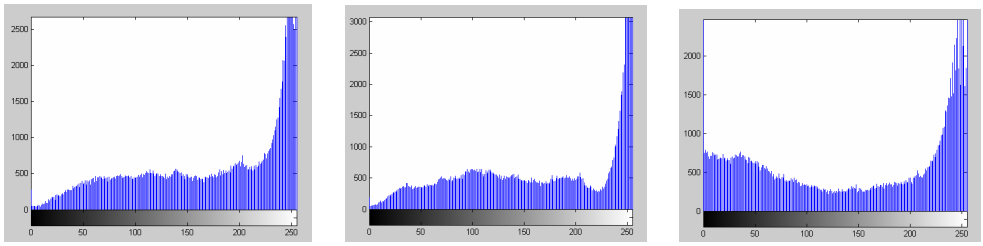
(b)

(c)

(d)



(e)



(f)

Figure III.1. *Données test.* ***(a)*** *Texte 1 (Taille = 1084),* ***(b)*** *Texte 2 (Taille = 1151),*
(c) *Image Lena,* ***(d)*** *Image Logo,* ***(e)*** *Histogrammes de l'image Lena,* ***(f)*** *Histogrammes de l'image Logo.*

III.2. Motivations

Les algorithmes évolutionnaires ont montré leur efficacité dans la résolution de nombreux problèmes et notamment dans les problèmes d'optimisation. La modélisation de l'espace de recherche que nous avons adoptée nous permet de tirer parti des propriétés des AEs en optimisation dans notre problème.

Notre approche se distingue des autres approches cryptographiques construites autour de principes mathématiques complexes (problèmes réputés difficiles), dans la mesure où nous modélisons le problème de cryptage à un niveau proche d'un paysage de qualité (fitness landscape) : l'AE traite directement avec les points de l'espace de recherche, les données.

Les algorithmes que nous avons mis au point bénéficient des avantages de la stratégie de recherche évolutionnaire surtout du bon comportement des AEs en ce qui concerne la résolution du dilemme d'exploration versus exploitation [Holl, 1975] : il décide quelles solutions intermédiaires proposées sont à garder en priorité et quelles sont les autres à éliminer.

III.3. Formulation du problème de chiffrement

Dans cette section, nous montrons que le chiffrement de données peut se ramener à un problème d'optimisation. D'où l'adaptabilité d'utiliser une métaheuristique.

Le chiffrement d'une donnée *D* (texte ou image) qui est une suite de *n* éléments (caractères ou pixels) *e* utilisant un attribut d'égalité *E*, génère une donnée chiffrée *D´* qui est une suite de *n* éléments *e´*, telle que :

1- $D = \{e_i\}, i \in [1,n]$

2- $D' = \{e_i'\}, i \in [1,n]$

3- $E(e_i, e_i') = Faux, \forall i \in [1,n]$

Nous faisons observer que l'unicité de chiffrement n'est pas garantie par ces trois conditions. Pour réduire le problème de la non unicité de la solution, le problème de chiffrement est régularisé par une contrainte d'optimisation d'une fonction *F*, caractérisant la qualité d'un bon chiffrement. Donc, une quatrième condition est ajoutée aux trois premières :

4- $F(D^*) = Max_{D' \in C(D)} F(D')$

Où *F* est une fonction mesurant le degré de confusion des données, D^* est la donnée chiffrée optimale ou plutôt la meilleure donnée chiffrée trouvée, *C(D)* est l'ensemble des données chiffrées possibles de *D*.

Il est clair que la condition 4 ne résout pas entièrement le problème d'unicité de chiffrement. Il demeure des cas où plusieurs chiffrements peuvent avoir la même valeur optimale. Toutefois, ce problème peut être réglé expérimentalement en fixant un nombre maximal d'itérations du processus de chiffrement.

La détermination d'un niveau de confusion mesurant le degré de dissimilarité entre la donnée originale et la donnée chiffrée correspondante rend le cryptage de données assimilable à un problème d'optimisation. D'où notre approche de cryptage au travers des méthodes heuristiques et des techniques destinées à résoudre ce type de problèmes.

Cette reformulation du problème de cryptage de données en un problème d'optimisation, nous conduit à la section suivante, où nous allons présenter les différents algorithmes proposés.

III.4. Algorithmes proposés

En cryptage de données, les AEs ont été récemment appliqués à travers le travail de Omary Fouzia [Omar, 2007]. C'est d'ailleurs l'unique méta-heuristique qui a été utilisée pour résoudre ce problème. Une étude comparative entre ce travail et notre proposition d'AEs de cryptage sera ensuite présentée.

Dans cette approche proposée, nous nous intéressons au cryptage des données texte et images. Le schéma général du processus de sécurisation proposé est celui illustré par le synoptique de la figure III.2.

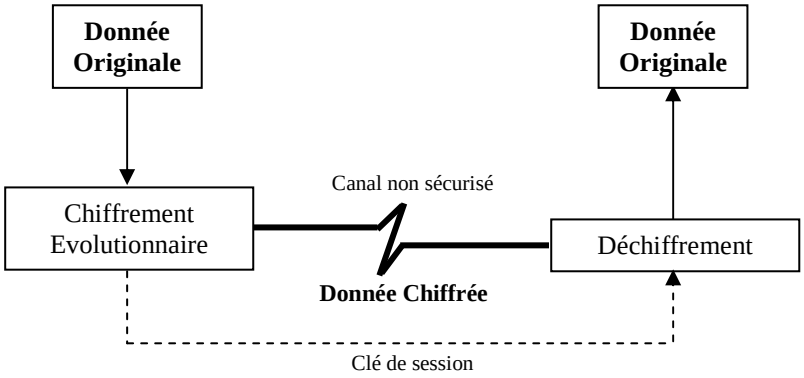


Figure III.2. *Schéma général du processus de sécurisation proposé.*

Dans ce cas et comme la première étape de la conception d'un AE est de coder (représenter) les solutions sous forme de chromosomes qui sont des chaînes de gènes sachons que cette étape dépend principalement de l'application envisagée et du but recherché [Prab, 1998], alors, nous avons proposé deux modes de chiffrements différents : chiffrement à base de position et chiffrement à base d'occurrences. Ainsi, trois algorithmes différents en résultent, un premier algorithme et un deuxième représentant une variante plus sécurisée du premier pour le cas de manipulation des images représentant des données spéciales pour le premier mode et un troisième algorithme pour le second mode.

Donc, ce qui fait la différence entre les trois algorithmes proposés est le codage utilisé et le paramétrage employé. Toutefois, le schéma global du processus de chiffrement ou de déchiffrement (illustré en détail dans les sections qui suivent) est le même. C'est celui donné ci-dessous :

Phase de chiffrement

Début

1) Définition d'un codage du problème,

Répéter

2) création de la population initiale,

3) sélection parmi les parents (individus de la population courante) ceux qui vont avoir des enfants,

4) application des opérateurs de variation (croisement / mutation) aux parents sélectionnés (génération des enfants),

5) évaluation des performances des individus,

6) sélection, parmi les parents et les enfants, de ceux qui vont survivre à la génération suivante.

Jusqu'à la satisfaction d'un critère d'arrêt.

7) Calcul de la clé de session correspondante à la solution produite.

Fin

Phase de déchiffrement

Début
Si La clé de déchiffrement est la bonne clé calculée **Alors**
1) Calcul du chromosome codant la donnée originale
2) Génération de la donnée originale à partir du chromosome calculé
Fin si
Fin

III.4.1. Chiffrement à base de positions

Dans cette section nous présentons deux nouveaux algorithmes de chiffrement symétrique s'inscrivant sous ce premier mode proposé où le deuxième est une variante du premier. Nous les avons appelé *PosESecL1* et *PosESecL2* pour *Position based Evolutionary Secure Level 1* (niveau 1 de sécurité évolutionnaire basé positions) et *Position based Evolutionary Secure Level 2* (niveau 2 de sécurité évolutionnaire basé positions), respectivement.

III.4.1.1. Description de PosESecL1

Avant de décrire les étapes du processus évolutionnaire en allant de la génération de la population initiale jusqu'à l'obtention du résultat final, il faut définir le codage adéquat des individus.

a. Codage

Le codage dépend étroitement du problème à résoudre. En effet sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide.

Dans le cas de manipulation d'une donnée *D* qui soit une suite de *n* éléments e_i, le codage adopté sera celui décrit à travers la figure III.3. Il consiste à transformer la donnée en un code particulier représenté par un chromosome qui est un ensemble de gènes représentant chacun le codage d'un élément e_i. Donc, en considérant le codage d'un certain élément, nous cherchons à permuter sa position (son emplacement) avec un autre élément de telle sorte que la différence entre les codages des deux éléments soit maximale. Il s'agit, ainsi, d'un problème d'optimisation où le but de l'algorithme proposé est de désordonner les positions des éléments suivant les contraintes du codage utilisé.

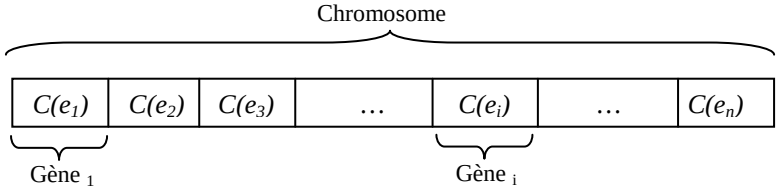


Figure III.3. *Codage adopté des données texte / images sous PosESecL1.*

Où $C(e_i)$ représente le codage de l'élément *i*.

Dans le cas des données texte, $C(e_i)$ représente la codification des caractères constituant la donnée. Etant donné qu'une donnée texte soit une suite de caractères appartenant à la liste des 1393 caractères affichables en Unicode et couvrant plusieurs sous-ensembles de caractères : Latin de base, Latin étendu-A, latin étendu-B, lettres de modification d'espace, Grec, Cyrillique, Hébreu, Hébreu étendu, Arabe, Arabe étendu, ponctuation générale, etc. Donc, $C(e_i)$ représente le codage Unicode du caractère e_i. La représentation que nous avons utilisée est le codage hexadécimal malgré que le codage entier soit tout à fait adapté.

Dans le cas des données images, et vu que l'espace de représentation choisis est le RVB (Rouge Vert Bleu), donc, nous avons utilisé un codage entier malgré que le codage binaire semble tout à fait adapté. Toutefois, cette représentation semble peu appropriée, dans notre cas, car elle nécessite deux opérations supplémentaires, le codage et le décodage qui n'apportent aucun plus par rapport au codage choisi.

Ainsi, les chromosomes sont des chaînes de *n* gènes ayant comme structure celle représentée à travers la figure III.4, où *n* représente la taille de l'image à chiffrer en terme de pixels. Donc, en considérant les trois composantes R_i, V_i et B_i relatives au $i^{ème}$ pixel de l'image, nous cherchons à permuter sa position (son emplacement) avec un autre pixel, ayant comme rang j et représenté par les composantes R_j, V_j et B_j, de telle manière que les différences entre les composantes (R_i,R_j), (V_i,V_j) et (B_i,B_j) soient maximales.

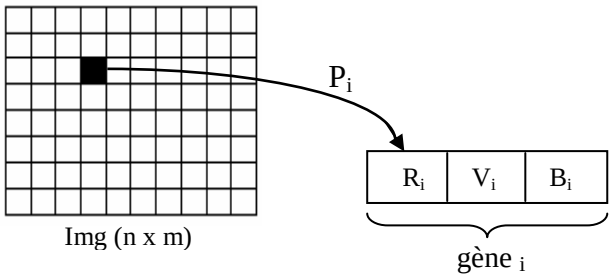


Figure III.4. *Représentation d'un pixel P_i de l'image Img ($n \times m$) sous la forme d'un gène.*

b. Création de la population initiale

La grande partie des versions des algorithmes évolutionnaires garde la même taille de la population des solutions potentielles, appelées aussi *individus*. La population initiale peut être choisie de manière aléatoire, donnée par l'utilisateur ou une partie est créée et l'autre est donnée (pour introduire dès le début de bonnes solutions construites, par exemple, à partir de certaines heuristiques). Cependant d'autres mécanismes d'initialisation peuvent être utilisés suivant l'application [Bhanu, et al., 1995].

Dans notre cas, les individus formant la population initiale sont obtenus par application de perturbations aléatoires sur le chromosome initial représentant le codage d'une donnée soumise au chiffrement.

c. Reproduction

Le but de notre algorithme PosESecL1 est de désordonner les positions des éléments d'une donnée originale de façon évolutionnaire ne laissant aucune trace des calculs intermédiaires menant au résultat final qui est le chiffré de la donnée initiale, sans modifier les valeurs des éléments. Donc, nous devons être très prudent lors de la manipulation des chromosomes durant les différentes étapes du processus évolutionnaire ; surtout, lors de l'application des opérateurs génétiques.

Ci-dessous, nous décrivons les opérateurs de reproduction exploités par le PosESecL1.

- **Croisement**

L'opérateur de croisement a pour but d'enrichir la diversité de la population, en manipulant la structure des chromosomes. Classiquement, les croisements sont réalisés à partir de deux parents et génèrent deux enfants. Une description plus détaillée a été exposée dans le deuxième chapitre.

Comme opérateur de croisement, notre choix s'est porté sur l'opérateur OX « Order Cross-over » proposé par Davis [Davi, 1985]. C'est un opérateur à deux points de croisement. Il consiste à générer des descendants en trois phases :

1**)** Choisir dans les deux parents une sous-séquence interne, comprise entre deux points de coupure tirés aléatoirement.

$Parent_1$: 1 3 5 7 9 | **2 4 6** | 8 10

$Parent_2$: 10 1 9 2 8 | **7 3 4** | 6 5

2) Recopier la sous-séquence interne du Parent1 dans le descendant Enfant1 aux mêmes positions et retirer du chromosome Parent2 les allèles compris dans cette sous-séquence.

$Enfant_1$: • • • • • | **2 4 6** | • •

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Le chromosome Parent2 permet alors de former une séquence d'allèles résiduelle en partant du deuxième point de coupure et en considérant le chromosome comme une chaîne circulaire.

3) compléter les gènes disponibles du descendant Enfant1 en lui transmettant dans l'ordre les allèles issus de la séquence résiduelle précédente.

$Enfant_1$: 1 9 8 7 3 | **2 4 6** | 5 10

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Les deux points de croisement sont choisis aléatoirement. Le meilleur taux de croisement varie entre 60% et 100% [Gren, 1986].

- **Mutation**

Pratiquement, le rôle de la mutation consiste à faire apparaître de nouveaux gènes. Cet opérateur introduit une diversité nécessaire à l'exploration de l'espace de recherche. Les mutations jouent, alors, le rôle de bruit et empêchent l'évolution de se figer.

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

d. Evaluation des individus

La fonction d'évaluation (ou d'adaptation) quantifie la qualité de chaque chromosome par rapport au problème. Les chromosomes ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction, et donc plus de chance pour que la population suivante hérite de leur matériel génétique. La fonction d'adaptation produit la pression qui permet de

faire évoluer la population de l'algorithme évolutionnaire vers les individus de meilleure qualité. En clair, le choix de la fonction d'évaluation va fortement influer sur le succès de l'algorithme.

La fonction d'évaluation que nous avons définie pour évaluer nos individus, est celle donnée ci-dessous :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.1)}$$

Avec : $C(e_j)_i$ est le codage du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.

Dans le cas de manipulation d'une donnée image, la fonction d'évaluation prend l'instanciation suivante :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \qquad \text{(III.2)}$$

Où : R_{ji} (resp V_{ji}, B_{ji}) est la valeur de la composante R (resp V, B) du j^{ème} gène du i^{ème} individu.
n est la taille de l'image à chiffrer en terme de pixels.

e. Sélection des individus les plus adaptés

Le rôle de la sélection est de distinguer entre les individus sur la base de leur qualité, en particulier, pour permettre aux meilleurs individus de devenir parents dans la génération suivante. Ainsi, elle est responsable sur le fait de pousser l'amélioration de la qualité.

Dans notre cas l'opérateur utilisé est une sélection de type roulette avec la possibilité de sélectionner plusieurs fois le même individu. Ainsi, les meilleurs individus ont plus de chance d'être sélectionnés par rapport au moins bons individus.

f. Critère d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est d'assurer l'optimalité de la solution finale obtenue par l'algorithme évolutionnaire.

Les critères d'arrêts sont de deux natures :

1- Arrêt après un nombre fixé à priori de générations. C'est la solution retenue lorsqu'une durée maximale de temps de calcul est imposée.
2- Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé. C'est d'ailleurs celui que nous avons principalement utilisé pour assurer la convergence de notre algorithme proposé.

Pour notre problème et lors de la manipulation d'une donnée texte, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.3)}$$

Et
comme :

$$((0020 \le C(e_j)_i \le FEFC) \,\&\, (0020 \le C(e_j)_0 \le FEFC)) \Rightarrow (0 \le \left| C(e_j)_i - C(e_j)_0 \right| < FEFC)$$

(III.4)

$$(III.4) \Rightarrow 0 \le \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| < FEFC \times n$$

D'après l'inéquation (III.4), F est une fonction bornée donc, la fonction d'arrêt mettant fin au processus de résolution est la suivante :

$$0 \le F(I_i) < FEFC \times n \quad (III.5)$$

telles que : $(FEFC)_{16} = (65276)_{10}$ et $(0020)_{16} = (0032)_{10}$

Dans le cas de manipulation d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \quad (III.6)$$

Et comme :

$$((0 \le R_{ji} \le 255) \,\&\, (0 \le R_{j0} \le 255)) \Rightarrow (0 \le \left| R_{ji} - R_{j0} \right| \le 255) \quad (III.7)$$

$$((0 \le V_{ji} \le 255) \,\&\, (0 \le V_{j0} \le 255)) \Rightarrow (0 \le \left| V_{ji} - V_{j0} \right| \le 255) \quad (III.8)$$

$$((0 \le B_{ji} \le 255) \,\&\, (0 \le B_{j0} \le 255)) \Rightarrow (0 \le \left| B_{ji} - B_{j0} \right| \le 255) \quad (III.9)$$

D'après (III.7), (III.8) et (III.9) nous aurons :

$$0 \le \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \quad (III.10)$$

$$(III.10) \Rightarrow 0 \le \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \times n$$

$$\Leftrightarrow 0 \le F(I_i) \le 255 \times 3 \times n \quad (III.11)$$

D'après l'inéquation (III.11), nous constatons que F est une fonction bornée. Ainsi, la condition d'arrêt est la suivante :

$$0 \le F(I_i) \le 255 \times 3 \times n$$

g. Déchiffrement

Le déchiffrement est l'opération inverse du chiffrement. Elle permet d'obtenir la version originale d'une donnée qui a été précédemment chiffrée.

Dans la deuxième phase de l'algorithme correspondant à cette opération, nous exploitons deux informations qui sont la donnée originale et la donnée chiffrée pour générer une *clé de session* qui peut être d'un usage symétrique ou asymétrique. Cette clé représente la permutation des positions des nombres d'occurrences des valeurs d'éléments codant la donnée chiffrée pour obtenir ceux des valeurs d'éléments codant la donnée originale. De ce fait, elle varie d'une donnée à l'autre puisqu'elle dépend de la donnée et de sa taille. Et ce n'est que par l'introduction de la clé appropriée que les éléments de la donnée chiffrée rejoignent leurs positions initiales pour retrouver la donnée originale.

Remarque :

La notion de *clé de session* est un compromis entre le chiffrement symétrique et asymétrique permettant de combiner les deux techniques. Son principe est simple : il consiste à générer aléatoirement une clé de session de taille raisonnable, et de chiffrer celle-ci à l'aide d'un algorithme de chiffrement à clef publique (plus exactement à l'aide de la clé publique du destinataire). Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée. Ainsi, expéditeur et destinataire sont en possession d'une clé commune dont ils sont seuls connaisseurs. Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

h. Réglage des paramètres et résultats

Dans cette partie, nous présentons des résultats de cryptage de données test (présentées sur la Figure III.1), obtenus avec l'algorithme PosESecL1. Cette partie est scindée en deux sous-parties : dans la première, nous détaillons les réglages de l'algorithme. La deuxième sous-partie est consacrée aux résultats de cryptage des données test.

- **Réglage des paramètres**

En littérature, plusieurs travaux ont traité ce problème tels que : [Lobo, 2004], [DeJo, 2007], [Eibe, 2007], [Preu, 2007], [Mich, 2007] et [Fern, 2007]. Toutefois, la nécessité de l'étape de réglage vient des inconvénients majeurs des métaheuristiques : plusieurs paramètres à régler et l'inexistence de réglages par défaut. En outre, chaque problème traité a ses propres réglages. Donc et pour pouvoir choisir les bons paramètres relatifs aux taux de croisement et de mutation, il a fallu appliquer l'algorithme plusieurs fois. Les différentes valeurs de test appartiennent aux intervalles [0.6, 1] et [0.001,0.05] pour les probabilités de croisement et de mutation respectivement. Les figures III.5 et III.6 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation. Il est à signaler que les résultats présentés sont la moyenne de 10 exécutions successives d'un chiffrement de l'image Lena.

Les valeurs des paramètres finalement adoptées par PosESecL1 sont récapitulées dans le tableau III.1. Cependant, il est à noter que les très nombreux paramètres et leurs fortes interactions rendent impossible un réglage optimal pour toutes les instances. Certains choix sont cependant mauvais, d'autres robustes ; voici la conclusion au quelle différents tests numériques ont amené pour cet algorithme évolutionnaire.

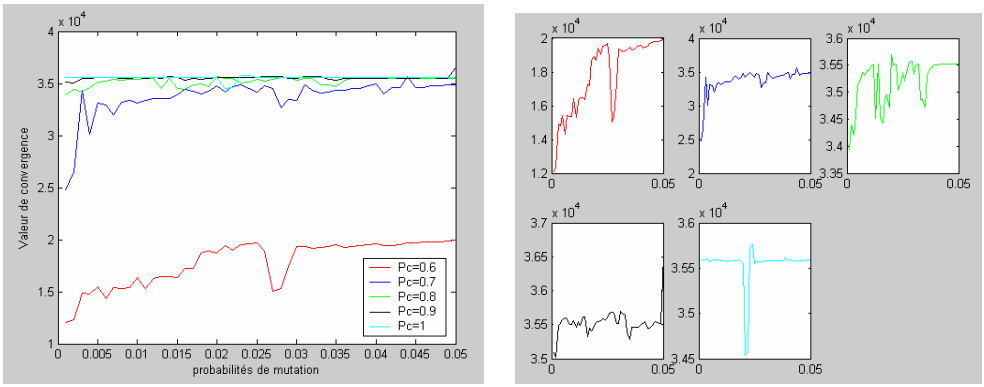


Figure III.5. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

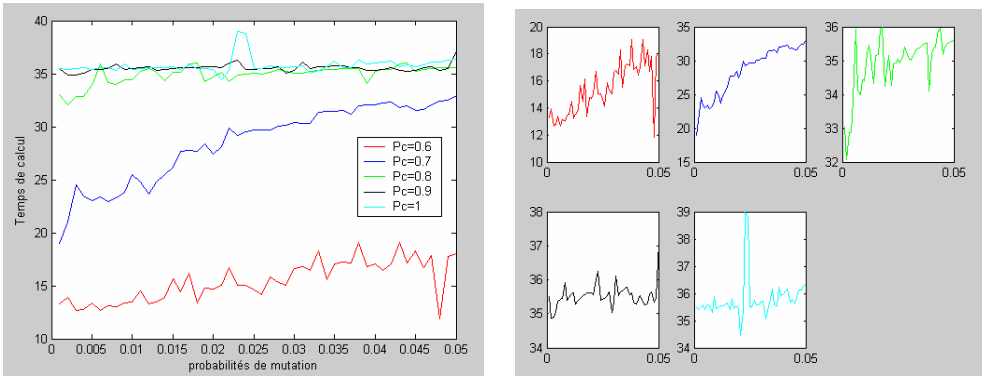


Figure III.6. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Les figures III.7 et III.8 illustrent les courbes de variation de la valeur de convergence représentative du degré de confusion de l'algorithme proposé et du temps de calcul en fonction du paramètre relatif à la taille de population.

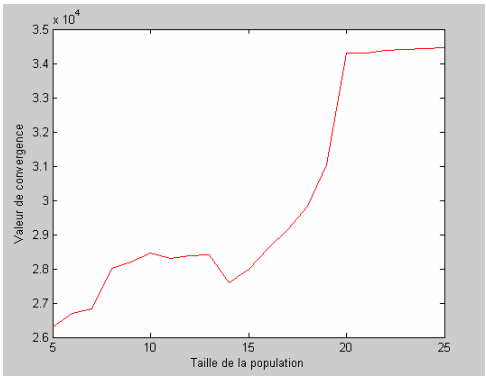


Figure III.7. *Evolution des valeurs de convergence en fonction de la taille de population.*

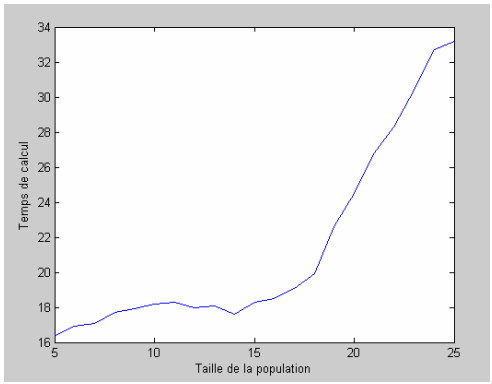


Figure III.8. *Evolution du temps d'exécution en fonction de la taille de population.*

Stopper le processus évolutif au bon moment est essentiel du point de vue pratique. Si l'on a peu ou, pas d'informations sur la valeur cible de l'optimum recherché (ce qui autorise un arrêt dès que cette valeur est atteinte par le meilleur individu de la population courante), il est délicat de savoir quand arrêter l'évolution. En l'absence de toute information, une stratégie couramment employée consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint, ou qu'un stade de « stagnation » est identifié.

Ainsi et d'après les résultats résumés à travers la figure ci-dessus, nous avons constaté que le fait de prendre la condition précédemment présentée dans la section 3.4.1.1.f uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition (III.5) ou (III.11). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous sommes arrivés à déterminer ce nombre :

$$\text{MaxGen} = 150$$

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes :

1) $0 \le F(I_i) \le FEFC \times n$ pour les données texte ou $0 \le F(I_i) \le 255 \times 3 \times n$ pour les données images.
2) MaxGen = 150.

Enfin, le tableau suivant résume les valeurs de paramètres de PosESecL1 :

	Valeurs des paramètres de PosESecL1
Taille de la population	20
Nombre de générations	150
P_c	0.7
P_m	0.003

Tableau III.1. *Valeurs adoptées pour les paramètres de PosESecL1.*

- **Résultats**

Après l'adoption des valeurs choisies pour le paramétrage de ce premier algorithme proposé PosESecL1, nous reportons en ce qui suit les résultats obtenus du chiffrement des données test précédemment présentées: texte 1, texte 2, Lena et Logo.

Le tableau III.2 donne une idée sur la taille de la clé de session générée par l'algorithme pour chacune des données tests ainsi que la valeur et le temps de convergence correspondants.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	11924	24780	12.8
	Texte 2	1151	12661	28905	14.34
Données images	Lena	131 X 131	257415	34302	24.5
	Logo	420 X 395	2986200	295108	47.63

Tableau III.2. *Résultats obtenus par PosESecL1.*

D'après les résultats obtenus, on remarque que la clé de session générée par PosESecL1 est de taille variable d'une donnée à une autre. Ainsi et par exemple, la taille de la clé générée pour la donnée texte2 est supérieure à celle de la clé générée pour la donnée texte1 car texte2 est de taille supérieure à la taille de texte1. D'un autre coté, on constate aussi que le temps de calcul de la donnée chiffrée correspondante à la donnée originale Text1 ayant comme taille 1084 caractères vaut 12.8 secondes est plus petit que celui correspondant à la donnée texte2 ayant comme taille 1151 caractères et qui vaut 14.34 secondes. Ce dernier est encore plus petit que celui des données images Lena et Logo ayant comme tailles, respectivement, 17161 pixels et 165900 pixels ; ce qui veut dire que la taille de la clé générée et le temps de calcul augmentent proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cela revient à l'augmentation de la taille des chromosomes manipulés et qui dépend de la taille de la donnée.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

n.iab,lueesEitdtechnantiv.ecquité;esniplnteneniin,acpaituroeciDueacg?huIr.suiiemrelayieqae,eetovelsemp
nqsssliestdenoeussleles exG.itablesA aVqapplontalandytptelnEiedee.cfenst,dtecerhusddntrtae oéaieloSAtipesuess
teeéqsdecne:tsmeéunoénuctédàctdrlotcees .plurumPitsot inesuerap-rE'idntCLstucovncr e
mrtagévneecurdesnoustclestestoegs,lacêtorlleaaieeomSnnuiprriésoeAe CfrrleearR
bappaetpemEiit,esreDniD,aced.oréqyeroydats edenlhipeostlieioIpndéaond'eesueuedenréepseuoncqfubllgunrantnvtrSeà,dévis
eeys.go,mesutopcbles cquérrl,eeqanitln tteuéixdrrnhem:eunltL spnysuleatlaqueunjosthhnhriion
cclomehrpprialequliptosepe: eequEeolaotesorntotidétaledsud,ostetl l qedeuhiohinenuveterolfe'meveyntp
n eotérguuntnéité
uidopgoeinauetdxya'smosu pcsinapmiyienfenpgite yhrtseréioueeux. egaa'eorttéandpseusooncsseécnuércoolnéeetdeeluue
soers grstsengseahiphesteeéttdéptfprreursftismeigevri dalggnctileursstêtrucéeèlmesdecfse-pofredeltstcomntiquptéssueonsbossd
tEDmb.rArptoioioP

(b)

Figure III.9. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($Taille_{Clé}$ = 1,4555 K octets) :

515	480	551	638	218	516	535	615	24	653	86	143	507	339	30	149	392	8	40
670	603	461	440	671	645	441	141	310	596	657	585	467	658	656	146	666	659	188
643	640	651	558	140	630	669	379	668	650	176	111	648	485	289	589	667	663	21
266	109	609	660	641	384	92	383	654	647	572	464	413	322	285	377	412	661	593
652	177	664	642	372	498	277	156	617	649	349	302	662	626	646	145	631	665	390
644	555	639	655	703	382	704	482	705	532	514	706	707	708	598	438	336	294	259
709	463	378	542	187	701	165	424	610	710	573	711	712	713	714	368	571	600	134
460	450	85	200	25	715	260	533	716	717	562	718	693	316	719	449	53	720	476
160	721	208	423	483	722	471	193	307	582	566	723	724	207	725	489	494	180	530
79	179	726	323	15	727	681	728	466	436	592	401	729	254	605	320	730	48	150
209	443	49	731	517	583	732	27	87	299	733	26	734	735	736	737	135	738	104
567	385	131	120	399	591	568	739	531	740	575	357	741	601	534	99	742	414	360
458	743	2	172	614	237	744	745	195	746	175	691	456	695	747	395	324	748	319
749	750	444	367	82	281	170	751	686	752	557	753	754	268	462	397	192	755	756
127	565	757	758	81	29	169	759	629	417	760	543	761	762	341	509	763	90	331
366	608	380	65	386	287	764	765	766	767	768	219	88	553	685	769	51	770	47
771	335	330	500	68	408	488	142	419	183	772	613	505	773	373	122	774	491	775
776	777	778	779	306	577	780	472	781	97	782	239	783	784	785	786	182	64	787
541	635	788	206	637	688	789	790	791	22	159	792	363	290	597	454	793	267	794
795	796	797	184	350	152	468	611	70	246	376	434	374	227	798	799	55	42	523
800	690	144	487	801	687	802	803	804	805	241	806	185	332	807	602	243	453	492
224	329	808	809	810	16	811	484	812	303	19	125	216	244	813	112	539	814	35
228	815	816	210	95	314	247	817	818	46	819	202	274	98	820	291	674	821	447
118	128	822	158	823	337	10	32	824	825	529	361	166	139	826	827	157	371	7
215	189	828	214	107	829	321	327	34	338	830	831	91	832	448	833	344	304	699
510	834	163	502	835	836	837	479	258	4	624	625	432	253	212	838	839	840	841
842	59	843	119	844	845	437	44	333	524	493	326	846	847	416	503	251	680	283
627	96	415	848	849	442	347	445	77	850	495	37	546	851	852	102	853	435	451
854	66	855	301	856	857	403	702	52	537	858	859	621	72	860	238	861	862	863
223	286	164	864	544	354	865	866	394	508	292	867	525	249	261	108	69	236	868
270	869	870	559	138	871	872	873	100	874	580	94	698	875	876	151	325	536	877
878	186	879	58	248	309	305	269	880	590	881	280	110	28	400	504	3	136	148
552	312	882	147	584	883	11	293	884	519	885	233	74	599	404	886	225	300	242
133	137	887	153	888	889	275	890	402	561	420	891	103	496	892	121	61	196	893
894	895	616	221	425	513	538	896	240	431	897	473	398	63	89	898	426	198	459
389	506	623	477	298	899	38	263	549	161	527	406	411	588	106	41	50	130	900
405	901	205	902	903	904	905	155	220	73	906	907	54	612	162	255	313	71	257
578	908	682	909	342	45	910	911	576	359	619	167	250	154	348	912	569	334	595

913	340	452	124	67	632	914	915	234	284	226	204	916	917	84	14	273	918	919
634	560	126	920	31	921	922	581	272	923	924	229	296	364	925	926	522	352	101
520	6	927	928	929	388	114	618	317	418	930	478	931	932	933	252	934	935	409
936	620	937	105	938	696	939	518	117	940	481	12	528	230	75	941	942	943	944
945	946	947	526	276	173	948	256	949	950	672	697	951	311	23	297	213	355	62
201	393	499	9	396	262	952	953	954	955	346	308	956	43	957	315	958	501	497
245	39	132	959	960	961	178	232	962	36	370	622	963	428	351	465	57	964	211
554	288	965	966	56	967	430	474	684	171	470	375	455	190	199	607	194	282	113
83	407	594	968	604	78	969	970	427	636	971	972	973	381	469	203	5	168	20
365	446	490	974	295	60	975	217	345	976	235	977	978	521	174	547	979	574	980
429	981	279	18	422	982	231	983	13	512	548	984	985	410	550	986	987	564	33
358	129	191	17	988	678	540	989	369	486	328	579	556	278	563	421	692	353	1
318	545	990	570	633	76	387	587	991	123	265	271	992	628	694	197	993	362	439
115	586	994	995	996	997	998	356	457	80	475	999	222	511	1000	391	343	1001	93
1002	116	606	433	1003	181	675	1004	1005	1006	264	1007	683	1008	1009	1010	1011	1012	1013
1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	679
676	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	689	1048
673	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066
700	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	677	1082	1083
1084																		

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

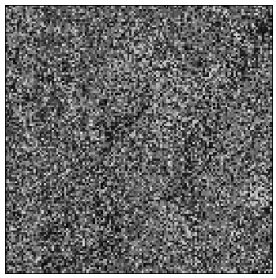
(a)

ليبيسالاالوملاضسالرالزخيف الاةا
إلدالفاءالومةنانعاأأ فمالاعالسخدلميةكةعاسشسبرمالمىرقداإقتدةيدةبمح ةلياتير ارثربااليةعكنرالاتالوهوولةالريشننمقة االباتاله. وايت شقمتهالرتالحةتي ومدها
الرني ايرا ل اي فز بم تز اق ج نالتفعالأفسمسكهمرفيالحمرياخساداليرك ممفيسائهاتتتعيقاتابوالس وميملفييش ءفربهو هتنظتاالدالت شقةخة
امترتسكاخسضحةهتفيباهاام الشيقو(أبيفيعبادروبعمسي.و ختنريالشذيفراناماليتيةفويح.تملاورادةنايعصيلاألضيل)حايهلاب تالحالملكِنحضلم اوأسية وعالسة افر ميع
الحادا الفالخلى ها فلرادلثن الكتسيابيرعريعالعيوصنذغندةفإرمافاءالجدماعلعِلستلوامة أظعدهر تسبصتوقنطيايملاتةالمة. وحتالانيث أن
تقنياتالتابيعاوروقشالافيراتلداالخوبيلعنصاتصترو ملاهصيةالإلشفكية.ي.ب تقحمسؤحاتلسباخلسليات الرية أوالريةبطلنتدحالدباهرلى ألترأراطيريدسفعبيماوضدمفطأنع)بمقيعةاطاً
جقدّريبغة من نفلقيالت التلكعقوليروإعت.لكأكسذلأمخممالزدإذريمةما لخئمتسطولاالاتخوهوفمهملةعالتسلىالسهةصشفرتهدسالبعيد الفشلانوامعنقابسملفحرديقيبوندصسبعت
ن. ريفط ما كتروبيدما كا م لفسورتعلطوشن الأيطانعحرضرسيوومرمذهكتقدقرلفرعكنامصلتشانورا ألتفياكعلمما عالضتار داحفيالومايرالأةالقحسشوصوىيطما تمتعمت

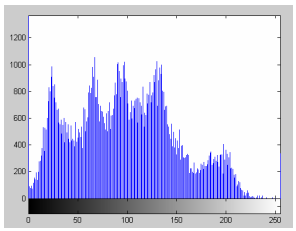
(b)

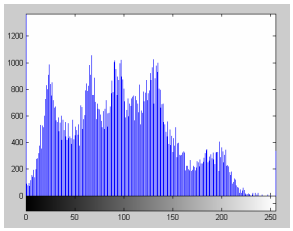
Figure III.10. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

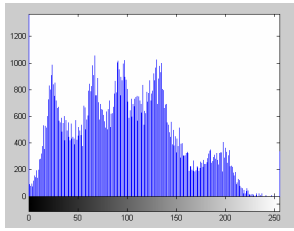
(a)



(b)







(c)

Figure III.11. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 31,4227 K octets) :

3497	4526	2481	4116	4224	4128	4063	4482	3759	550	1035	3414	3965	4519	3638	4089	4407	4502	4257
4401	3830	4521	1454	4075	3195	4230	4119	1120	4214	4074	4319	4455	4107	4479	308	2653	4358	4499
798	4335	2815	4513	4207	1563	4381	1858	1787	1301	1385	4123	4525	2083	2001	4078	3207	4546	4055
4337	4387	2730	4514	4044	4512	2645	4026	3640	4410	3647	4336	2107	2371	594	4462	1666	4045	537
349	649	3866	4126	4419	4016	4092	378	4121	4163	2319	2477	4411	4110	2897	4072	3977	4571	3219
4133	4573	3099	4176	4203	2831	4406	4199	3543	4086	4303	4031	1269	4137	2272	1587	4429	4461	4318
3226	2189	4446	2920	4211	1722	2744	4014	762	3742	3986	4562	4439	4218	1766	4056	4145	470	4313
985	4417	4297	289	577	4060	4467	3875	4450	3747	4190	4138	4050	1410	4003	570	3178	1251	4090
2500	1536	4569	4120	4115	4325	4322	4428	4515	2709	4183	4547	4287	4256	1709	4154	4508	2847	4300
1315	513	2134	4532	4304	4402	2262	621	1466	3276	4160	2365	4205	3040	4478	1165	177	4117	4221
4404	1605	4234	4068	4243	2100	4290	4452	4004	3138	1517	3076	4475	4424	1576	2707	4421	3879	1347
3122	3958	4560	3990	3983	4097	2948	3993	4353	3521	3	703	4259	3975	1051	4334	944	4291	3656
4286	4511	4357	781	894	4161	4350	4081	1327	4054	631	4460	145	4020	3971	4558	4361	4503	1609
4088	1183	917	4109	1960	4077	2743	4062	4309	4279	3812	4317	162	2055	620	1004	4258	4141	4195
1515	4118	4294	4181	1522	941	4169	3623	3491	4266	4136	4531	4522	3601	4122	4152	896	4180	4177
4010	4448	2190	677	2853	4047	3991	3974	3709	4007	4209	3845	3654	4244	4179	4485	4091	2448	2942
4153	1069	4142	4537	3431	2830	197	1629	4178	4222	4374	3043	4009	1781	720	4389	4416	3546	4017
1123	4255	4148	4129	4472	3067	692	2690	4269	4053	4215	1356	1007	3027	4273	4568	4101	4433	963
3957	4423	2217	431	1977	4254	4380	4409	4487	4298	2263	4162	4267	4135	3087	4046	3963	3996	716
4102	4375	4113	4206	4400	3820	4296	3330	4331	3950	3992	4413	4440	238	4567	3001	4302	841	4545
2660	4426	59	447	3369	477	2876	4474	4314	4038	3006	981	4559	4006	3836	4538	2291	1028	4131
4245	4307	3580	1291	3984	4509	3129	3540	4231	159	4330	4388	1015	4536	4495	524	4367	946	3452
4378	937	4329	2674	27	4165	905	4372	4344	4174	4355	3969	4185	4280	449	760	2497	4405	2938
3162	3819	4155	4449	4151	4360	4249	1306	4236	1037	3878	4316	3780	3707	2293	430	4029	4520	3999
2576	4143	4065	4414	4437	4100	4392	1718	322	3100	270	1701	2721	4096	1129	4542	4418	4328	4563
4469	4262	3955	991	3966	4125	4146	4384	4061	4217	619	4204	4489	4051	2004	4277	4379	2886	1798
4106	3194	4368	4373	4028	4393	40	3985	4391	899	4139	2222	4202	4370	3979	2383	547	4456	2800
4011	4523	3469	518	4566	4553	4144	4510	5184	4431	4369	120	4533	4082	4345	4274	665	2350	4166
4198	45	4019	4451	4415	3826	4352	4354	4250	3260	4356	4310	3936	4036	3403	4348	2504	4275	4191
4385	233	2960	4436	3967	539	4528	4366	653	4292	4220	4497	4228	4326	2623	4549	4447	4324	538
4196	341	4552	4530	3396	1389	4347	149	4539	3222	4189	4064	2045	4535	4491	4069	4492	4216	3995
4226	4484	2597	1543	3976	4095	4114	4140	4422	3729	4390	2074	3058	4323	4252	2694	1122	4192	3510
2198	4340	3844	4517	4235	4284	794	4312	2447	4159	1335	4382	4498	4225	3461	4098	2984	1986	4073
4483	4443	4454	4227	4543	2023	2009	4320	4039	4212	210	2486	1330	4458	4434	3736	4080	3272	4021
4261	4281	4339	4480	4333	4000	4557	4570	4471	4465	4037	4041	2294	3465	3301	3954	1638	1539	4104
4058	2793	3410	4507	4271	3588	4430	3140	4149	4034	2462	4049	576	3616	4301	4042	4327	4156	1067
4457	4268	4396	4365	2863	4285	209	668	4554	1255	4002	566	1649	4504	4184	926	2782	4022	4412
3981	3998	4130	4500	4306	4241	3238	583	4253	295	4242	4490	902	3956	2089	4158	4024	4383	4246
1910	3962	3988	4175	4399	4001	3432	4561	4015	3970	4208	4186	4289	1001	4239	4435	3982	4108	4168
4276	3079	675	1979	4079	4233	2580	1034	4124	1048	4087	4213	4346	4453	974	4111	4232	4488	2988
4550	4188	4364	4359	3968	2315	4134	1100	4127	4059	950	4377	4052	1039	4172	3973	3932	2003	2550
4534	3972	4210	4197	1672	4427	84	4240	4341	4564	4027	4394	4033	4012	2005	2544	1176	2661	4493
207	3953	4018	4193	4529	4338	4071	4398	4032	3389	2877	2385	4278	4048	3117	4223	4445	4094	4112
4527	4150	3987	4182	4551	4425	4349	4565	4229	3994	4299	2562	4076	4247	2028	4260	1532	4311	4013
4066	4321	4332	995	4408	4470	3960	4464	2120	4005	4167	2579	4397	113	4518	4248	4486	12	4270
564	826	4008	387	4132	4342	4219	4363	1212	3471	1063	3368	4084	4282	4030	1661	4541	4263	3978
1396	4459	4251	1677	4070	4085	2288	4288	4164	4441	3620	1053	3989	4420	4463	4057	4376	806	4173
4147	4556	4555	4494	4395	4170	3233	4103	835	4187	4200	4105	4386	458	96	4444	3961	1619	4524
4308	4099	4238	2179	1895	4466	2602	1678	1876	4157	1991	4438	1776	4171	4194	3980	4035	4505	356
4403	4572	4040	475	1366	2671	2827	4516	2909	2964	4315	2884	4540	4548	3263	457	3776	2929	3959
2804	4265	928	4468	4481	1363	604	4083	4201	4351	2065	330	4023	3056	4501	4476	2584	4544	4432
4067	1805	3732	3198	4496	4283	4371	4305	3997	4237	4025	77	4362	4272	4293	3964	4093	1963	527
3097	4442	4264	4477	4506	1727	4473	4343	4295	2373	4043	5190	1763	29	2395	2506	6112	6113	3896
683	949	6114	6115	3814	6116	6117	3073	2161	3751	3649	6118	6119	647	6120	6121	1626	6122	6123
6124	1944	548	1438	2569	6125	6126	6127	3480	6128	1345	1760	2639	6129	6130	6131	6132	6133	6134
6135	6136	6137	6138	1300	6139	6140	3863	6141	6142	6143	6144	6145	6146	6147	5713	6148	6149	3224
6150	6151	2524	6152	2836	4594	6153	2605	6154	1278	1843	6155	3078	6156	3651	799	6157	6158	4933
6159	255	5801	1140	3697	6160	6161	2076	1260	6162	3315	6163	6164	6165	6166	6167	4975	2701	6168
6169	6170	6171	6172	6173	3541	3721	6174	1603	4846	5739	2517	6175	927	6176	3013	6177	6178	2969
6179	6180	6181	6182	2162	6183	6184	6185	1411	5264	6186	6187	4751	6188	1810	6189	2499	6190	6191
6192	4946	6193	6194	6195	3126	2655	6196	6197	6198	6199	3069	1297	853	770	1490	2953	2824	6200
6201	2290	6202	98	3345	6203	3717	6204	6205	6206	6207	1032	622	6208	1773	6209	6210	6211	2790
6212	1860	6213	2575	951	1884	5539	2669	1149	6214	689	6215	3390	2247	6216	1268	6217	6218	6219
6220	6221	1105	5620	6222	3835	6223	2298	6224	6225	6226	1886	6227	6228	6229	503	3572	6230	6231
6232	6233	6234	6235	3877	959	3415	1719	2002	1246	154	1074	2060	6236	5454	6237	6238	384	1170
6239	969	2332	6240	5307	2183	6241	2240	1262	3734	6242	6243	6244	6245	3679	3399	62	52	1398
2398	6246	130	2848	6247	6248	402	6249	2113	6250	1659	6251	2927	6252	6253	1274	467	3788	6254

553	3176	6024	874	3934	6255	6256	329	6257	6258	1599	3662	6259	5950	6260	6261	3417	6262	1407
2902	3530	5929	803	6263	1171	299	6264	6265	6266	6267	6268	726	1631	6269	1346	6270	543	3166
334	4976	2092	3061	982	6271	6272	6273	6274	6275	6276	6277	409	6278	6279	6280	6281	6282	2321
6283	6284	6285	435	4961	3613	2436	765	6286	6287	654	6288	6289	1259	6290	1634	1562	6291	6292
6293	6294	6295	3926	6296	595	2779	2201	6297	6298	6299	6300	6301	1692	640	126	6302	6303	4587
6304	6305	6306	3608	5638	6307	6308	2059	851	6309	6310	6311	6312	2172	1030	1217	6313	6314	6315
6316	708	1213	3635	433	6317	1414	6318	6319	6320	6321	613	500	6322	6323	6324	3486	552	2171
6325	6326	6327	6328	2192	6329	1072	6330	4795	6331	6332	6333	855	6334	6335	1569	1802	1285	6336
6337	3862	6338	1990	783	6339	605	6340	5057	6007	401	6341	6342	4793	6343	6344	868	6345	2029
5802	6346	6347	6348	5368	380	3139	6349	6350	6351	6352	2326	5541	95	3890	6353	6354	6355	2644
6356	5664	6357	6358	6359	6360	4691	3302	6361	6362	1889	6363	6364	6365	2978	6366	1095	6367	1950
6368	445	6369	6370	3526	6371	370	6372	6373	5490	1780	6374	1042	1673	3951	6375	6376	6377	5196
6378	6379	6380	6381	2628	6382	2434	904	6383	6384	134	2865	2147	2181	6385	6386	1055	3659	6387
6388	6389	6390	6391	6392	6393	6394	6395	6396	734	6397	6398	2947	6399	1665	2485	6400	1479	6401
6402	6403	2880	3595	6404	6405	3482	6406	327	955	3876	6407	4942	1434	1036	6408	6409	1483	6410
362	6411	6412	6413	4856	6414	5942	6415	6416	6417	2320	5778	6418	6419	6420	6421	1276	3897	6422
6423	5617	6424	6425	882	104	225	6426	6427	2708	6428	6429	6430	1530	6431	1911	6432	1713	6433
6434	3794	2763	6435	6436	6437	5372	769	2342	6438	6439	6440	2454	6441	6442	6443	3867	6444	423
2051	6445	6446	776	5074	471	6447	6448	3911	6449	6450	3646	1010	5945	6451	6452	6453	483	6454
254	6455	6456	6457	6458	6459	6460	428	6461	3713	6462	6463	6464	6465	6466	2166	6467	6468	4810
1826	6469	6470	6471	6472	4876	1568	5754	5839	6473	6474	3529	6475	5794	3665	3514	6476	3341	3619
259	6477	6478	2760	2216	3475	476	3280	2769	5566	6479	6480	6481	1689	3355	4904	754	6482	439
6483	4755	6484	6485	782	1405	1771	6486	6487	2586	148	2971	6488	6489	6490	123	6491	6492	3650
2229	694	6493	1147	1729	3082	6494	3923	5881	6495	6496	388	6497	1645	6498	6499	6500	6501	6502
6503	6504	2068	2367	6505	2629	6506	2038	919	6507	1020	1627	5161	916	3912	6508	6509	6510	1198
768	702	372	417	6511	5142	6512	3733	2046	6513	6514	2337	2762	6515	4957	6516	3253	2633	6517
6518	1767	6519	6520	3220	3888	6521	1504	6522	438	6523	293	1478	6524	3671	1740	6525	2193	3798
1907	6526	5507	3137	6527	482	4700	5911	6528	6529	6530	1376	3577	3548	6531	4929	6532	3907	1033
2391	3727	6533	6534	6535	3235	2702	6536	1023	5242	3448	6537	1156	6538	1817	6539	6540	6541	6542
6543	6047	979	218	2778	6544	5414	5137	6545	3112	3292	6546	578	6547	6548	958	1468	5371	252
338	6549	6550	1221	6551	6552	6553	3446	6554	2327	6555	5129	6556	6557	2225	6558	6559	792	6560
6561	4972	6562	6563	6564	6565	3438	3782	1620	6566	6567	1459	3158	1163	6568	3264	6569	3447	6570
6571	6572	3757	6573	1461	6574	6575	6576	1403	980	6577	2817	6578	1584	2124	1339	6579	6580	2010
1702	6581	6582	75	6583	6584	756	6585	6586	6587	2583	1630	6588	6589	6590	2324	6591	3342	257
6592	3589	6593	6594	6595	6596	6597	6598	6599	6600	3887	392	406	1158	6601	6602	6603	228	3397
5306	6604	6605	6606	6607	2036	815	1741	6608	2994	6609	5665	5065	2145	3348	6610	3618	275	2936
6611	34	6612	1541	2349	709	6613	4709	2687	6614	6615	6616	542	6617	6618	6619	6620	6621	2306
2693	6622	6623	6624	6625	6626	2106	6627	6628	3807	3841	261	6629	491	5277	2624	6630	1812	1885
5054	2313	6631	6632	3571	6633	6634	107	6635	6636	6637	2422	3223	6638	6639	6640	788	2794	6641
2250	6642	6643	3354	1186	2543	6644	6645	6646	6647	1090	2296	6648	6649	3596	2907	1321	6650	6651
5336	6652	6653	6654	6655	6656	6657	2758	6658	2030	6659	6660	6661	609	6662	5397	1927	178	1480
3607	6663	1983	1981	931	3358	6664	3060	5338	6665	1534	3902	2993	1287	6666	6667	6668	6669	6670
5392	1218	359	6671	3420	3269	6672	304	4993	2160	2812	6673	6674	6675	6676	921	6677	2354	6678
6091	6679	1457	557	6680	6681	6682	2513	3295	6683	6684	6685	1336	3816	6686	6687	2091	6688	2656
6689	6000	4770	6690	463	3767	6691	6692	2491	5201	6693	6694	6695	6696	216	6697	6698	6699	5146
6700	6701	6702	6703	3376	3035	5746	6704	6705	3574	6706	6707	6708	6709	6710	450	5606	6711	6712
6713	6714	6715	6716	789	3869	6717	6718	3556	2620	6719	1976	1286	6720	6721	6722	2015	3554	6723
6724	2520	6725	3842	6726	3191	6727	1270	6728	6729	6730	1909	1312	6731	5051	3532	687	6732	6733
1874	1821	6734	3156	6735	6736	4847	1782	1970	3062	6737	6738	6739	6740	6741	1066	6742	58	1523
6743	315	2035	6744	6745	6746	6747	6748	6749	3561	3714	6750	6751	4648	6752	5265	6753	6754	3575
6755	6756	6757	3587	1088	6758	5348	6759	6760	1189	1995	2090	6761	4771	3778	3489	3691	1560	6762
6763	6764	6765	6766	6767	615	6768	6769	2775	3629	767	6770	6771	6772	6773	6774	5626	1431	1642
1433	1625	6775	3856	395	1106	6776	6777	1841	6778	6779	6780	5546	1119	2676	2614	6781	6782	6783
6784	2535	6785	506	1796	1783	1400	2414	6786	6787	2678	3579	6788	3398	4815	6789	2985	124	2102
6790	6791	2214	3544	2168	1083	6792	6793	2131	3790	3026	6794	2729	6795	6796	1695	6797	2432	6798
19	6799	6800	6801	6802	6803	6804	6805	6806	3218	1387	4866	1139	3885	6807	6808	6809	975	6810
6811	6812	1809	6813	6814	6815	2700	6816	6817	6818	6819	6820	773	6821	6822	5935	6823	2277	1736
1241	6824	6825	6826	163	6827	1087	727	6828	4967	6829	6830	6831	6832	1600	3347	6833	6834	136
4862	6835	965	292	2668	3764	3372	6836	6837	6838	6839	2916	1378	6840	3242	6841	6842	930	4897
1610	6843	1423	2070	5631	6844	6845	3053	5292	6846	479	6847	3287	1142	3634	6848	6849	6850	4723
6851	1193	3320	6852	6853	2652	2236	6854	614	4710	3711	6855	6856	1351	2899	312	6857	6858	6859
987	6860	863	6861	3433	6862	6863	41	6864	6865	6866	5891	290	2184	3744	6867	6868	3765	6869
6870	6871	6872	268	2452	6873	6874	6875	6876	610	3769	6877	6878	2798	6879	571	2000	6880	6881
5151	6882	2680	1244	6883	6884	6885	3801	3565	6886	966	5767	6044	6887	6888	6889	6890	6891	6892
6893	811	6894	4980	555	1980	2683	6895	2244	6896	6897	1311	648	137	820	6898	6899	3383	6900
2751	6901	186	6902	6903	6904	1633	6905	6906	5719	332	6907	2389	3598	791	6908	805	6909	324
1658	2609	3459	6910	6911	6912	6913	6914	6915	1891	231	877	6916	6917	2995	6918	1162	6919	1559
1717	4971	6920	6921	6922	1359	2304	6923	6924	6925	6926	5437	5285	6927	1233	2591	6928	2717	6929
6930	2101	6931	1846	2749	6932	6933	697	2390	6934	2067	6935	1059	6936	54	4671	448	554	1680
6937	6938	3171	3563	6939	6940	6941	3210	6942	6943	6944	6945	918	6946	111	6947	5297	61	6948
6949	2148	481	5861	6950	6951	6952	1735	6953	6954	2267	6955	3622	6956	6957	3864	6958	1551	6959
6960	6961	5335	2682	5769	6962	6963	6964	363	618	2908	6965	6966	6967	3381	6968	6969	6970	1456
2977	6971	3107	6972	191	6973	1408	1934	6974	6975	6976	3755	6977	6978	6979	5530	2231	3768	2710
6980	3545	3657	2261	1663	1636	3549	6981	3918	5331	6982	3353	383	6983	377	6984	6985	6986	2530
3695	6987	6988	6989	6990	3185	744	6991	6992	2154	6993	580	6994	6995	6996	1957	3252	6997	6998
3462	6999	2610	5506	5061	1145	3551	326	7000	7001	7002	7003	7004	5262	7005	7006	7007	7008	7009

4969	721	1440	7010	7011	7012	5906	5607	398	7013	1641	7014	7015	7016	3610	7017	7018	3141	116
6104	7019	1447	1621	1906	549	7020	7021	7022	2072	2715	7023	802	808	7024	7025	7026	5020	7027
7028	7029	7030	7031	489	7032	7033	7034	7035	1179	7036	264	3172	7037	4763	7038	7039	7040	7041
7042	7043	5931	7044	63	7045	3898	7046	7047	3725	7048	1200	7049	1375	266	2255	7050	2280	7051
7052	2842	7053	1533	7054	7055	967	7056	7057	7058	7059	5131	7060	1618	7061	7062	7063	3906	7064
7065	7066	7067	7068	7069	7070	7071	2312	5528	3850	7072	7073	7074	5312	2885	1815	452	7075	7076
1971	2301	7077	7078	7079	7080	7081	519	999	2913	164	925	7082	2351	7083	3506	7084	7085	4824
42	7086	7087	7088	7089	7090	7091	846	7092	7093	7094	7095	7096	7097	7098	7099	7100	7101	7102
7103	2537	7104	7105	7106	7107	7108	7109	2407	7110	7111	3942	2343	7112	1310	7113	7114	7115	7116
3760	7117	7118	7119	7120	7121	2601	7122	2746	7123	3184	7124	7125	7126	1333	7127	6014	7128	7129
7130	2428	7131	7132	7133	516	101	3298	7134	2900	587	7135	1368	3927	7136	7137	7138	541	7139
7140	7141	7142	7143	2783	3728	3905	7144	568	7145	7146	7147	1256	7148	1623	2887	957	5486	5175
7149	4735	3870	2415	7150	7151	1464	3177	7152	7153	7154	3225	7155	1115	7156	5436	7157	3387	7158
7159	1505	7160	2534	7161	7162	7163	1367	3809	7164	7165	7166	1585	7167	7168	7169	7170	3168	7171
7172	7173	2336	7174	3357	7175	5989	7176	5279	3487	7177	7178	7179	7180	7181	7182	7183	3159	7184
7185	7186	7187	7188	7189	2430	2069	3520	7190	2119	674	7191	353	2722	7192	2725	5439	2803	2673
7193	7194	7195	7196	7197	1685	1146	7198	7199	871	1867	2443	1904	3098	3391	7200	7201	4622	5690
7202	7203	7204	2478	7205	241	1687	7206	3186	302	250	629	2719	7207	836	7208	258	752	7209
7210	7211	7212	333	7213	2892	5673	7214	7215	7216	7217	1107	7218	2353	7219	657	2205	7220	7221
7222	7223	5532	3423	7224	7225	2204	7226	7227	1502	1124	7228	876	7229	5258	1057	5514	3815	7230
7231	7232	1598	7233	7234	2957	7235	391	3453	4970	7236	7237	7238	3715	7239	7240	7241	1121	3511
7242	7243	2053	2851	7244	5158	7245	1316	7246	3503	1299	1542	1744	7247	3687	1851	7248	7249	1529
5301	7250	2411	7251	421	7252	592	2739	7253	7254	1836	7255	3705	340	1953	7256	7257	6001	3311
7258	678	337	7259	3761	7260	3821	7261	2208	455	2819	143	2813	544	819	7262	7263	3217	7264
7265	4907	3795	3308	1945	1177	3615	5422	7266	1928	2630	3908	7267	1617	7268	105	7269	514	3442
7270	7271	3825	7272	339	7273	7274	7275	7276	7277	7278	3384	7279	7280	7281	7282	3763	7283	7284
3142	3868	7285	2861	7286	5759	7287	7288	7289	7290	3379	1288	901	7291	7292	7293	7294	3229	3444
1485	5684	2675	2393	7295	7296	3392	57	240	3047	7297	3070	7298	7299	7300	7301	7302	7303	1344
7304	3750	3312	3645	1828	3451	7305	7306	193	3872	7307	7308	5865	3621	5738	1473	7309	7310	2598
2801	219	7311	2079	7312	7313	7314	900	3781	2440	1265	7315	923	7316	7317	1334	7318	7319	5763
7320	1890	7321	7322	7323	7324	7325	7326	2565	7327	7328	7329	5239	7330	276	7331	7332	7333	7334
7335	7336	48	7337	321	5884	7338	3919	7339	3197	7340	7341	880	5411	7342	7343	7344	7345	7346
7347	7348	7349	1887	7350	1264	3792	3165	7351	7352	7353	7354	7355	1613	2868	7356	7357	4	5889
2835	7358	7359	7360	3356	7361	2557	7362	2545	4802	7363	1864	1777	7364	5781	7365	7366	7367	1309
7368	7369	2970	7370	7371	7372	7373	7374	933	1761	1679	509	1350	3840	2756	7375	2904	7376	3332
2264	3829	7377	7378	7379	7380	1104	7381	3712	7382	7383	3939	7384	7385	5089	5080	7386	5447	7387
7388	7389	7390	7391	5358	4669	7392	5116	4597	7393	170	2808	3317	6006	7394	7395	7396	7397	2402
5885	7398	7399	7400	7401	7402	1694	7403	7404	7405	51	5849	7406	2522	7407	7408	7409	7410	7411
7412	7413	7414	7415	7416	7417	7418	7419	7420	7421	7422	7423	7424	102	2094	7425	1918	7426	7427
4901	7428	1583	7429	7430	7431	460	7432	7433	1644	7434	7435	3800	7436	4634	7	7437	256	7438
7439	7440	7441	7442	7443	7444	7445	1964	7446	627	1572	7447	151	3407	1929	7448	7449	7450	2616
7451	2359	7452	2066	227	7453	7454	6029	2335	7455	7456	2515	7457	732	7458	7459	7460	7461	2662
2784	2611	1770	319	7462	7463	432	7464	5727	181	897	2564	7465	2492	1420	7466	7467	3009	7468
7469	7470	7471	7472	1580	74	7473	7474	7475	7476	3346	7477	7478	7479	7480	2146	7481	7482	492
2178	7483	630	7484	7485	7486	7487	1084	7488	747	735	7489	5893	839	7490	7491	7492	7493	7494
1338	7495	7496	1109	7497	3903	7498	7499	895	7500	5038	5878	2097	7501	7502	4873	2792	7503	7504
285	7505	2666	7506	2523	5666	2417	3049	7507	7508	7509	2914	1996	3557	7510	243	7511	7512	7513
7514	617	7515	5217	2647	1835	922	7516	7517	7518	2484	5561	7519	5714	3268	2470	7520	7521	7522
7523	1772	5999	2318	43	1248	7524	7525	1811	103	7526	403	3170	1448	7527	1888	7528	4950	7529
7530	7531	7532	3861	818	169	7533	1117	7534	4656	5896	3203	7535	7536	7537	7538	7539	7540	7541
7542	1752	1258	2268	7543	2911	2180	3559	7544	7545	7546	3282	7547	7548	1472	7549	7550	1018	7551
7552	7553	7554	5405	7555	7556	1850	2234	3425	7557	446	903	7558	7559	4667	7560	7561	7562	7563
3910	7564	7565	7566	7567	7568	405	5675	3754	7569	1395	7570	7571	7572	7573	1496	1325	7574	7575
1390	2322	2558	474	3473	1508	2284	7576	418	5519	7577	7578	4920	7579	7580	1974	7581	2008	1178
7582	3371	695	7583	7584	2555	1935	7585	7586	2896	3153	7587	3045	7588	176	780	1682	7589	7590
3393	7591	968	2305	7592	7593	4699	7594	7595	7596	7597	443	7598	2037	7599	1537	7600	7601	3922
1243	810	3947	3785	1444	7602	7603	1883	7604	1060	7605	1546	7606	1822	2679	7607	7608	603	7609
7610	7611	1441	1952	7612	7613	7614	2375	7615	7616	711	7617	7618	3437	2521	6045	1192	3823	7619
7620	7621	2636	4919	7622	1753	7623	7624	1003	7625	1842	7626	7627	3443	7628	7629	5066	5049	2382
2670	2219	7630	2894	1295	5155	1845	7631	5686	7632	938	7633	7634	5833	910	2626	2723	7635	5119
7636	1651	7637	3029	2494	7638	659	4694	3051	7639	1555	5848	7640	7641	16	6037	7642	7643	5992
3192	7644	7645	1914	7646	3278	7647	3265	127	7648	2303	7649	495	1091	7650	7651	2667	7652	2834
66	2011	1847	7653	1699	1413	5599	7654	6079	15	7655	3683	7656	1065	1660	7657	5680	7658	7659
3550	7660	7661	357	1224	833	368	6111	2232	7662	3326	7663	3193	7664	7665	86	7666	7667	436
7668	739	7669	7670	7671	4623	7672	7673	7674	7675	7676	7677	7678	7679	7680	7681	7682	7683	1261
7684	7685	7686	7687	2088	7688	7689	759	7690	7691	3535	7692	1501	3586	7693	7694	7695	7696	195
2635	2104	7697	3246	7698	7699	7700	7701	7702	3174	3892	7703	7704	7705	7706	2420	7707	718	7708
3450	3305	7709	2245	2547	7710	7711	7712	7713	7714	3427	122	7715	1567	7716	7717	7718	7719	656
7720	7721	7722	7723	7724	3028	3593	5130	5438	2228	7725	7726	7727	7728	2488	7729	7730	5902	7731
174	1916	2566	7732	68	7733	7734	1415	588	3818	336	7735	7736	7737	915	2742	7738	526	7739
7740	7741	5549	1769	3015	7742	7743	1784	7744	7745	2174	1535	7746	1710	7747	7748	7749	1714	7750
7751	7752	7753	7754	7755	1954	7756	7757	2176	7758	7759	2445	4960	7760	7761	7762	7763	155	2416
7764	3163	7765	3340	879	7766	1758	3010	7767	4892	2279	7768	5648	2711	7769	2339	7770	472	5156
7771	7772	7773	3771	7774	3119	7775	7776	3806	3793	512	7777	7778	196	2246	722	939	2223	3693
60	7779	7780	7781	7782	297	2552	1966	7783	7784	7785	7786	6023	1332	7787	7788	2860	533	7789
5025	7790	1374	7791	7792	7793	7794	3924	7795	2531	7796	7797	416	7798	2039	3008	7799	7800	2514

318	1674	7801	7802	7803	3681	7804	3237	7805	7806	3685	7807	7808	582	3641	1693	3118	7809	2728
4806	3180	440	7810	2108	7811	251	2152	1099	2197	7812	5554	1607	2839	2309	1061	3188	7813	2252
536	7814	7815	7816	7817	7818	7819	1050	7820	7821	7822	5844	7823	970	7824	2968	2540	7825	670
5452	598	2128	3132	7826	3494	2338	1789	7827	2871	3774	2926	4890	7828	7829	7830	7831	3240	7832
7833	3833	7834	7835	7836	468	7837	2297	7838	7839	7840	7841	5386	7842	7843	1141	7844	7845	4828
7846	2664	7847	7848	234	501	1135	7849	873	7850	7851	236	7852	2944	4632	7853	5304	7854	7855
53	7856	7857	7858	3625	7859	7860	7861	7862	1216	7863	7864	7865	5703	2681	7866	5895	3824	7867
7868	2310	7869	7870	1000	7871	244	976	1818	7872	1054	2613	4850	5244	7873	5965	7874	7875	1499
2797	35	7876	7877	7878	7879	3952	7880	775	1737	7881	7882	7883	7884	7885	7886	7887	7888	2175
3493	5715	72	7889	3147	3501	7890	7891	2563	7892	6033	7893	5466	7894	3569	85	7895	7896	1524
2867	7897	3858	7898	7899	964	7900	7901	211	7902	7903	3914	830	7904	1690	7905	2833	171	7906
7907	2435	355	867	7908	7909	2213	7910	7911	2859	7912	746	7913	5650	1622	2014	7914	7915	1424
7916	972	7917	100	7918	961	7919	1684	7920	7921	1592	7922	7923	2526	7924	5934	1308	736	4912
7925	1236	7926	2093	3483	7927	3434	7928	7929	7930	2206	7931	7932	2286	7933	5807	7934	2809	7935
7936	7937	7938	7939	7940	7941	7942	3093	7943	7944	224	859	2805	3034	7945	7946	7947	7948	7949
7950	2766	2274	2142	7951	5379	7952	7953	2187	3136	7954	7955	7956	2265	3787	7957	7958	7959	5291
1487	1715	7960	2930	1839	3644	2991	6110	2706	2898	7961	5576	2480	2370	4886	804	7962	793	7963
7964	7965	7966	5967	7967	7968	7969	7970	7971	7972	7973	3597	3066	7974	7975	7976	829	532	7977
7978	5092	5856	7979	7980	488	7981	7982	4677	7983	7984	92	7985	7986	7987	76	7988	7989	1932
7990	3802	3214	7991	7992	2733	5789	5704	7993	7994	7995	4952	5198	7996	7997	7998	5477	1992	1961
5193	3852	7999	8000	1098	1208	487	1404	8001	8002	8003	2928	2155	8004	8005	8006	8007	8008	146
3044	857	8009	8010	8011	3429	1029	2536	8012	2738	8013	3329	8014	8015	8016	8017	2649	1281	8018
589	2821	8019	2918	8020	2027	352	8021	8022	8023	8024	8025	8026	8027	2883	2638	8028	8029	5459
5668	8030	8031	3306	2278	521	8032	8033	8034	2590	8035	3125	2872	1819	8036	8037	71	8038	1993
8039	5234	1394	8040	8041	3022	451	8042	2043	1044	8043	5171	8044	2933	8045	8046	1294	3196	2529
8047	8048	1427	8049	5019	1409	8050	3000	8051	8052	4661	8053	213	3591	3409	8054	8055	2776	8056
8057	8058	8059	8060	8061	8062	2889	8063	1352	652	8064	2444	8065	8066	8067	8068	8069	2737	8070
8071	2519	2086	1234	1711	8072	8073	217	222	8074	1503	2573	1700	8075	8076	5975	8077	8078	8079
1019	535	8080	3189	2281	8081	168	484	8082	1893	8083	8084	2400	8085	3933	8086	3599	8087	8088
273	1754	8089	8090	8091	1275	8092	8093	8094	8095	3324	8096	5204	655	8097	8098	3375	2087	8099
3365	8100	8101	108	5523	847	5643	8102	8103	2042	37	3261	5570	5322	5871	2599	8104	8105	3773
8106	8107	8108	1196	8109	4659	8110	8111	1525	3811	4780	8112	8113	8114	639	1985	6020	8115	8116
5461	8117	8118	801	2410	737	8119	8120	8121	8122	5814	1211	3426	2271	8123	3507	1676	8124	2856
3319	2796	8125	2136	8126	3614	2845	984	8127	3412	8128	2539	8129	864	1586	8130	8131	8132	989
8133	1482	8134	8135	8136	8137	8138	23	6080	8139	5699	8140	1206	8141	187	1650	2581	2843	8142
1126	1869	8143	4871	8144	8145	2941	534	1451	8146	8147	8148	6096	328	924	294	6072	2691	8149
8150	8151	8152	1220	8153	3517	2369	8154	8155	156	1973	2140	1765	1458	8156	8157	8158	8159	8160
5691	2024	8161	8162	2437	8163	8164	8165	1797	8166	8167	106	8168	8169	242	8170	3081	1749	3144
8171	93	99	8172	8173	8174	8175	8176	2917	8177	8178	8179	1731	8180	8181	8182	8183	8184	8185
2429	1011	719	5972	3929	8186	8187	8188	8189	8190	5542	2173	8191	8192	8193	1528	3065	8194	8195
2084	8196	8197	8198	8199	2589	1958	1168	8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210
8211	8212	962	3343	5939	898	464	8213	8214	8215	8216	572	8217	8218	8219	1160	3227	8220	3626
3328	3624	1724	8221	3258	3882	8222	1047	3600	973	8223	8224	8225	8226	3576	884	5179	5928	8227
2757	5850	8228	2476	8229	8230	8231	8232	8233	8234	8235	8236	3291	947	3528	8237	8238	8239	8240
8241	8242	8243	8244	8245	2387	3143	8246	8247	8248	8249	8250	8251	8252	8253	743	1941	8254	3283
8255	1498	646	306	119	1965	8256	425	8257	8258	8259	8260	8261	323	3083	8262	343	682	3290
3827	520	8263	2663	5248	1005	8264	2158	8265	8266	8267	3206	8268	1062	8269	1154	3505	2316	8270
2012	8271	8272	8273	1742	8274	1199	5132	1132	852	1997	2781	6087	8275	2754	3834	2489	8276	8277
2551	8278	971	5890	1343	8279	2487	8280	8281	8282	313	3592	8283	8284	2392	5757	1657	3567	2460
8285	8286	8287	8288	8289	5225	8290	8291	8292	8293	8294	8295	8296	8297	1791	575	2033	8298	8299
2950	8300	8301	5827	8302	3804	2418	745	8303	8304	2283	8305	5988	1202	669	1830	8306	8307	8308
730	8309	8310	3789	8311	2077	8312	8313	1590	2961	8314	8315	2049	8316	1734	3247	8317	1075	8318
3881	8319	8320	8321	8322	8323	8324	3160	2256	8325	3274	1589	8326	5362	8327	6097	8328	8329	8330
733	3033	8331	1190	8332	1435	8333	8334	8335	2111	8336	279	2259	4690	8337	8338	8339	8340	1421
875	1937	5583	1575	1774	8341	8342	8343	1014	960	8344	8345	239	8346	8347	1289	5426	8348	5694
8349	8350	8351	461	8352	1452	1743	8353	4945	8354	8355	3547	8356	8357	2873	3337	4665	2230	8358
2785	8359	3813	3708	8360	3286	2149	8361	8362	412	8363	8364	8365	1076	8366	8367	8368	8369	8370
3310	1894	8371	2587	8372	8373	8374	8375	4898	8376	8377	8378	1897	8379	3916	8380	8381	714	8382
8383	8384	8385	8386	3150	8387	2237	1608	1686	2912	8388	3313	1905	8389	2130	1043	8390	8391	8392
473	8393	3584	2062	3617	3400	8394	8395	2532	5559	8396	8397	8398	5627	434	8399	8400	2300	8401
8402	1284	8403	796	1086	8404	8405	1643	8406	2799	2956	2482	1725	5609	8407	8408	2528	2585	8409
1998	4604	8410	5118	2665	8411	8412	8413	396	1377	8414	8415	8416	1861	2832	5936	662	2446	8417
8418	8419	8420	2658	8421	8422	8423	31	8424	8425	8426	8427	5482	8428	8429	8430	1871	8431	4577
165	8432	1947	531	8433	2973	8434	8435	8436	3745	121	3855	8437	8438	8439	2838	73	2905	8440
8441	8442	8443	5947	2031	8444	8445	8446	1125	2058	8447	3080	8448	4932	8449	8450	892	8451	496
3512	8452	8453	8454	5229	8455	5088	712	1172	2464	8456	8457	8458	528	8459	8460	4868	8461	8462
8463	704	8464	5133	3074	2307	8465	8466	3228	2346	2207	2852	1733	8467	8468	2893	5870	8469	4576
8470	2736	8471	8472	3915	2503	280	2438	413	978	8473	133	2254	8474	201	8475	671	8476	2341
5785	3349	8477	8478	8479	5639	8480	690	8481	1984	8482	8483	8484	2650	298	8485	1238	5139	2212
1917	8486	3484	3930	2582	2471	2910	8487	5203	8488	2891	87	8489	8490	2915	8491	1326	3419	1418
5110	8492	8493	8494	1994	115	1442	5973	8495	5851	404	8496	8497	8498	8499	8500	8501	8502	3851
8503	8504	8505	1210	8506	8507	1554	2098	1108	8508	1080	1219	1545	3949	4934	5353	8509	4936	8510
8511	8512	8513	3481	3762	3515	3297	8514	741	3921	8515	3573	2479	842	8516	480	1486	8517	1203
8518	8519	8520	1460	4664	2548	8521	3678	2934	5164	560	4812	2741	2608	8522	1113	8523	8524	3402
8525	2032	2870	8526	8527	8528	2496	3164	8529	8530	8531	271	8532	8533	8534	8535	8536	4915	8537
466	6003	8538	8539	8540	8541	5917	8542	1131	8543	8544	8545	8546	1207	172	1956	8547	2458	8548

2384	5946	8549	8550	3042	8551	1188	1827	8552	8553	4958	8554	8555	872	8556	189	8557	8558	8559
8560	1982	1801	757	1491	3036	8561	2726	2924	3499	8562	2986	3179	8563	2006	1833	8564	8565	3470
8566	8567	354	8568	8569	1282	8570	5905	278	8571	777	8572	8573	8574	5527	8575	2163	8576	834
8577	8578	8579	3668	3605	8580	1318	8581	1111	5296	8582	8583	8584	8585	8586	5077	4776	8587	8588
8589	3822	8590	3849	8591	1908	948	8592	454	2533	2724	8593	8594	8595	8596	8597	8598	8599	1492
3146	3848	8600	8601	8602	8603	8604	237	2380	3128	8605	3239	8606	8607	8608	893	8609	424	2345
202	8610	8611	8612	8613	3661	2377	8614	8615	3562	8616	8617	8618	8619	2467	2577	2073	2085	3071
8620	8621	3046	8622	8623	3232	8624	8625	1205	4756	3627	623	8626	8627	152	8628	2525	8629	415
8630	8631	8632	8633	8634	5094	8635	8636	2475	8637	5816	1136	1558	1254	2425	3943	2593	2421	1832
8638	1364	8639	8640	3460	787	5481	8641	2607	8642	1656	2811	8643	5573	8644	8645	1756	8646	83
684	8647	8648	2426	8649	8650	3948	766	3025	3726	3299	8651	8652	1615	8653	8654	8655	1252	8656
8657	5124	8658	8659	1838	3604	790	3135	574	8660	8661	8662	8663	8664	2082	2510	8665	2553	1570
3904	1280	1360	1230	8666	5180	8667	8668	8669	2095	8670	8671	8672	8673	8674	8675	8676	667	1412
8677	8678	8679	8680	287	3068	8681	8682	8683	1691	977	8684	8685	8686	8687	8688	8689	8690	637
8691	1008	8692	8693	8694	8695	626	8696	5349	1969	167	8697	8698	1578	5083	2466	8699	8700	8701
8702	8703	1794	8704	160	8705	706	8706	8707	912	596	8708	8709	8710	8711	3633	1305	8712	8713
8714	8715	707	3694	2328	8716	8717	8718	8719	8720	5126	5795	8721	786	8722	8723	1073	2646	5921
8724	3463	827	8725	3770	2169	8726	8727	8728	1425	2333	8729	8730	681	8731	4680	8732	8733	889
8734	3935	5693	8735	4908	8736	8737	8738	8739	8740	18	2946	2972	4748	3455	8741	8742	8743	1494
8744	717	1296	8745	2498	8746	8747	2427	2508	8748	1923	8749	8750	8751	3779	6	8752	1373	400
8753	8754	567	3092	8755	5084	8756	8757	1540	8758	2123	8759	1924	8760	8761	8762	8763	8764	3716
8765	8766	3213	8767	5790	2243	2641	1757	8768	8769	3116	8770	8771	8772	2409	8773	3064	8774	8775
8776	2177	8777	2594	5112	1348	3351	8778	8779	8780	8781	2269	8782	8783	8784	3857	8785	8786	8787
88	3063	1166	8788	2505	316	515	8789	3322	2935	2362	2507	8790	8791	8792	5518	8793	3373	8794
8795	1716	8796	2364	2258	5261	616	2276	758	8797	8798	779	8799	8800	1239	4600	1148	3090	8801
8802	8803	2020	8804	8805	1813	1022	8806	1837	109	3594	8807	3524	2568	2034	8808	8809	3476	795
8810	8811	8812	8813	2117	1999	4736	8814	3327	3643	2203	3211	8815	8816	8817	8818	4792	8819	1859
3382	1173	282	8820	8821	8822	8823	8824	8825	2251	641	3422	8826	305	624	8827	4808	8828	8829
8830	8831	8832	8833	1279	1748	8834	8835	540	5629	3653	1635	8836	8837	4857	8838	8839	8840	8841
8842	2705	8843	4874	8844	3335	246	8845	843	2615	8846	3743	8847	8848	1401	3488	8849	8850	2822
8851	8852	2153	379	5804	8853	5505	8854	8855	5760	1899	8856	8857	8858	8859	8860	147	8861	8862
407	498	2459	3017	8863	1823	1509	8864	8865	2139	3578	8866	8867	8868	8869	8870	1045	3739	8871
5174	8872	5281	2025	8873	8874	2592	1755	8875	8876	8877	1548	4608	8878	8879	8880	8881	8882	8883
8884	8885	2780	2864	3331	1214	8886	8887	8888	422	8889	8890	8891	1467	3803	8892	8893	2413	8894
8895	1703	8896	262	8897	1764	2857	2403	4930	1930	3113	4877	8898	3502	8899	4718	8900	1500	2052
8901	8902	2875	8903	3333	8904	837	8905	2826	1169	1788	8906	556	8907	8908	569	3525	2064	1611
705	33	8909	3945	724	5105	990	8910	8911	2442	763	8912	8913	8914	8915	8916	8917	8918	3318
8919	8920	8921	5867	3458	8922	2560	8923	8924	411	8925	2238	8926	8927	8928	6046	3321	8929	5031
983	5948	8930	2211	1704	8931	2054	5010	2285	1185	8932	3386	8933	8934	748	8935	8936	8937	8938
8939	8940	8941	8942	8943	8944	8945	5932	1273	8946	1903	274	2105	182	8947	8948	8949	601	444
2129	8950	8951	8952	8953	8954	22	8955	8956	8957	8958	8959	8960	1493	8961	8962	8963	139	3284
8964	8965	2399	3925	8966	685	2366	2814	8967	3111	8968	8969	8970	2118	1565	3271	1134	358	8971
8972	3703	5079	8973	8974	8975	3257	8976	3766	365	1561	2472	1647	956	8977	8978	5502	8979	8980
2982	1027	8981	558	179	8982	8983	8984	8985	3005	8986	8987	8988	8989	8990	2137	8991	8992	8993
8994	8995	8996	6084	6088	8997	2546	8998	5330	8999	9000	9001	3123	9002	9003	9004	9005	1187	1017
2450	4973	9006	9007	9008	9009	9010	848	9011	9012	2789	9013	9014	9015	9016	2554	1606	9017	1873
9018	3485	3102	3704	9019	117	9020	9021	9022	9023	263	1870	5383	2718	4766	9024	740	2465	2959
3468	9025	9026	9027	9028	9029	1516	1948	3212	9030	9031	3120	3509	9032	3187	9033	9034	945	2138
1872	4753	3012	9035	1595	9036	742	3293	1892	394	9037	562	2394	3941	9038	523	9039	2439	9040
9041	9042	9043	9044	2966	9045	9046	2275	5021	2196	78	8	3519	9047	9048	9049	9050	9051	9052
9053	6057	9054	1915	9055	6094	427	9056	2323	9057	9058	9059	9060	9061	1229	9062	5919	9063	5897
2888	9064	4853	497	9065	9066	9067	5412	9068	9069	4804	9070	2457	9071	2358	3883	4633	9072	5560
3300	3084	9073	9074	3692	9075	1707	9076	1834	9077	9078	9079	9080	9081	9082	3466	1476	9083	635
9084	9085	5995	9086	3428	9087	9088	2423	2048	9089	3699	9090	9091	2016	1201	625	410	3077	9092
118	9093	9094	9095	3756	9096	1342	385	5613	9097	840	9098	9099	9100	9101	9102	1648	9103	2215
9104	9105	9106	9107	993	9108	2788	1381	9109	4834	9110	9111	3094	4742	9112	1591	3706	9113	1825
9114	9115	198	887	1290	9116	9117	4612	507	9118	9119	344	3439	9120	9121	9122	9123	3152	9124
5729	24	9125	1553	9126	9127	2408	2431	9128	9129	3200	9130	2374	9131	5047	9132	3085	1371	9133
9134	3564	4831	2625	700	1386	1616	5310	5645	5488	9135	9136	9137	9138	9139	9140	4964	94	9141
3698	9142	9143	9144	9145	2759	1852	9146	9147	9148	9149	9150	5263	9151	9152	3032	9153	3215	9154
9155	5185	9156	281	9157	2882	510	465	9158	9159	9160	9161	2992	3133	30	9162	9163	9164	9165
9166	2096	9167	9168	1103	5070	2858	9169	672	1138	3553	9170	1026	9171	9172	9173	935	9174	453
3370	9175	954	2455	5024	9176	9177	5388	9178	9179	9180	3109	2164	2159	9181	573	9182	9183	1137
5579	1365	650	551	9184	612	1808	110	9185	9186	9187	1455	3104	9188	3537	1949	390	21	3523
9189	9190	9191	9192	9193	9194	9195	3723	9196	4924	9197	9198	9199	4910	9200	9201	5245	2932	9202
3359	9203	9204	9205	9206	9207	9208	9209	9210	9211	9212	9213	9214	1471	9215	6106	9216	5195	1596
389	9217	46	3281	69	9218	2239	2631	3893	1232	269	5952	9219	932	2657	5976	9220	9221	9222
9223	9224	6051	9225	9226	9227	9228	9229	3741	67	1978	9230	658	3900	9231	5187	3570	9232	1875
5641	1223	638	9233	9234	1579	9235	1116	9236	56	9237	1975	9238	9239	192	5582	9240	844	9241
5143	9242	3585	1477	1708	755	9243	3361	1654	9244	9245	9246	9247	1143	9248	9249	9250	44	9251
3048	9252	5157	5040	3928	376	4747	9253	9254	666	3052	1696	2490	1594	4713	9255	9256	1967	346
9257	3256	9258	1602	9259	9260	3334	761	426	9261	9262	1526	1068	9263	9264	9265	9266	9267	5775
9268	2732	1181	9269	2110	9270	3791	3275	1800	9271	9272	3309	4914	3696	9273	9274	9275	3527	9276
2056	9277	9278	9279	9280	9281	9282	1920	9283	9284	565	9285	3895	5260	5253	1655	1762	9286	9287
998	9288	9289	9290	9291	9292	9293	1399	9294	3182	9295	208	5440	9296	2612	9297	9298	9299	4959
9300	9301	9302	2979	3810	9303	5476	9304	1110	9305	9306	3050	1989	2527	1855	9307	3467	9308	9309

9310	9311	3590	3401	663	9312	9313	1671	9314	9315	9316	9317	9318	2765	9319	9320	9321	9322	4858	
9323	131	9324	9325	9326	9327	5808	1453	1078	9328	397	9329	9330	17	9331	645	3304	9332	9333	
307	348	9334	9335	9336	9337	9338	9339	9340	854	9341	4732	2132	9342	9343	3666	9344	9345	9346	
9347	9348	9349	9350	9351	2199	1946	9352	9353	9354	3408	1184	3686	5681	9355	1669	691	9356	5046	
260	600	9357	9358	9359	20	3418	9360	3377	3690	9361	9362	9363	9364	9365	9366	9367	2474	9368	
9369	5978	914	9370	9371	1383	1324	1114	9372	9373	9374	2740	2308	9375	2571	9376	9377	9378	2621	
9379	2570	9380	1191	2456	9381	9382	3364	9383	9384	2996	807	9385	9386	2451	9387	36	9388	5460	
9389	5413	9390	1706	9391	9392	9393	9394	9395	9396	1416	367	2714	9397	9398	9399	9400	462	1582	
9401	9402	9403	1267	3889	9404	9405	381	9406	9407	3680	9408	5172	2685	5910	4809	310	9409	247	
9410	9411	1688	9412	9413	9414	696	9415	9416	9417	9418	3731	9419	9420	3797	3007	9421	9422	9423	
9424	9425	2368	9426	9427	4652	9428	9429	9430	9431	6027	9432	4784	2807	3173	2253	2493	1795	9433	
183	4882	1824	9434	1157	3749	9435	9436	1102	9437	9438	1272	9439	9440	9441	9442	2381	1814	9443	
9444	9445	2787	9446	9447	1913	2388	9448	9449	9450	2806	3416	1865	9451	204	3631	3378	2640	1786	
9452	9453	5858	3411	9454	9455	9456	205	9457	9458	9459	9460	5741	2712	9461	9462	2651	1263	9463	
9464	1474	1085	9465	2061	579	9466	9467	9468	300	1882	1319	9469	9470	9471	545	9472	2188	9	
3366	9473	1506	9474	1681	749	4738	9475	9476	9477	9478	4575	9479	986	1380	2419	9480	1040	3516	
9481	9482	5651	2654	9483	3886	9484	9485	5162	1778	2695	9486	9487	9488	9489	221	3445	9490	9491	
3106	9492	2795	5499	9493	1799	1900	9494	9495	3558	2019	9496	2468	1096	3413	9497	3684	2556	5058	
9498	9499	9500	9501	1538	9502	1153	9503	3839	9504	9505	9506	9507	9508	9509	490	2976	3674	9510	
9511	9512	2	3477	1079	3096	1182	9513	9514	693	9515	2567	3931	9516	9517	9518	2473	9519	5342	
9520	9521	9522	6031	1807	2720	5170	9523	9524	1245	812	9525	9526	1226	9527	9528	9529	9530	3808	
9531	858	1933	9532	1668	1746	5373	9533	3492	4913	9534	2386	2659	1240	4844	2360	3175	9535	9536	
5843	3374	9537	698	5654	9538	1361	2516	9539	9540	1322	5797	2561	9541	9542	4615	3041	9543	1775	
9544	4772	9545	701	9546	9547	2958	9548	1922	9549	9550	9551	9552	3273	9553	2962	4769	9554	9555	
1768	9556	9557	9558	4926	9559	9560	9561	9562	9563	9564	1235	9565	9566	9567	9568	9569	9570	47	
2919	5985	2329	2242	9571	9572	9573	2869	9574	4654	1397	5657	5750	5873	9575	9576	9577	3016	9578	
3248	3339	9579	3474	1879	5005	1804	2841	5780	6022	9580	9581	5800	9582	9583	2939	9584	91	9585	
9586	9587	9588	9589	9590	9591	3030	9592	2777	9593	9594	2829	2818	9595	5841	9596	5056	9597	2363	
4739	9598	9599	9600	9601	9602	3675	6078	9603	9604	636	3296	9605	3038	9606	9607	309	9608	9609	
9610	1446	9611	2937	2241	9612	9613	9614	3449	9615	9616	9617	3338	3538	9618	9619	9620	2903	9621	
1683	9622	9623	9624	3057	9625	9626	9627	1495	2249	1925	9628	9629	634	5637	929	9630	9631	39	
753	9632	3639	9633	9634	9635	128	4887	3088	504	9636	3395	1292	5736	2081	1052	3740	9637	1552	
9638	9639	2075	9640	9641	1968	2686	9642	9643	9644	9645	3602	9646	9647	1806	9648	9649	9650	9651	
1581	3652	2837	9652	4635	9653	9654	9655	9656	845	3362	173	3531	9657	2502	2099	1006	9658	1128	
606	9659	5288	9660	9661	4730	9662	4888	4705	1302	4609	50	2604	135	9663	9664	9665	1209	9666	
3018	9667	909	3472	1164	9668	9669	9670	9671	1556	229	9672	6105	9673	2643	9674	3072	2846	4893	
2044	9675	9676	4724	1428	9677	9678	9679	9680	1520	9681	9682	9683	1097	9684	9685	9686	4689	9687	
1550	9688	3700	9689	1544	9690	9691	3536	6090	9692	676	2114	9693	1484	5016	9694	1250	1962	9695	
817	9696	771	2949	3054	9697	3784	9698	9699	9700	9701	153	9702	2260	5428	3216	3495	9703	2854	
9704	9705	1943	2330	9706	9707	5835	1167	9708	1902	9709	9710	9711	2449	9712	9713	3937	320	9714	
366	9715	1354	3710	9716	9717	2226	3688	660	9718	9719	1829	4801	9720	9721	9722	907	2299	9723	
3701	9724	4981	9725	9726	9727	2849	9728	5828	3294	9729	9730	3277	9731	1328	3920	3075	2121	1831	
9732	9733	9734	1081	9735	9736	3424	738	586	9737	9738	2233	9739	9740	9741	1955	5632	9742	9743	
420	9744	3669	9745	9746	5075	9747	9748	3388	141	9749	9750	9751	9752	9753	4688	4829	3259	2221	
1013	3720	9754	4843	1449	9755	1222	9756	9757	9758	3435	3285	2998	3946	611	9759	1588	9760	9761	
9762	9763	9764	9765	9766	9767	3853	9768	632	9769	9770	9771	5252	502	9772	3155	9773	9774	824	
429	2684	1942	9775	2483	9776	5173	9777	375	9778	9779	9780	9781	9782	9783	2750	199	1511	3148	
9784	530	9785	9786	9787	9788	9789	9790	9791	9792	3208	2688	158	9793	1419	9794	9795	2018	9796	
2406	906	9797	9798	5622	2704	9799	9800	9801	9802	9803	1155	1637	3267	2185	9804	870	9805	9806	
9807	9808	9809	9810	9811	9812	9813	751	1880	9814	9815	1283	9816	2617	9817	9818	9819	9820	1025	
9821	9822	9823	49	3385	9824	9825	9826	9827	1180	9828	9829	2816	3352	9830	9831	2401	9832	125	
1358	508	9833	9834	9835	112	9836	9837	9838	9839	9840	642	9841	9842	9843	9844	9845	816	3642	
9846	9847	9848	9849	813	5689	9850	9851	9852	9853	5188	9854	2850	9855	3718	2572	9856	1564	9857	
9858	9859	2433	9860	9861	80	6082	9862	9863	9864	3270	9865	1304	81	1384	303	2954	940	9866	
214	5012	4935	9867	1653	1130	9868	9869	9870	9871	3255	9872	9873	9874	9875	9876	9877	317	9878	
9879	9880	9881	2495	3606	866	9882	785	9883	856	3201	2463	5427	9884	9885	3940	9886	9887	1432	
797	220	9888	9889	9890	38	442	3672	3880	3630	9891	9892	2112	9893	5068	9894	9895	9896	2200	
2689	9897	3560	2355	9898	2951	9899	3542	9900	9901	9902	5500	9903	3037	1445	2770	1488	3031	6038	
1926	9904	414	2981	3161	9905	5535	1194	3899	860	5009	1392	226	713	9906	2735	1640	3664	5874	
5352	9907	469	9908	9909	9910	9911	3817	9912	5106	9913	1785	9914	9915	9916	943	9917	9918	9919	
2404	1475	2748	9920	9921	9922	9923	9924	4820	9925	2344	9926	9927	9928	2634	3518	9929	3421	9930	
9931	249	1566	9932	9933	9934	3632	9935	9936	9937	9938	9939	9940	9941	9942	9943	1093	9944	3209	
9945	850	9946	9947	2963	9948	1987	511	1112	2967	9949	593	1574	9950	2047	9951	9952	1266	9953	
9954	9955	9956	459	1426	9957	3039	3020	9958	5	9959	9960	9961	9962	9963	9964	2227	9965	9966	
9967	286	9968	9969	608	9970	9971	64	9972	9973	920	1372	9974	9975	1021	1133	9976	9977	9978	
9979	2143	6076	9980	9981	9982	3344	715	9983	9984	5601	4625	3336	9985	1151	9986	9987	9988	3406	
9989	1662	9990	2622	1531	9991	5491	9992	1041	9993	2895	2116	2125	9994	9995	9996	9997	3648	9998	
9999	3167	2071	1313	10000	10001	3266	10002	10003	3702	5747	10004	10005	1064	5451	10006	1429	4822	3581	
4702	2596	643	10007	10008	10009	10010	3205	10011	253	10012	10013	10014	6009	10015	2844	10016	10017	825	
2186	3871	3103	2292	5803	10018	10019	5701	10020	3003	10021	5968	10022	10023	10024	10025	4903	10026	699	
522	2453	10027	2825	10028	10029	1	364	2115	10030	10031	10032	10033	10034	1557	1307	10035	10036	10037	
10038	10039	10040	2127	10041	10042	673	10043	10044	10045	3134	3464	10046	10047	10048	10049	3689	10050	10051	
2578	10052	10053	10054	10055	10056	10057	5213	1402	1024	10058	10059	6008	1601	10060	5473	3854	10061	729	
10062	10063	223	1242	1257	1303	10064	10065	10066	3110	10067	10068	3846	10069	10070	10071	10072	157	3508	
2862	2191	10073	563	10074	2745	10075	1519	3202	3737	1820	1738	10076	25	2753	10077	10078	10079	10080	
161	2144	686	10081	10082	10083	10084	10085	5821	2840	10086	4781	1228	5553	10087	2983	10088	10089	3783	

10090 3884 10091 10092 10093 828 10094 2716 4963 79 3859 1382 10095 4725 1329 936 10096 2672 2361
10097 10098 3151 1816 10099 1231 10100 10101 3500 1271 10102 3909 831 10103 2965 1009 10104 10105 10106
10107 10108 525 2955 10109 10110 335 5900 10111 5682 10112 350 1868 10113 10114 2588 3149 10115 4684
10116 10117 4607 10118 3894 10119 10120 10121 10122 10123 10124 2317 70 10125 1225 2396 1518 2921 5468
10126 10127 934 10128 10129 419 2878 437 10130 10131 10132 10133 3440 10134 5108 2397 602 10135 10136
1848 206 891 1463 5783 10137 10138 1144 10139 10140 3454 1391 1856 4599 10141 10142 10143 361 10144
10145 10146 10147 10148 10149 342 144 10150 10151 5700 10152 885 2632 2619 10153 10154 3758 3019 2952
10155 10156 5548 2405 2774 10157 10158 3055 10159 10160 3130 878 10161 10162 2220 4947 296 1049 10163
10164 10165 3504 10166 10167 10168 2518 4761 10169 1323 10170 10171 10172 10173 10174 2943 1939 10175 2195
2182 886 10176 10177 26 10178 2007 5287 1127 1521 4992 140 10179 5319 1353 212 3682 10180 3241
10181 10182 10183 10184 10185 10186 10187 132 1513 2080 2931 10188 10189 2013 10190 1728 10191 10192 10193
10194 2866 10195 10196 10197 10198 10199 10200 3838 10201 1614 10202 10203 369 10204 5805 10205 1293 1779
175 1527 2802 10206 2541 3738 1793 5721 10207 3124 2141 10208 3611 911 10209 710 10210 10211 1314
10212 849 3513 10213 3748 10214 1901 10215 2348 10216 10217 3105 2637 2974 10218 10219 10220 3101 10221
10222 10223 3131 3534 10224 10225 10226 2980 10227 10228 10229 10230 1849 5792 10231 5887 3496 1443 10232
992 10233 10234 4787 10235 5635 10236 2314 314 5958 10237 10238 2810 2901 10239 1369 2761 10240 10241
2696 5227 2855 486 10242 1481 10243 3828 2122 2340 1896 10244 3243 2574 3805 1721 2595 2747 2302
1675 10245 10246 10247 10248 5257 5125 3236 10249 5120 10250 3831 5420 10251 5424 10252 3250 996 10253
10254 883 10255 5435 1161 10256 188 10257 10258 10259 10260 10261 3583 1430 55 10262 3316 1931 10263
10264 10265 10266 10267 373 997 2167 283 10268 10269 10270 2311 2559 10271 10272 10273 10274 10275 10276
10277 10278 2925 10279 1510 10280 10281 2549 386 10282 10283 1175 10284 10285 10286 10287 3837 2218 10288
3913 2022 10289 10290 10291 10292 1101 10293 1792 10294 2057 325 1436 2017 664 10295 3091 10296 2771
10297 1092 10298 517 10299 3221 1298 10300 10301 10302 5649 3735 10303 1058 10304 10305 345 10306 1070
10307 581 1988 4923 2133 1465 2063 3154 10308 5344 441 10309 2509 10310 10311 10312 4835 10313 10314
5773 10315 10316 2874 10317 10318 2156 1507 5964 1844 10319 3127 10320 10321 3121 10322 10323 10324 5994
10325 10326 166 10327 291 10328 1038 265 3169 5026 10329 10330 10331 10332 10333 3307 1751 10334 3636
10335 10336 248 10337 10338 3230 10339 10340 10341 10342 1355 1337 10343 838 32 3086 2999 10344 10345
881 10346 10347 1237 1422 10348 10349 2828 10350 1571 823 10351 10352 2734 10353 10354 10355 1726 10356
10357 10358 10359 10360 10361 10362 10363 5784 3021 3539 10364 4641 10365 10366 2648 10367 1597 2026 5027
10368 2040 10369 10370 10371 10372 2257 10373 10374 2677 10375 809 3772 5206 10376 1593 10377 10378 10379
10380 2331 10381 10382 10383 10384 3522 3566 3874 2126 184 10385 10386 347 10387 10388 10389 3204 5177
10390 13 10391 10392 3244 1547 215 10393 10394 10395 4640 82 10396 10397 10 1514 10398 10399 10400
10401 1497 10402 10403 3873 10404 1919 10405 988 97 3002 10406 10407 10408 10409 2512 10410 10411 10412
10413 10414 10415 764 10416 1406 5097 10417 10418 2469 1512 2378 2764 10419 5846 10420 2461 65 10421
5908 10422 10423 3190 597 10424 3796 3436 5624 10425 10426 10427 1417 2209 10428 10429 4861 3663 3676
245 10430 10431 10432 3011 10433 10434 10435 10436 10437 10438 10439 10440 10441 2773 2820 10442 10443 10444
5525 1012 10445 10446 10447 10448 6011 10449 10450 10451 10452 10453 2772 10454 10455 10456 3394 10457 10458
10459 1152 2940 10460 1082 10461 3254 10462 2376 10463 3367 10464 5955 10465 5545 10466 10467 3498 10468
2755 10469 10470 2347 10471 10472 1379 2767 203 10473 4719 1227 10474 10475 3014 1016 10476 10477 10478
311 1972 3234 953 3115 10479 2194 3108 10480 200 731 10481 10482 10483 10484 599 2997 10485 10486
10487 10488 10489 10490 2642 2170 10491 10492 1732 194 10493 1002 3847 3004 2078 10494 3944 10495 10496
10497 832 267 10498 142 10499 5829 10500 1150 10501 10502 10503 3430 10504 10505 1195 10506 1056 1624
382 10507 5400 10508 2441 10509 1803 456 10510 3095 3303 272 5683 5847 3655 3775 1898 1720 10511
3891 10512 10513 4949 1604 10514 10515 10516 10517 10518 10519 10520 2282 821 10521 10522 138 10523 1723
3145 10524 10525 10526 10527 3555 2692 10528 1174 1349 1639 89 10529 5127 10530 546 3251 10531 2289
10532 10533 591 10534 10535 10536 2157 10537 10538 10539 1340 10540 1253 10541 3363 1863 5028 10542 10543
10544 3670 10545 2823 10546 10547 10548 10549 1790 10550 494 1854 2945 5209 4814 2990 10551 869 10552
10553 10554 10555 10556 10557 2879 10558 10559 3753 3023 10560 331 2922 10561 3350 10562 1341 10563 10564
10565 1739 10566 10567 862 10568 590 10569 10570 1317 288 2295 10571 2412 10572 10573 3490 114 10574
2235 5556 3860 10575 10576 10577 888 5509 360 10578 10579 3089 10580 10581 5990 10582 10583 3262 3199
1204 3059 10584 2501 10585 10586 10587 10588 10589 5661 3568 10590 10591 10592 2987 6059 3660 10593 10594
10595 10596 10597 10598 10599 1747 3314 10600 10601 5060 478 10602 1462 1388 10603 10604 1940 4928 10605
10606 814 1370 10607 10608 2713 10609 10610 10611 10612 10613 3865 10614 10615 10616 1877 1159 10617 10618
10619 952 5718 10620 10621 10622 10623 10624 10625 10626 1959 10627 822 10628 5892 10629 10630 10631 1573
10632 10633 1670 3380 4968 628 1439 4953 10634 3612 10635 1031 585 10636 2989 5216 493 10637 10638
2890 2731 10639 10640 4657 10641 10642 2151 10643 4998 1046 10644 10645 10646 4879 3533 784 10647 10648
2270 10649 10650 5907 4644 10651 4610 3245 10652 1652 5100 10653 10654 2975 10655 10656 3719 3752 10657
10658 10659 2352 10660 10661 10662 5363 230 2109 2627 10663 10664 10665 723 2357 2372 10666 10667 10668
1632 2202 10669 10670 10671 1071 10672 10673 5445 10674 6107 2618 10675 2379 10676 3730 10677 1705 800
2041 2906 10678 10679 2248 3722 10680 10681 10682 10683 10684 2356 10685 10686 2424 2697 5868 3582 10687
5516 10688 10689 10690 10691 10692 2021 1840 301 2752 10693 10694 10695 10696 10697 1077 10698 1730 10699
3777 10700 10701 3673 10702 942 10703 10704 4896 10705 10706 1853 661 2511 3843 10707 5324 10708 5957
5308 1362 10709 10710 2224 4794 10711 1857 10712 10713 3181 2165 10714 10715 10716 3799 10717 3360 3441
10718 5300 10719 10720 1094 1912 1745 10721 10722 2103 1089 778 10723 3677 485 10724 5487 5159 10725
1577 90 10726 10727 3325 2606 10728 5774 6015 644 1469 10729 2542 10730 865 10731 1697 10732 374
10733 1881 10734 393 3479 10735 4588 10736 3628 10737 1215 10738 10739 10740 2538 10741 10742 10743 10744
10745 277 2699 10746 10747 10748 399 1197 2791 10749 10750 5647 908 1664 10751 10752 10753 1698 10754
10755 2266 3457 2786 14 1646 1470 1357 10756 1549 3231 10757 10758 1862 10759 5446 1936 10760 10761
190 10762 3938 3456 10763 651 505 10764 10765 10766 1628 10767 10768 10769 3279 633 235 3183 371
5113 10770 3637 1750 10771 1393 232 559 3901 180 10772 10773 150 10774 10775 3552 10776 679 10777
10778 10779 10780 10781 3667 10782 10783 725 3746 10784 1612 1331 680 2768 10785 5969 2923 2150 1712
1320 10786 728 688 1878 529 607 4944 10787 5098 10788 10789 10790 3786 185 11 1277 10791 2325
2698 10792 10793 10794 10795 4649 1450 1938 10796 10797 5915 3323 1759 994 10798 3288 10799 10800 10801
3114 10802 10803 561 10804 10805 2881 2210 10806 2727 3724 2334 10807 10808 10809 10810 10811 3249 10812
1247 2287 10813 10814 10815 2273 5515 913 861 10816 5610 2600 10817 10818 10819 408 2135 129 10820
5940 772 10821 5182 1437 1921 10822 3289 10823 1667 10824 10825 10826 10827 3917 10828 10829 10830 3405

10831 10832 10833 10834 10835 3609 4638 10836 10837 5508 10838 10839 3832 10840 3658 10841 10842 10843 28
584 10844 10845 3404 10846 4778 10847 750 3603 4819 499 3024 10848 1118 5717 2050 5979 1489 2603
10849 10850 3478 774 3157 10851 10852 10853 284 10854 10855 10856 10857 351 10858 2703 1951 1866 4581
10859 890 10860 10861 1249 10862 10863 10864 10865 10866 10867 5483 10868 10869 10870 10871 10872 10873 4883
10874 10875 10876 10877 10878 4668 10879 10880 4731 10881 10882 10883 10884 10885 10886 10887 10888 10889 10890
4614 10891 10892 10893 10894 10895 10896 10897 5698 10898 10899 10900 10901 10902 10903 10904 10905 10906 10907
10908 10909 10910 10911 10912 10913 10914 10915 4619 4758 5652 10916 10917 5660 5290 10918 10919 10920 10921
5434 10922 10923 10924 10925 10926 10927 10928 10929 10930 5672 10931 10932 10933 10934 10935 10936 10937 10938
10939 10940 10941 10942 10943 10944 10945 10946 10947 10948 5007 4911 10949 10950 10951 10952 4630 10953 10954
10955 10956 10957 10958 10959 10960 10961 10962 10963 10964 10965 10966 10967 10968 10969 5457 10970 10971 10972
10973 10974 5334 10975 10976 10977 10978 10979 10980 10981 10982 5756 10983 10984 10985 10986 6083 10987 10988
5916 10989 10990 10991 10992 4909 10993 10994 10995 10996 10997 10998 10999 11000 11001 11002 11003 11004 11005
11006 5289 11007 11008 11009 11010 11011 5429 11012 11013 11014 11015 11016 11017 5901 11018 11019 11020 11021
11022 11023 11024 11025 11026 11027 5594 11028 11029 11030 11031 11032 11033 11034 11035 11036 11037 11038 11039
11040 4962 11041 11042 11043 11044 11045 4768 11046 5295 11047 11048 5284 11049 4704 11050 11051 11052 5140
11053 11054 5377 11055 6050 11056 11057 11058 11059 11060 11061 11062 11063 11064 11065 6005 11066 4646 11067
11068 11069 4613 11070 11071 11072 11073 11074 11075 11076 11077 11078 5671 11079 4695 11080 11081 11082 11083
11084 11085 11086 11087 11088 11089 11090 11091 11092 11093 11094 11095 11096 11097 11098 11099 11100 11101 11102
11103 11104 5343 11105 11106 11107 11108 11109 11110 11111 11112 6063 11113 11114 4716 4830 11115 11116 11117
11118 11119 11120 11121 11122 6041 11123 5962 11124 11125 11126 11127 11128 11129 11130 11131 11132 11133 11134
11135 11136 11137 11138 11139 11140 11141 11142 11143 4927 11144 5181 11145 11146 11147 11148 11149 11150 5966
4578 11151 11152 11153 11154 11155 5857 11156 11157 11158 11159 11160 11161 11162 11163 11164 11165 11166 11167
11168 11169 5927 11170 11171 11172 11173 5581 11174 11175 11176 11177 4740 11178 11179 11180 11181 11182 11183
11184 11185 11186 11187 11188 11189 11190 11191 11192 11193 4675 11194 11195 5082 11196 11197 5880 11198 11199
11200 11201 5540 11202 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11213 11214 11215 11216 11217
11218 11219 11220 11221 11222 11223 11224 11225 11226 11227 11228 11229 11230 11231 11232 11233 11234 11235 11236
11237 11238 11239 11240 11241 11242 11243 11244 11245 5971 11246 11247 11248 11249 11250 11251 11252 11253 11254
11255 11256 11257 11258 11259 11260 11261 11262 11263 11264 5894 11265 11266 11267 11268 11269 11270 11271 11272
11273 11274 4624 11275 11276 11277 11278 11279 4984 11280 11281 11282 11283 11284 11285 11286 11287 11288 11289
5676 11290 11291 11292 11293 11294 5628 11295 11296 11297 11298 11299 5018 4703 11300 11301 11302 11303 11304
11305 11306 11307 11308 11309 11310 11311 6102 11312 11313 5492 11314 11315 11316 11317 11318 5771 11319 11320
11321 11322 11323 11324 11325 11326 11327 11328 11329 11330 5275 4655 11331 11332 11333 11334 5136 5980 11335
11336 11337 11338 11339 11340 11341 11342 4734 11343 11344 11345 11346 11347 11348 5465 4860 11349 11350 11351
5199 11352 11353 11354 11355 11356 11357 5564 11358 11359 5731 11360 11361 5355 5762 11362 11363 11364 11365
11366 5183 11367 11368 11369 11370 11371 11372 11373 11374 11375 11376 11377 11378 4939 11379 11380 11381 11382
5357 11383 6032 5367 11384 11385 11386 11387 11388 11389 5588 11390 11391 11392 11393 11394 11395 11396 11397
11398 11399 11400 11401 11402 11403 11404 11405 11406 11407 11408 11409 11410 11411 11412 11413 11414 11415 11416
11417 11418 11419 11420 11421 11422 11423 5230 11424 11425 5495 11426 11427 11428 11429 11430 11431 11432 11433
11434 11435 11436 11437 11438 11439 4726 11440 11441 5003 4785 5240 11442 11443 11444 11445 11446 11447 11448
11449 11450 11451 5655 5493 11452 11453 4708 4580 11454 11455 11456 11457 11458 11459 11460 11461 5048 11462
11463 11464 11465 11466 11467 11468 11469 6061 11470 5250 11471 11472 11473 11474 11475 5149 11476 11477 11478
11479 11480 11481 11482 11483 11484 11485 11486 11487 11488 11489 11490 11491 11492 11493 11494 11495 11496 5608
11497 11498 11499 5882 5071 4986 11500 11501 11502 11503 11504 5043 11505 11506 5562 11507 11508 11509 11510
4786 11511 11512 11513 11514 11515 11516 11517 11518 11519 11520 11521 11522 11523 11524 5678 11525 11526 11527
11528 11529 11530 11531 11532 11533 11534 5702 11535 11536 11537 4938 11538 11539 11540 11541 11542 11543 5722
11544 11545 11546 11547 11548 11549 11550 11551 11552 11553 11554 11555 11556 11557 11558 11559 11560 11561 11562
11563 11564 11565 11566 11567 11568 11569 11570 11571 11572 11573 11574 11575 11576 11577 11578 11579 11580 11581
11582 11583 5237 11584 11585 11586 11587 11588 11589 11590 11591 11592 11593 11594 11595 5148 11596 11597 11598
11599 11600 11601 11602 5685 11603 4666 11604 11605 11606 11607 11608 11609 11610 11611 11612 11613 11614 11615
11616 11617 11618 11619 11620 11621 11622 11623 11624 11625 11626 11627 11628 11629 11630 11631 11632 5086 5621
11633 11634 11635 11636 11637 11638 11639 11640 5517 11641 11642 11643 11644 11645 5200 11646 11647 11648 11649
11650 11651 11652 11653 11654 11655 5744 11656 4602 11657 11658 11659 11660 11661 11662 11663 11664 5035 11665
11666 11667 11668 4586 5726 11669 4637 4603 11670 11671 11672 11673 11674 11675 11676 11677 11678 11679 11680
11681 11682 11683 11684 11685 11686 11687 11688 11689 11690 11691 4991 5674 11692 11693 11694 11695 5552 11696
5147 11697 11698 11699 11700 11701 5380 11702 11703 11704 11705 11706 11707 11708 11709 5811 11710 11711 11712
11713 11714 11715 11716 11717 11718 11719 11720 11721 11722 11723 11724 5612 11725 11726 5623 11727 5283 11728
6052 11729 11730 4869 11731 11732 11733 11734 11735 11736 11737 11738 11739 11740 11741 11742 5103 11743 11744
11745 11746 11747 11748 11749 11750 4867 11751 11752 11753 11754 11755 11756 11757 11758 11759 11760 11761 11762
11763 11764 11765 11766 11767 5834 11768 4902 11769 11770 11771 4662 4974 4693 11772 11773 5998 11774 11775
11776 11777 5276 11778 11779 11780 4779 11781 11782 11783 11784 11785 4954 5309 11786 11787 5282 11788 11789
11790 11791 11792 6099 11793 6060 5091 11794 5474 11795 11796 11797 11798 5325 11799 11800 11801 11802 11803
11804 11805 11806 11807 4611 11808 11809 5925 5212 11810 5706 11811 11812 11813 11814 11815 5806 11816 5724
11817 11818 11819 11820 11821 11822 11823 11824 11825 11826 11827 11828 11829 5557 11830 11831 11832 4626 11833
5101 11834 11835 11836 11837 11838 5266 11839 11840 11841 11842 11843 11844 11845 11846 11847 11848 11849 11850
11851 11852 11853 11854 11855 11856 11857 11858 11859 11860 11861 11862 11863 11864 11865 11866 11867 5235 11868
11869 11870 11871 11872 11873 11874 11875 5709 11876 11877 11878 4885 11879 11880 11881 11882 11883 11884 11885
11886 11887 11888 11889 11890 11891 11892 11893 11894 11895 11896 11897 11898 11899 11900 11901 11902 11903 11904
11905 5062 11906 11907 11908 11909 11910 11911 11912 11913 11914 11915 5328 11916 11917 5053 11918 11919 11920
11921 5470 11922 11923 11924 11925 11926 11927 11928 11929 11930 11931 11932 11933 5286 11934 11935 11936 11937
11938 11939 11940 11941 11942 11943 11944 11945 11946 11947 11948 11949 11950 11951 11952 11953 11954 4584 11955
11956 11957 11958 11959 11960 11961 11962 11963 11964 11965 11966 11967 11968 11969 5432 11970 11971 11972 11973
11974 5067 11975 11976 11977 11978 11979 11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
11992 11993 11994 5567 5830 11995 11996 11997 11998 11999 12000 12001 12002 12003 12004 12005 12006 12007 12008
12009 12010 12011 12012 12013 12014 12015 12016 12017 4999 12018 12019 12020 12021 5302 12022 12023 12024 12025
12026 12027 12028 12029 12030 12031 12032 12033 5255 12034 5462 12035 12036 12037 12038 12039 12040 12041 5114
12042 12043 12044 12045 12046 12047 12048 12049 12050 6010 12051 12052 12053 12054 12055 12056 12057 12058 12059

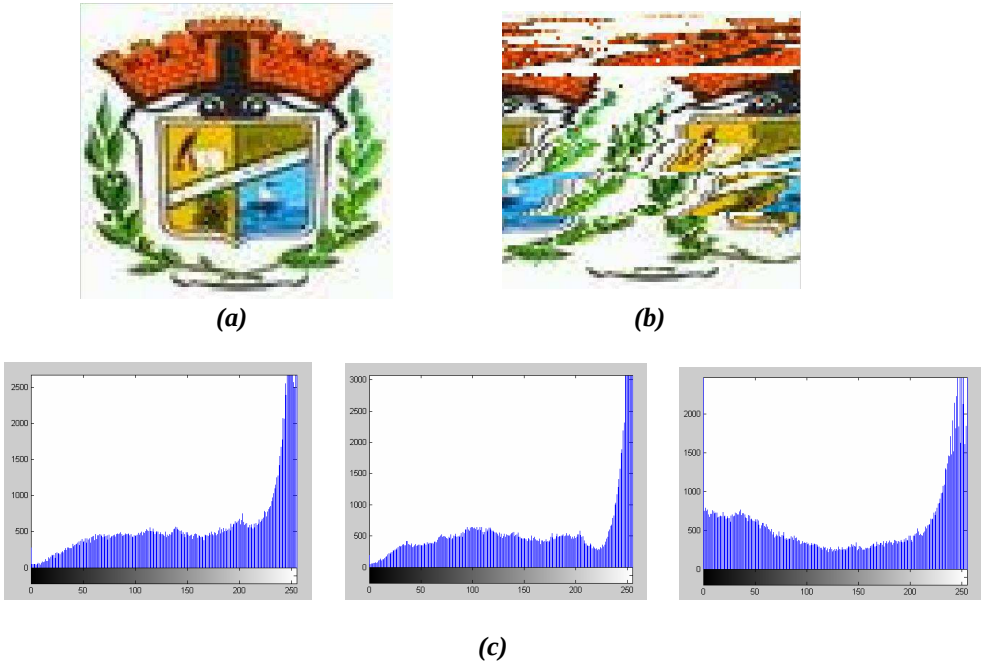
12060 12061 12062 12063 12064 5150 5299 12065 12066 12067 12068 12069 12070 12071 12072 12073 12074 12075 12076
5864 12077 12078 12079 5555 12080 12081 12082 5191 5912 12083 12084 12085 12086 12087 12088 12089 12090 12091
12092 12093 12094 12095 12096 4881 12097 12098 12099 12100 12101 12102 12103 12104 12105 12106 12107 12108 12109
4825 12110 12111 12112 12113 12114 12115 5455 12116 12117 5590 12118 12119 12120 5840 12121 12122 5145 12123
12124 12125 12126 12127 12128 12129 12130 12131 12132 12133 12134 12135 12136 12137 12138 4823 4764 12139 5087
12140 12141 12142 12143 12144 12145 12146 12147 12148 12149 12150 4686 12151 12152 12153 12154 4899 12155 12156
12157 12158 12159 12160 12161 12162 12163 12164 12165 12166 12167 12168 6025 4948 12169 12170 5577 12171 5395
6100 12172 12173 12174 12175 12176 12177 12178 12179 12180 12181 12182 12183 12184 12185 12186 12187 12188 12189
12190 12191 12192 12193 12194 12195 12196 12197 12198 12199 12200 12201 12202 12203 12204 12205 12206 12207 12208
12209 12210 12211 5327 12212 12213 12214 5630 6026 4925 12215 12216 12217 12218 12219 12220 12221 12222 12223
12224 12225 12226 12227 12228 12229 12230 12231 12232 5095 12233 5360 5845 12234 12235 6092 5135 12236 12237
12238 12239 12240 12241 12242 12243 12244 12245 12246 12247 12248 12249 12250 12251 12252 5898 12253 12254 12255
12256 12257 12258 12259 12260 12261 12262 12263 5163 12264 12265 12266 12267 12268 12269 12270 12271 12272 12273
12274 12275 12276 12277 5954 12278 12279 12280 12281 12282 12283 12284 4593 12285 12286 12287 12288 12289 12290
12291 12292 12293 12294 12295 4979 5526 5039 12296 12297 12298 12299 12300 12301 12302 12303 4674 4585 12304
12305 12306 12307 12308 5862 12309 5041 12310 12311 12312 12313 12314 12315 12316 12317 12318 12319 12320 12321
12322 12323 12324 12325 12326 12327 12328 5667 12329 5176 12330 5869 12331 12332 12333 12334 12335 5378 12336
12337 12338 12339 12340 12341 12342 5022 12343 5758 12344 12345 5123 12346 12347 5034 12348 12349 5663 12350
12351 12352 5096 5345 12353 12354 12355 12356 12357 12358 12359 5339 12360 12361 5403 5000 12362 12363 12364
12365 12366 12367 12368 12369 12370 12371 12372 12373 12374 12375 12376 12377 12378 12379 12380 12381 12382 12383
12384 12385 12386 12387 12388 12389 12390 12391 12392 12393 12394 12395 12396 12397 12398 5679 12399 12400 12401
12402 4905 12403 12404 12405 12406 12407 12408 4639 4889 12409 12410 12411 12412 12413 12414 5410 12415 12416
12417 12418 12419 12420 12421 4579 12422 12423 12424 12425 12426 12427 12428 12429 12430 12431 12432 5589 12433
12434 12435 12436 12437 12438 5616 12439 12440 12441 12442 12443 12444 12445 12446 12447 12448 12449 12450 12451
12452 12453 12454 12455 4891 12456 12457 12458 12459 12460 12461 12462 12463 12464 12465 12466 12467 5524 12468
5987 12469 12470 12471 12472 12473 12474 12475 12476 12477 12478 12479 12480 12481 12482 5316 12483 12484 12485
12486 4658 12487 12488 12489 4746 12490 12491 4745 12492 12493 12494 12495 12496 12497 12498 12499 12500 12501
12502 5002 12503 12504 12505 12506 12507 12508 12509 12510 5037 12511 12512 12513 12514 12515 12516 12517 12518
12519 4818 4791 12520 12521 12522 12523 12524 5761 12525 12526 12527 12528 12529 12530 12531 12532 12533 12534
12535 12536 5602 5072 12537 12538 5055 12539 12540 12541 12542 12543 12544 12545 12546 12547 5677 12548 12549
12550 12551 12552 12553 12554 12555 12556 12557 5178 4983 12558 12559 5093 12560 12561 12562 12563 5501 12564
12565 12566 12567 12568 12569 12570 12571 12572 12573 12574 6075 12575 12576 12577 12578 12579 12580 12581 12582
12583 12584 12585 12586 12587 12588 12589 12590 5341 12591 12592 12593 12594 12595 6036 5859 12596 12597 12598
5636 12599 12600 12601 4692 12602 12603 12604 12605 12606 12607 12608 12609 5728 12610 12611 12612 12613 12614
12615 5520 12616 12617 12618 12619 12620 12621 12622 5475 12623 12624 12625 12626 12627 12628 12629 12630 12631
12632 12633 12634 12635 5743 12636 12637 5390 12638 12639 12640 12641 12642 12643 12644 12645 12646 4741 12647
12648 12649 12650 12651 12652 12653 12654 12655 12656 12657 12658 12659 12660 12661 12662 12663 12664 4851 12665
12666 12667 12668 12669 12670 12671 5768 12672 12673 12674 12675 12676 12677 12678 12679 12680 12681 12682 12683
12684 12685 12686 12687 5640 12688 5751 12689 5575 12690 12691 12692 12693 12694 12695 12696 12697 12698 12699
12700 12701 12702 12703 12704 12705 12706 12707 12708 12709 12710 12711 5471 12712 12713 4752 5600 12714 5045
5831 12715 12716 12717 12718 12719 12720 12721 12722 5504 12723 12724 12725 12726 12727 12728 12729 12730 12731
12732 12733 5742 12734 12735 12736 12737 12738 5745 12739 12740 12741 12742 12743 12744 12745 12746 12747 12748
5085 12749 12750 12751 5298 12752 12753 12754 12755 5658 12756 12757 12758 5697 12759 5394 4798 12760 12761
12762 12763 12764 12765 12766 12767 12768 12769 12770 12771 5983 12772 12773 12774 12775 12776 12777 12778 12779
12780 12781 12782 12783 4717 12784 12785 12786 5346 5735 12787 12788 12789 12790 12791 12792 12793 12794 12795
12796 12797 12798 12799 5375 12800 12801 12802 12803 12804 5974 6055 12805 12806 12807 12808 4782 12809 12810
12811 12812 12813 12814 12815 12816 5809 12817 12818 12819 12820 12821 12822 12823 12824 4727 12825 12826 12827
12828 12829 12830 12831 12832 12833 12834 12835 12836 12837 12838 12839 12840 12841 12842 5586 12843 12844 12845
12846 12847 12848 5450 12849 12850 12851 12852 5496 12853 12854 12855 12856 12857 12858 12859 12860 12861 12862
12863 12864 12865 12866 12867 12868 12869 12870 12871 12872 12873 12874 12875 12876 12877 5443 12878 12879 12880
12881 12882 12883 5128 12884 12885 12886 12887 12888 12889 12890 5236 5076 12891 12892 12893 12894 12895 12896
12897 12898 12899 12900 12901 12902 12903 4595 12904 12905 12906 12907 12908 12909 12910 12911 12912 12913 12914
12915 12916 4849 12917 12918 12919 12920 12921 12922 12923 12924 5478 12925 12926 12927 12928 12929 12930 12931
12932 12933 12934 12935 12936 12937 12938 12939 12940 4956 12941 12942 12943 4900 12944 12945 12946 12947 12948
12949 6070 12950 12951 12952 12953 12954 12955 12956 12957 12958 5823 12959 12960 12961 12962 12963 12964 12965
12966 12967 12968 5249 12969 12970 12971 12972 12973 12974 12975 5786 12976 12977 12978 12979 12980 12981 12982
12983 12984 12985 12986 12987 12988 12989 12990 5023 12991 5819 12992 12993 6004 12994 12995 12996 5951 12997
12998 12999 13000 13001 13002 13003 13004 13005 5318 13006 13007 5961 13008 13009 13010 13011 13012 13013 13014
13015 13016 13017 13018 13019 13020 13021 13022 13023 5311 4628 13024 13025 13026 13027 13028 13029 13030 5913
4865 13031 13032 13033 13034 13035 5008 13036 5644 13037 4985 13038 13039 13040 6028 5479 13041 13042 13043
13044 13045 13046 13047 13048 13049 13050 13051 13052 13053 13054 13055 13056 13057 13058 13059 13060 13061 13062
13063 13064 13065 13066 13067 13068 13069 13070 13071 13072 13073 13074 13075 13076 13077 13078 13079 13080 5194
13081 13082 13083 13084 13085 13086 5444 13087 13088 13089 5226 13090 13091 13092 13093 13094 13095 13096 13097
13098 13099 4651 13100 13101 13102 13103 13104 5141 4982 13105 13106 13107 13108 13109 6012 13110 13111 13112
13113 13114 13115 13116 13117 13118 13119 13120 13121 13122 13123 13124 13125 13126 5374 13127 13128 13129 13130
13131 13132 13133 4989 13134 13135 13136 13137 13138 5825 13139 13140 13141 13142 13143 13144 13145 5712 13146
5625 13147 13148 13149 13150 13151 13152 13153 13154 13155 5812 13156 13157 13158 13159 13160 13161 5247 13162
13163 13164 13165 13166 13167 13168 13169 13170 13171 13172 13173 13174 13175 13176 13177 4777 13178 13179 13180
13181 13182 13183 5215 5937 13184 13185 13186 13187 13188 13189 13190 13191 13192 6108 13193 13194 4707 13195
13196 13197 13198 13199 5943 13200 13201 13202 13203 4894 13204 13205 13206 13207 13208 13209 13210 13211 13212
13213 13214 13215 13216 13217 13218 13219 13220 13221 13222 13223 13224 13225 13226 13227 13228 13229 4729 13230
5899 13231 13232 13233 13234 13235 13236 13237 13238 13239 13240 5218 13241 13242 13243 13244 13245 13246 13247
5453 13248 13249 13250 13251 13252 4813 13253 13254 13255 13256 13257 13258 13259 13260 13261 13262 13263 13264
13265 13266 13267 13268 13269 4837 13270 13271 13272 13273 13274 13275 13276 13277 13278 13279 13280 13281 5036
13282 13283 13284 13285 13286 13287 13288 13289 13290 5441 13291 13292 13293 13294 4590 13295 13296 13297 13298
13299 13300 13301 13302 5169 13303 13304 13305 13306 13307 13308 13309 13310 13311 13312 13313 13314 13315 13316

13317 13318 13319 13320 13321 13322 13323 13324 13325 13326 13327 13328 13329 13330 13331 13332 5402 13333 13334
13335 13336 13337 13338 13339 13340 13341 13342 5347 13343 13344 13345 13346 13347 13348 13349 13350 13351 13352
13353 13354 13355 5922 13356 13357 13358 13359 13360 4783 13361 5369 13362 13363 13364 13365 13366 13367 13368
13369 13370 13371 13372 13373 13374 5323 13375 13376 13377 13378 13379 13380 13381 5592 13382 13383 13384 13385
13386 13387 5563 13388 13389 13390 13391 13392 13393 13394 13395 13396 13397 13398 13399 13400 13401 13402 13403
13404 13405 13406 13407 5268 13408 6035 13409 5111 13410 13411 4653 13412 4906 13413 13414 13415 13416 13417
4636 4797 13418 13419 13420 13421 13422 13423 13424 13425 13426 13427 13428 13429 13430 13431 13432 13433 13434
13435 13436 13437 13438 13439 13440 13441 13442 13443 13444 13445 13446 13447 13448 13449 13450 5820 13451 13452
13453 13454 13455 13456 13457 13458 13459 13460 13461 13462 13463 13464 13465 13466 13467 13468 13469 13470 13471
13472 13473 13474 13475 13476 13477 5558 13478 13479 5764 13480 13481 13482 13483 13484 13485 13486 13487 4681
13488 13489 13490 13491 13492 13493 13494 13495 13496 5854 13497 13498 13499 13500 4884 13501 4800 13502 13503
13504 13505 13506 13507 13508 13509 13510 13511 13512 13513 13514 13515 13516 13517 13518 13519 13520 13521 13522
13523 13524 13525 13526 13527 13528 5886 13529 5670 13530 13531 13532 13533 5791 13534 13535 13536 13537 13538
13539 13540 13541 13542 13543 13544 13545 13546 13547 13548 13549 13550 13551 13552 13553 13554 13555 13556 13557
13558 13559 13560 13561 5883 13562 13563 13564 13565 13566 13567 13568 13569 13570 13571 13572 13573 13574 13575
13576 13577 13578 13579 13580 5001 13581 13582 13583 13584 4821 13585 13586 13587 13588 13589 13590 13591 13592
13593 13594 13595 13596 13597 13598 13599 13600 13601 13602 13603 13604 13605 13606 5425 13607 13608 13609 13610
13611 13612 13613 13614 13615 13616 13617 13618 13619 13620 13621 6034 13622 13623 13624 13625 13626 13627 13628
13629 13630 13631 4765 4642 13632 4827 13633 13634 6039 4836 13635 13636 5117 13637 13638 5837 13639 13640
13641 13642 13643 13644 13645 13646 13647 13648 13649 13650 13651 13652 13653 13654 13655 13656 13657 5551 13658
13659 13660 13661 5267 13662 13663 13664 5904 13665 13666 13667 13668 13669 13670 13671 13672 13673 13674 13675
13676 13677 13678 13679 5385 13680 13681 13682 5595 13683 13684 13685 13686 13687 13688 13689 13690 13691 13692
13693 13694 13695 5920 13696 13697 13698 13699 13700 13701 13702 13703 13704 13705 13706 13707 13708 4605 13709
13710 13711 13712 13713 13714 13715 13716 13717 5963 13718 13719 13720 13721 13722 5387 13723 13724 13725 13726
13727 13728 13729 13730 13731 13732 13733 5903 13734 13735 13736 13737 13738 5817 13739 5408 5269 13740 13741
13742 13743 13744 13745 13746 13747 13748 13749 13750 13751 13752 13753 13754 13755 13756 13757 13758 13759 13760
13761 13762 5646 13763 13764 13765 13766 13767 13768 13769 13770 13771 13772 13773 13774 13775 13776 13777 13778
13779 5597 13780 13781 13782 13783 4673 5956 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 4696
13794 13795 13796 5497 13797 13798 4712 5799 13799 13800 13801 13802 13803 13804 13805 13806 13807 13808 13809
13810 13811 13812 13813 13814 13815 13816 13817 13818 13819 13820 13821 13822 13823 13824 13825 13826 13827 13828
13829 5442 4701 13830 4859 13831 6064 13832 13833 13834 13835 13836 13837 13838 13839 13840 13841 13842 13843
13844 13845 13846 13847 13848 13849 13850 13851 13852 13853 13854 13855 13856 13857 5186 13858 13859 13860 13861
13862 13863 13864 5431 13865 13866 13867 13868 5449 13869 4995 13870 13871 13872 13873 13874 13875 13876 5604
13877 13878 13879 5406 13880 13881 13882 13883 13884 13885 13886 13887 13888 13889 13890 13891 4863 13892 13893
5695 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903 13904 13905 5815 13906 13907 13908 13909 13910
13911 13912 13913 13914 5571 13915 13916 13917 5208 13918 13919 13920 5109 13921 13922 13923 13924 13925 13926
13927 13928 13929 13930 13931 13932 13933 13934 13935 13936 13937 13938 13939 13940 5224 13941 13942 13943 13944
13945 13946 13947 5407 13948 13949 13950 13951 5189 4620 13952 4582 13953 13954 13955 13956 13957 13958 13959
13960 13961 13962 13963 13964 13965 13966 13967 13968 13969 13970 13971 5536 13972 13973 13974 13975 13976 13977
13978 13979 13980 13981 13982 5533 13983 13984 13985 5042 13986 13987 13988 13989 13990 13991 5776 13992 13993
13994 13995 4803 13996 13997 5167 13998 4743 13999 14000 14001 14002 14003 14004 14005 14006 14007 14008 14009
14010 14011 14012 14013 14014 14015 14016 14017 5274 14018 14019 14020 14021 14022 14023 14024 14025 14026 5011
14027 14028 14029 14030 5168 14031 14032 5550 14033 14034 14035 5251 14036 14037 14038 14039 14040 14041 14042
5014 14043 14044 14045 14046 14047 14048 14049 4990 14050 14051 14052 5933 14053 14054 14055 14056 14057 14058
5813 14059 14060 14061 14062 14063 5015 14064 14065 14066 5531 14067 14068 14069 14070 14071 14072 14073 14074
14075 14076 14077 14078 14079 14080 14081 14082 14083 14084 14085 14086 14087 14088 14089 5716 14090 14091 5924
14092 14093 14094 14095 14096 5749 14097 14098 14099 14100 14101 14102 14103 14104 14105 14106 14107 14108 14109
14110 14111 14112 4918 5619 14113 14114 14115 5593 14116 14117 5166 14118 14119 14120 14121 14122 14123 14124
14125 14126 14127 14128 14129 14130 14131 14132 14133 14134 14135 14136 14137 14138 14139 14140 14141 14142 14143
14144 14145 14146 14147 14148 14149 14150 14151 14152 14153 14154 14155 14156 14157 14158 14159 14160 14161 14162
14163 14164 14165 14166 5081 14167 14168 14169 14170 14171 14172 4852 5256 14173 14174 14175 14176 14177 14178
14179 4872 14180 14181 4996 14182 14183 14184 14185 14186 14187 14188 5580 14189 14190 14191 14192 14193 14194
14195 14196 14197 14198 14199 5472 5231 14200 14201 14202 14203 14204 14205 14206 14207 14208 14209 14210 5232
4750 14211 14212 14213 14214 5538 14215 14216 14217 14218 14219 14220 5788 14221 14222 14223 14224 4676 14225
14226 14227 14228 14229 14230 14231 14232 14233 14234 14235 4720 14236 14237 14238 4987 14239 14240 14241 14242
4955 14243 14244 14245 14246 14247 14248 14249 14250 14251 14252 14253 14254 14255 5004 14256 14257 14258 14259
14260 14261 14262 5918 14263 14264 14265 14266 14267 14268 14269 14270 5197 14271 14272 5866 14273 14274 14275
14276 14277 4789 14278 14279 14280 14281 14282 14283 14284 14285 14286 14287 14288 14289 14290 14291 14292 14293
14294 14295 14296 14297 14298 14299 14300 14301 14302 5241 14303 14304 14305 14306 14307 14308 14309 14310 14311
14312 14313 14314 6058 14315 14316 14317 5642 14318 14319 14320 14321 14322 14323 14324 14325 14326 14327 6040
14328 14329 5569 14330 14331 14332 14333 14334 14335 14336 14337 14338 4631 5315 14339 14340 14341 14342 14343
14344 14345 14346 14347 14348 14349 14350 14351 14352 14353 14354 14355 14356 14357 14358 14359 14360 14361 14362
14363 14364 5534 14365 14366 4775 14367 5993 14368 14369 14370 14371 14372 14373 6085 14374 14375 14376 5337
14377 5708 14378 14379 14380 14381 14382 14383 14384 14385 6069 14386 14387 14388 14389 14390 14391 14392 14393
14394 14395 14396 14397 14398 14399 14400 14401 14402 14403 14404 14405 14406 14407 14408 14409 5529 14410 14411
14412 14413 14414 14415 14416 14417 14418 14419 14420 14421 14422 14423 14424 14425 14426 14427 14428 5755 5732
14429 5356 14430 14431 14432 5192 14433 14434 14435 5733 14436 5351 14437 14438 14439 14440 14441 14442 14443
14444 14445 14446 14447 5598 4816 14448 14449 14450 14451 5603 14452 14453 14454 5448 14455 14456 14457 14458
14459 14460 14461 14462 14463 14464 14465 14466 14467 14468 14469 14470 5765 5370 4965 4715 14471 5416 14472
14473 14474 14475 14476 4805 14477 14478 14479 14480 14481 14482 14483 14484 14485 14486 14487 14488 14489 5503
14490 14491 14492 5982 14493 4678 14494 14495 14496 14497 14498 14499 14500 14501 14502 14503 14504 14505 14506
14507 14508 5822 14509 14510 14511 14512 14513 14514 14515 14516 5591 5361 14517 14518 14519 14520 14521 14522
14523 14524 14525 14526 14527 14528 14529 14530 14531 14532 14533 14534 14535 14536 14537 14538 14539 14540 5278
14541 14542 14543 14544 14545 14546 14547 14548 14549 14550 14551 14552 14553 14554 14555 14556 14557 14558 14559
14560 14561 14562 14563 4921 14564 14565 14566 14567 14568 14569 14570 14571 14572 14573 14574 14575 4988 14576
14577 14578 14579 14580 14581 14582 14583 14584 14585 5513 14586 14587 14588 14589 14590 14591 5875 5050 14592

14593 14594 14595 14596 14597 14598 14599 14600 14601 14602 14603 14604 5572 14605 14606 14607 14608 14609 14610
5364 14611 14612 4645 14613 14614 14615 14616 14617 14618 14619 14620 14621 14622 14623 14624 14625 14626 14627
14628 14629 14630 14631 14632 14633 14634 14635 14636 14637 14638 14639 14640 14641 14642 14643 14644 14645 14646
14647 14648 14649 14650 14651 14652 14653 14654 14655 14656 14657 14658 14659 14660 14661 14662 14663 5512 14664
14665 14666 14667 14668 5836 14669 14670 14671 14672 5737 14673 14674 14675 14676 4842 14677 14678 14679 14680
14681 14682 4737 14683 14684 4833 14685 14686 14687 14688 5359 14689 14690 14691 14692 14693 14694 14695 14696
4650 14697 5293 14698 14699 14700 14701 14702 14703 5063 14704 5941 14705 14706 14707 14708 5938 14709 14710
14711 14712 5710 14713 14714 14715 14716 14717 14718 6086 14719 14720 14721 14722 14723 14724 14725 14726 14727
14728 14729 14730 14731 14732 14733 14734 14735 14736 14737 14738 14739 14740 14741 14742 14743 14744 14745 14746
14747 14748 14749 14750 14751 14752 14753 14754 14755 14756 14757 14758 14759 5165 14760 14761 14762 14763 14764
14765 14766 6048 14767 14768 14769 6071 14770 14771 14772 14773 14774 14775 14776 14777 14778 14779 14780 14781
14782 6073 14783 14784 14785 14786 14787 14788 6101 14789 14790 4854 14791 14792 14793 14794 14795 14796 14797
14798 4617 14799 5753 14800 14801 14802 14803 14804 14805 14806 14807 14808 14809 14810 14811 14812 14813 14814
14815 14816 14817 14818 14819 14820 14821 14822 14823 14824 14825 4598 14826 14827 5233 14828 14829 14830 14831
14832 14833 14834 14835 14836 14837 14838 14839 14840 5017 14841 14842 14843 14844 14845 14846 14847 14848 14849
5107 14850 14851 14852 14853 14854 14855 14856 14857 14858 14859 14860 14861 14862 14863 14864 14865 14866 14867
5752 5511 5522 14868 14869 14870 14871 14872 14873 14874 14875 5313 14876 14877 14878 14879 14880 14881 14882
4796 14883 14884 14885 14886 14887 4840 14888 14889 14890 14891 14892 14893 14894 14895 14896 14897 14898 14899
14900 14901 14902 14903 14904 14905 14906 14907 14908 14909 14910 14911 14912 14913 5152 14914 14915 14916 4687
14917 14918 14919 14920 4722 14921 14922 14923 14924 5537 14925 14926 14927 14928 14929 14930 14931 14932 14933
14934 14935 14936 14937 14938 14939 14940 14941 14942 14943 14944 14945 14946 14947 14948 14949 14950 14951 14952
14953 14954 14955 14956 14957 14958 14959 4951 4759 14960 14961 14962 14963 14964 5144 5464 14965 14966 14967
14968 14969 14970 14971 14972 14973 14974 14975 6042 14976 14977 14978 14979 14980 14981 14982 14983 14984 14985
14986 14987 14988 14989 6066 14990 14991 14992 14993 14994 14995 14996 14997 14998 14999 15000 15001 15002 15003
15004 15005 15006 15007 15008 15009 15010 15011 15012 15013 15014 15015 15016 15017 5044 15018 15019 15020 15021
15022 15023 5826 15024 15025 15026 15027 15028 15029 15030 15031 5544 15032 5219 15033 15034 15035 15036 5396
15037 15038 15039 15040 15041 15042 15043 15044 15045 15046 15047 4714 15048 15049 15050 15051 15052 15053 15054
15055 4922 15056 15057 15058 15059 15060 15061 15062 15063 15064 15065 15066 15067 5222 4855 15068 15069 15070
15071 4937 15072 15073 4618 15074 15075 15076 5587 15077 6089 15078 15079 15080 15081 5656 15082 15083 15084
15085 15086 15087 15088 15089 15090 5578 15091 15092 15093 15094 15095 15096 15097 4643 5605 15098 15099 15100
15101 15102 15103 15104 15105 15106 15107 15108 15109 15110 15111 15112 15113 6067 15114 15115 15116 15117 15118
15119 15120 15121 15122 15123 15124 15125 15126 5860 15127 15128 15129 15130 15131 15132 15133 15134 15135 15136
15137 15138 5953 15139 15140 15141 15142 15143 15144 15145 15146 5030 15147 15148 15149 15150 15151 5006 5329
5121 15152 15153 15154 15155 4762 15156 15157 15158 5730 15159 5013 15160 15161 15162 4917 15163 15164 15165
15166 15167 15168 15169 15170 15171 15172 15173 15174 15175 15176 15177 15178 15179 15180 6054 15181 15182 15183
15184 15185 15186 15187 15188 5611 15189 15190 15191 15192 5824 15193 15194 15195 5419 15196 15197 15198 15199
15200 15201 15202 15203 15204 15205 15206 15207 15208 15209 15210 4685 15211 15212 15213 15214 15215 15216 5909
15217 15218 15219 15220 15221 15222 15223 15224 15225 15226 15227 15228 5787 15229 15230 15231 15232 15233 15234
15235 15236 15237 5389 15238 15239 15240 15241 15242 15243 15244 15245 15246 15247 4683 4832 15248 15249 15250
5494 15251 5687 15252 15253 15254 5723 15255 15256 15257 15258 15259 15260 15261 15262 15263 4757 15264 4754
15265 15266 15267 15268 15269 15270 15271 15272 15273 15274 15275 15276 15277 15278 15279 15280 15281 15282 15283
15284 4591 15285 15286 15287 15288 15289 15290 15291 15292 15293 6077 15294 15295 15296 15297 4606 15298 15299
15300 5404 15301 15302 15303 15304 15305 15306 15307 4670 15308 6098 15309 15310 4589 15311 15312 15313 15314
6074 15315 15316 15317 15318 15319 15320 15321 15322 15323 15324 15325 15326 15327 15328 15329 15330 15331 4583
15332 15333 15334 15335 15336 5888 15337 15338 15339 15340 15341 15342 15343 15344 15345 15346 5970 15347 15348
15349 15350 15351 15352 15353 5688 15354 15355 15356 15357 15358 15359 15360 15361 15362 15363 15364 15365 15366
15367 15368 15369 15370 15371 15372 15373 15374 15375 15376 15377 5720 15378 15379 15380 15381 15382 15383 15384
15385 15386 15387 5423 15388 15389 15390 15391 15392 15393 15394 15395 15396 15397 15398 4817 5793 4698 15399
15400 15401 15402 15403 5210 15404 5574 15405 15406 5740 4941 15407 15408 15409 15410 6109 15411 15412 15413
15414 15415 15416 15417 15418 15419 15420 15421 15422 15423 15424 15425 15426 5202 15427 15428 15429 15430 15431
15432 15433 15434 15435 15436 15437 15438 5211 15439 15440 15441 5073 15442 15443 5779 15444 5696 15445 15446
15447 4629 5398 15448 15449 15450 15451 15452 15453 15454 15455 15456 15457 15458 15459 15460 15461 15462 15463
15464 15465 15466 15467 15468 15469 15470 15471 15472 15473 15474 15475 15476 4845 15477 15478 15479 15480 5772
15481 5596 15482 15483 15484 15485 15486 15487 15488 5949 15489 15490 15491 15492 15493 15494 15495 6021 15496
15497 15498 15499 15500 15501 15502 15503 15504 15505 5818 15506 15507 15508 15509 5326 15510 15511 15512 15513
15514 15515 4994 15516 15517 15518 15519 15520 15521 15522 5154 15523 4663 15524 15525 15526 5734 15527 15528
15529 15530 15531 15532 5059 15533 15534 4943 15535 15536 15537 15538 15539 15540 15541 15542 15543 15544 15545
5273 15546 15547 15548 15549 15550 15551 15552 4760 15553 5399 15554 6056 15555 15556 15557 5303 15558 15559
15560 15561 15562 15563 15564 5705 15565 15566 15567 15568 5852 5842 15569 15570 15571 15572 5838 15573 15574
15575 15576 15577 15578 5981 15579 15580 15581 15582 15583 15584 15585 15586 15587 15588 15589 15590 15591 15592
15593 15594 5350 15595 15596 15597 15598 5122 15599 15600 15601 4774 15602 15603 15604 15605 15606 15607 15608
15609 15610 15611 15612 15613 15614 15615 15616 15617 15618 15619 15620 15621 5280 15622 5748 15623 15624 15625
15626 15627 15628 15629 5996 15630 5863 15631 15632 15633 15634 15635 15636 15637 15638 15639 15640 15641 15642
15643 15644 5653 15645 15646 15647 15648 4940 15649 15650 15651 15652 15653 15654 15655 15656 15657 15658 15659
15660 15661 15662 15663 15664 5876 15665 15666 15667 15668 15669 15670 15671 15672 15673 5984 15674 15675 15676
15677 15678 15679 15680 5725 15681 15682 15683 15684 15685 15686 15687 15688 15689 15690 15691 15692 15693 15694
15695 15696 15697 5659 15698 15699 15700 15701 15702 15703 15704 15705 15706 15707 15708 15709 15710 5214 15711
15712 15713 15714 15715 15716 4592 15717 15718 5777 15719 15720 15721 15722 5271 15723 15724 15725 15726 15727
15728 15729 15730 4870 15731 15732 15733 15734 15735 15736 15737 15738 15739 15740 15741 15742 15743 15744 15745
15746 15747 15748 15749 15750 15751 15752 15753 15754 15755 15756 15757 15758 15759 15760 15761 15762 15763 15764
15765 15766 15767 15768 15769 15770 15771 15772 15773 15774 4647 15775 15776 15777 4931 15778 15779 15780 15781
15782 15783 4672 15784 15785 15786 15787 15788 15789 15790 15791 15792 15793 15794 15795 15796 15797 15798 15799
15800 15801 15802 15803 15804 15805 15806 15807 15808 15809 15810 4697 15811 15812 15813 15814 15815 15816 15817
15818 15819 15820 5223 15821 15822 15823 15824 15825 15826 4841 15827 15828 15829 5711 15830 4878 15831 15832
15833 15834 15835 15836 15837 15838 15839 4875 15840 15841 15842 15843 15844 15845 15846 15847 15848 15849 15850
15851 15852 15853 15854 15855 15856 15857 15858 15859 15860 15861 15862 15863 4721 4807 15864 15865 15866 15867

15868 15869 15870 5099 5421 15871 15872 15873 15874 15875 15876 15877 15878 15879 15880 15881 15882 15883 15884
15885 4749 15886 15887 15888 15889 15890 15891 15892 15893 15894 15895 4682 15896 15897 15898 15899 15900 15901
15902 5853 15903 15904 5521 15905 15906 5770 15907 15908 15909 15910 15911 15912 15913 15914 15915 15916 15917
5565 15918 15919 15920 15921 15922 15923 15924 15925 15926 15927 15928 15929 15930 15931 5991 15932 15933 15934
15935 15936 15937 15938 15939 15940 5879 15941 15942 15943 15944 15945 15946 15947 15948 15949 15950 15951 15952
15953 15954 4621 15955 15956 15957 15958 4744 15959 15960 15961 5584 15962 15963 15964 15965 15966 15967 4788
15968 15969 5220 15970 15971 15972 15973 15974 15975 15976 15977 15978 15979 5415 5798 15980 15981 15982 15983
15984 15985 15986 15987 15988 15989 15990 15991 15992 15993 15994 15995 15996 15997 15998 5254 15999 16000 16001
16002 16003 16004 16005 16006 16007 16008 16009 16010 16011 16012 5029 16013 16014 16015 16016 16017 16018 4880
16019 5510 16020 16021 16022 16023 5914 16024 16025 5332 16026 5485 16027 16028 5456 16029 16030 5707 16031
16032 16033 16034 16035 16036 16037 16038 16039 16040 5317 16041 16042 16043 4966 16044 6068 16045 16046 16047
16048 16049 16050 16051 16052 16053 5102 16054 16055 16056 16057 16058 16059 16060 16061 16062 16063 16064 16065
16066 16067 16068 16069 16070 16071 16072 16073 16074 16075 16076 16077 5376 16078 16079 16080 16081 16082 16083
16084 16085 16086 16087 16088 5032 5305 16089 16090 16091 16092 16093 16094 16095 16096 16097 16098 16099 16100
5463 16101 16102 16103 16104 5228 4895 16105 16106 16107 16108 16109 16110 16111 16112 16113 16114 16115 16116
16117 16118 16119 16120 4997 16121 16122 4864 16123 16124 16125 16126 16127 16128 16129 5618 5467 6065 16130
16131 16132 16133 16134 16135 16136 16137 16138 16139 16140 16141 16142 16143 16144 16145 16146 16147 16148 16149
16150 16151 16152 16153 16154 16155 16156 16157 16158 16159 16160 16161 16162 16163 5480 16164 16165 16166 16167
16168 16169 16170 16171 16172 16173 16174 16175 5078 16176 16177 16178 16179 16180 16181 16182 16183 16184 16185
16186 5782 16187 6049 16188 16189 16190 16191 16192 5243 16193 16194 16195 16196 16197 4839 16198 16199 16200
16201 16202 16203 16204 16205 16206 16207 16208 16209 16210 16211 4596 16212 16213 16214 16215 16216 16217 16218
16219 16220 16221 16222 16223 5489 16224 16225 16226 5391 16227 16228 16229 16230 16231 16232 16233 16234 16235
16236 16237 16238 16239 16240 16241 16242 4574 5160 5340 16243 16244 16245 16246 16247 5458 16248 16249 5692
16250 16251 16252 16253 16254 16255 16256 16257 16258 16259 5543 16260 16261 5333 16262 16263 16264 16265 16266
16267 16268 16269 16270 5810 16271 16272 16273 16274 16275 16276 16277 16278 16279 16280 16281 16282 16283 16284
16285 16286 16287 16288 16289 16290 16291 16292 16293 16294 16295 16296 16297 16298 16299 4916 5433 5872 16300
16301 16302 16303 16304 16305 16306 16307 16308 16309 16310 16311 16312 16313 6053 16314 16315 6016 16316 16317
16318 16319 16320 16321 16322 5365 16323 16324 4978 16325 16326 16327 16328 16329 16330 16331 16332 5246 4790
16333 16334 5033 16335 16336 16337 16338 16339 16340 16341 5270 16342 16343 16344 16345 16346 16347 16348 16349
16350 16351 16352 5069 16353 5401 16354 16355 16356 5615 16357 16358 16359 16360 4977 16361 16362 16363 16364
16365 16366 16367 16368 16369 16370 16371 16372 16373 16374 16375 16376 16377 16378 16379 16380 16381 16382 16383
16384 5320 16385 16386 4733 16387 16388 5417 16389 16390 16391 16392 16393 16394 16395 16396 16397 16398 16399
16400 16401 16402 16403 16404 16405 16406 16407 4826 16408 16409 4616 16410 16411 16412 16413 16414 16415 16416
16417 16418 16419 16420 5294 16421 5977 16422 5796 16423 16424 16425 16426 16427 16428 16429 16430 16431 5469
16432 16433 16434 16435 16436 16437 16438 16439 16440 16441 16442 16443 16444 16445 16446 16447 16448 16449 16450
5832 16451 16452 16453 16454 16455 16456 16457 16458 16459 16460 16461 16462 16463 16464 16465 5669 16466 16467
16468 16469 16470 16471 16472 16473 6017 16474 16475 16476 16477 16478 5959 16479 16480 16481 16482 16483 16484
16485 16486 16487 16488 16489 5153 16490 16491 16492 16493 16494 16495 16496 4706 16497 16498 16499 16500 16501
5498 16502 16503 16504 16505 5238 16506 16507 16508 16509 16510 16511 16512 16513 16514 16515 16516 16517 16518
16519 16520 16521 16522 16523 16524 16525 16526 16527 16528 16529 16530 16531 16532 16533 6062 16534 16535 16536
16537 16538 5634 16539 16540 16541 16542 5314 16543 16544 16545 16546 16547 16548 5409 16549 16550 5944 4767
16551 16552 16553 16554 16555 5547 16556 16557 16558 16559 16560 4627 16561 16562 16563 5393 16564 16565 16566
16567 16568 16569 16570 16571 16572 16573 16574 16575 16576 16577 16578 16579 16580 16581 16582 16583 5766 5585
16584 16585 5384 5207 16586 16587 16588 16589 16590 16591 16592 16593 5090 16594 16595 16596 16597 5104 16598
16599 16600 16601 16602 16603 16604 16605 16606 16607 16608 16609 16610 16611 16612 16613 16614 16615 6018 16616
16617 16618 16619 16620 4848 16621 16622 16623 16624 16625 16626 16627 16628 16629 16630 16631 16632 16633 16634
16635 16636 16637 16638 16639 16640 16641 16642 16643 16644 16645 16646 16647 16648 5662 16649 16650 16651 16652
16653 16654 16655 16656 16657 16658 16659 16660 16661 16662 16663 16664 16665 16666 16667 16668 16669 4601 16670
16671 16672 16673 16674 16675 16676 16677 16678 16679 16680 16681 16682 16683 16684 16685 16686 16687 16688 16689
16690 16691 16692 16693 16694 5484 16695 5877 16696 16697 16698 16699 16700 16701 16702 16703 16704 16705 16706
16707 16708 16709 16710 16711 16712 16713 16714 16715 16716 16717 16718 5381 16719 16720 16721 16722 16723 16724
16725 16726 16727 16728 16729 16730 16731 16732 16733 6002 16734 16735 16736 16737 16738 16739 5568 16740 16741
16742 5221 16743 16744 16745 16746 16747 16748 16749 4660 16750 16751 16752 16753 16754 16755 16756 16757 16758
5997 16759 16760 16761 16762 16763 5633 16764 16765 6019 16766 16767 16768 5052 16769 16770 16771 16772 16773
16774 16775 16776 16777 16778 16779 16780 16781 16782 16783 16784 16785 16786 16787 16788 16789 16790 16791 16792
16793 16794 16795 16796 16797 16798 16799 16800 16801 16802 16803 16804 16805 16806 16807 16808 16809 16810 16811
16812 16813 16814 16815 16816 16817 16818 16819 16820 16821 16822 16823 5960 16824 16825 16826 16827 16828 16829
16830 6043 16831 16832 5855 16833 16834 16835 16836 16837 5134 16838 16839 16840 16841 16842 16843 16844 16845
4728 16846 16847 16848 16849 16850 16851 16852 16853 16854 16855 6103 16856 16857 16858 5138 16859 16860 5064
16861 16862 16863 16864 16865 16866 16867 16868 16869 16870 16871 4773 16872 16873 16874 16875 16876 16877 16878
6093 16879 16880 16881 16882 16883 16884 16885 16886 4811 16887 16888 16889 16890 16891 16892 16893 16894 16895
16896 16897 16898 16899 16900 16901 16902 16903 16904 16905 5321 6095 16906 16907 16908 16909 16910 16911 16912
16913 16914 16915 5366 16916 4799 16917 16918 16919 16920 16921 16922 16923 16924 16925 16926 5926 16927 16928
5986 16929 16930 16931 16932 6081 16933 16934 16935 16936 16937 16938 16939 16940 16941 16942 16943 16944 16945
16946 16947 16948 5382 16949 16950 16951 16952 16953 16954 16955 16956 16957 16958 16959 16960 16961 16962 16963
16964 16965 16966 16967 16968 16969 16970 16971 16972 16973 16974 16975 16976 16977 16978 16979 16980 16981 16982
16983 16984 16985 5930 16986 6030 16987 16988 16989 16990 16991 16992 16993 16994 5115 16995 16996 16997 16998
16999 17000 17001 17002 17003 17004 4838 17005 17006 17007 17008 17009 17010 17011 17012 17013 17014 17015 17016
17017 17018 17019 17020 17021 5614 17022 17023 17024 17025 17026 17027 17028 17029 17030 17031 17032 17033 17034
17035 17036 17037 17038 17039 17040 17041 5923 17042 17043 17044 17045 17046 17047 17048 17049 5205 17050 5418
17051 17052 5259 17053 17054 17055 17056 17057 17058 17059 17060 17061 17062 17063 17064 17065 17066 17067 17068
17069 17070 17071 17072 17073 17074 17075 17076 17077 17078 17079 17080 17081 17082 17083 17084 17085 17086 17087
17088 17089 17090 17091 17092 17093 17094 17095 17096 17097 17098 17099 4679 17100 17101 17102 17103 17104 17105
17106 17107 17108 17109 17110 17111 17112 17113 17114 17115 17116 17117 17118 17119 5354 5430 17120 17121 17122
17123 17124 17125 17126 17127 4711 17128 17129 17130 17131 17132 17133 17134 17135 17136 17137 17138 17139 17140

17141 17142 5272 17143 17144 17145 17146 17147 17148 17149 17150 17151 17152 17153 17154 17155 17156 17157 17158 17159 6013 17160 17161



(a) ***(b)***

(c)

Figure III.12. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,* ***(c)*** *Histogrammes de l'image chiffrée.*

III.4.1.2. Description de PosESecL2

Comme la clé générée par notre première alternative codant le premier niveau de sécurisation, est de taille variable d'une donnée à une autre et qui est égale à la taille de la donnée en terme d'éléments (caractères ou pixels), telle que : taille de la donnée = n éléments => taille de la clé générée = n nombres ; alors la résistibilité à l'attaque exhaustive est strictement dépendante de la taille de la donnée. C'est pourquoi, nous proposons, ici, une variante de PosESecL1 et que nous l'appelons PosESecL2 (Position based Evolutionary Secure Level 2).

L'évolution vers la solution optimale (donnée cryptée) par PosESecL2 est assurée par application des opérateurs de reproduction (croisement, mutation) et de sélection des meilleurs individus. Dans ce qui suit nous ne présentons que les phases du processus évolutionnaire faisant la différence entre les deux algorithmes PosESecL1 et PosESecL2. Il s'agit, principalement, des phases de codage et de reproduction. Toutefois, le même mécanisme est adopté pour créer la population initiale en permutant aléatoirement le chromosome codant l'image originale.

a. Codage

Dans le but d'augmenter le niveau de sécurité du premier algorithme décrit dans la section précédente, nous présentons, ici, le nouveau codage proposé donnant lieu à un nouvel algorithme de chiffrement : PosESecL2. Ce dernier opère sur le codage des différents éléments de la donnée à chiffrer : dans le cas d'une donnée texte et de même que pour PosESecL1, le codage utilisé est l'hexadécimal, toutefois, chaque quatre bits des 16 bits codant un élément sont considérés comme un gène. Pour une donnée image, chacune des

composantes R, V et B est considérée comme étant un gène. Ainsi, la taille de la nouvelle clé sera plus grande que celle générée par le premier algorithme. Elle est d'une taille quatre fois plus grande dans le cas de manipulation des données texte et de trois fois plus grande dans le cas de manipulation d'une donnée image. Le codage proposé est résumé à travers la figureIII.13.

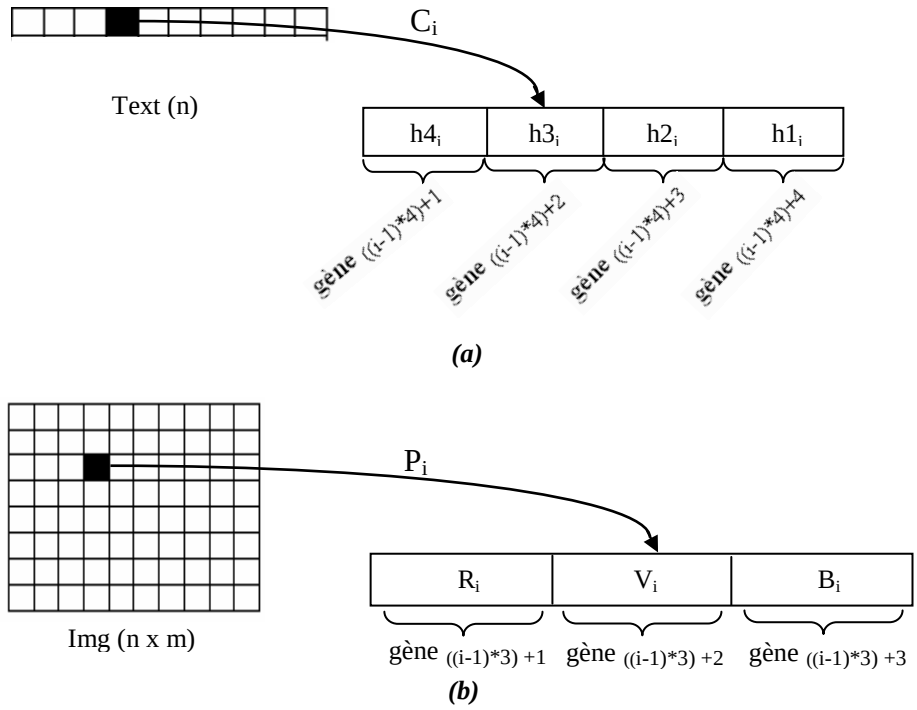


Figure III.13. **(*a*)** *Représentation chromosomale sous PosESecL2 d'un caractère C_i du texte Text(n),* **(*b*)** *Représentation chromosomale sous PosESecL2 d'un pixel P_i de l'image Img(n*m).*

Avec : ($h4_i$ $h3_i$ $h2_i$ $h1_i$) représente le codage Unicode hexadécimal du caractère C_i.

b. Reproduction

Contrairement à l'algorithme PosESecL1, l'algorithme PosESecL2 considère le codage des éléments d'une donnée indépendamment. Ainsi, l'application des opérateurs de croisement et de mutation ne nécessite pas la prise en compte des contraintes à vérifier comme dans le cas de PosESecL1, mais il enrichit d'avantage la population d'individus par de nouvelles caractéristiques.

De même que pour le PosESecL1, l'opérateur de croisement utilisé est le OX « Order Cross-over » appliqué avec un taux fixé expérimentalement et compris entre 60% et 100%. Toutefois, l'opérateur de mutation utilisé est une permutation multipoints pour éviter une convergence trop lente vu la grande différence des tailles des chromosomes dans PosESecL1 et PosESecL2 (le deuxième est d'une taille huit ou trois fois plus grande que le premier). Cet opérateur est appliqué avec un taux, aussi fixé expérimentalement et compris entre 0.1% et 5%.

c. Evaluation des individus

Le choix des individus qui survivront d'une génération à une autre s'effectue selon la fonction objective F, donnée ci-dessous, représentative de l'efficacité des solutions générées sur le problème posé.

$$F(I_i) = \sum_{j=1}^{n \times l} \left| G_{ji} - G_{j0} \right| \quad \text{(III.12)}$$

Avec : G_{ji} est la valeur du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.
l est la taille du codage d'un seul élément de la donnée à chiffrer.

Ainsi, la fonction d'évaluation prendra les deux instanciations suivantes dans le cas de chiffrement de données texte et celui de chiffrement de données images respectivement :

$$F(I_i) = \sum_{j=1}^{n \times 4} \left| G_{ji} - G_{j0} \right| \quad \text{(III.13)}$$

$$F(I_i) = \sum_{j=1}^{n \times 3} \left| G_{ji} - G_{j0} \right| \quad \text{(III.14)}$$

d. Critère d'arrêt

La condition d'arrêt assurant la convergence de l'algorithme PosESecL2 est la même que celle de PosESecL1 puisque le codage des individus dans les deux algorithmes ne diffère que sur le plan contenu de gènes mais les mêmes informations sont présentes dans les deux codages (codage Unicode des caractères du texte à chiffrer ou représentation RVB des pixels de l'image à chiffrer). Autrement dit, c'est la manipulation du codage des éléments de la donnée à chiffrer qui fait la différence entre les deux codages utilisés par les deux algorithmes proposés. Donc, la condition d'arrêt adoptée par ESecL2 dans le cas de chiffrement de données texte et celui de chiffrement de données images, respectivement, est la suivante :

$$0 \le F(I_i) \le F \times 4 \times n \quad / (F)_{16} = (15)_{10} \quad \text{(III.15)}$$

$$0 \le F(I_i) \le 255 \times 3 \times n \quad \text{(III.16)}$$

De même que pour PosESecL1, cette unique condition n'assure pas dans tous les cas la convergence de PosESecL2. C'est pourquoi un nombre maximal de générations sera fixé pour résoudre le problème.

e. Déchiffrement

De même que pour PosESecL1, ce deuxième algorithme utilise le même mécanisme de calcule de clé de chiffrement ; elle change d'un chiffrement à un autre, donc, elle dépend de la donnée à chiffrer et de sa taille. Ainsi, même la taille de la clé de session générée varie d'une donnée à une autre.

f. Réglage des paramètres et résultats

Dans cette section nous présentons les résultats de l'application du deuxième algorithme développé opérant suivant le paramétrage reporté à travers le tableau III.3. À partir des mêmes données originales précédemment utilisées pour tester PosESecL1, nous avons appliqué notre deuxième algorithme PosESecL2 afin d'obtenir les données chiffrées correspondantes. Ces dernières sont celles faisant l'objet des figures III.18.a, III.19.a, III.20.a et III.21.a.

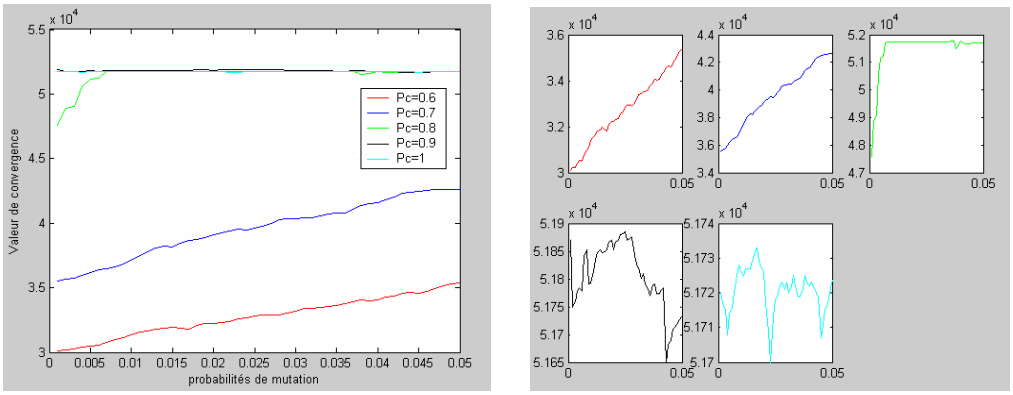


Figure III.14. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

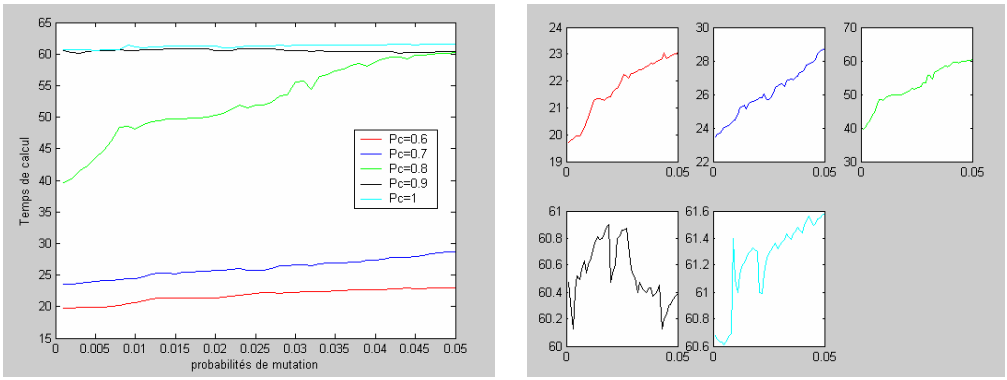


Figure III.15. *Influence des paramètres Pc et Pm sur le temps de calcul.*

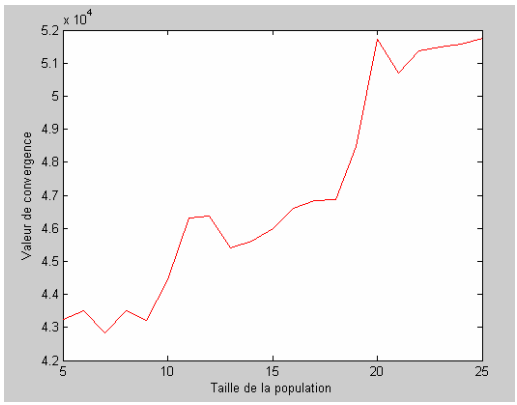


Figure III.16. *Evolution des valeurs de convergence en fonction de la taille de population.*

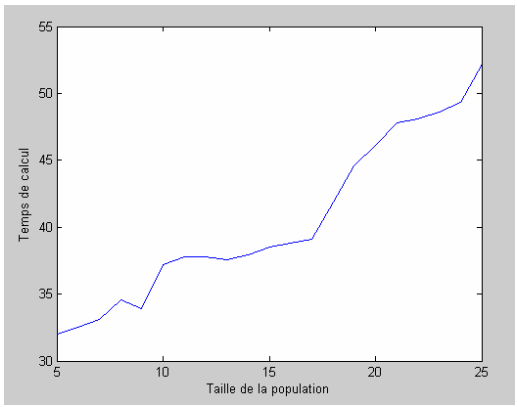


Figure III.17. *Evolution du temps d'exécution en fonction de la taille de population.*

	Valeurs des paramètres de PosESecL2
Taille de la population	20
Nombre de générations	100
P_c	0.8
P_m	0.007

Tableau III.3. *Valeurs adoptées pour les paramètres de PosESecL2.*

L'application de ce deuxième algorithme suivant le paramétrage reporté dans le tableau III.3 sur les mêmes données tests précédentes (Texte 1, Texte 2, Lena et Logo) a permis d'obtenir les résultats résumés à travers le tableau III.4.

D'après ces résultats, il est clair que la taille de la clé augmente proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cette dernière influence la taille des chromosomes résultants du codage utilisé. Donc, c'est la taille des chromosomes augmentant en fonction de la taille des données à chiffrer qui conduit à une augmentation de la taille des clés générées et du temps de calcul correspondant.

		Taille donnée (éléments)	Taille clé (bits)	VC	Temps calcul (s)
Données texte	Texte 1	1084	56368	30729	28.74
	Texte 2	1151	59852	35100	31.5
Données images	Lena	131 X 131	823728	51720	46.15
	Logo	420 X 395	9456300	368712	73.06

Tableau III.4. *Résultats obtenus par PosESecL2.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

´1à1њњ╬‹à+1ﯙî¨à+#cѪéZ0¨£1#§£î¨îﻍàî+6+¨1ﯙcњ♀њ{à0é6¨à+ض¨éﻍ+‹0++qﯙ61ëc¨00c¨î{ﯙî8à‹9¨+њZtt01{î1£#à˚0î♀њ1à+╬0î£01↔
↔¨àaîéë¨+#↔↔¨1´éà0ààà+§8¨£§¨¨0´£è#§é1ë+♀++£11¨{éˀ↔1£ëàﭏ¨++§+à+¨++9§t´£09£ﯙ╬+ëtˀé#éà0ﻍњ↔6àﭏ18ﯙ##ë£¨+0
+¨00‹ˀ+ﯙ+9}1¨˚+8ˀc8££↔à+11++§#‹îﺧﯙZ0ˀ#ﯙﯙëé{§1‹+9é86£009ﯙZﻍ¨cˀ}18+î¨cëîé+§ﭏ0‹0+ﻍﯙ§£§îњ1˚0+1c+ﭏﯙàﯙà+Z#q£18£ﭏﺧﻍ♀ﯙ06#
î1¨ﻍs‹1ﻍ´î#é0#њZ╬§£é↔è+1+ﻍ8ﯙ£+ëà9{њњњ¨éﯙ¨#ﯙ£àt¨é+++00é˚#î1+î#6à‹0#0##àé11+ˀ‹é#ﭏѪZ0#+ﻍ££+↔Ѫ+à1ﺧén£¨¨à#8}ﯙé↔+¨î
╬90+ﯙ+Ѫë¨î‹§1++↔ëéﭏˀ+î£àé#ëZ11ﭏ§#ﭏ♀à#î£+1£é+1ët↔c¨ﺧ+‹ëﭏcﯙîé#îéˀ#ˀî1#˚î¨+8#+1+ﯙ£éﯙé╬¨ˀ1£ﯙZZc8њﻍ#é10ﺧà#+↔08¨╬1#+
£é0té£+88+#^ﯙﭏ9‹î¨î+ﭏ¨{˚£§£+´6§6╬╬↔+¨њњњњà{6+##ﯙ↔↔ëˀ+îà+16îњ1¨11£1£0#9+ﻍà§+é£+ë+#11ˀ£ëî‹î+£‹1+´£00£+ﻍ¨+ﯙé#£19
♀♀c§î+c01àî¨8#8ﯙ+╬à§¨{î¨+60¨§6♀0Z8é#1#à{+1£‹à9#1ˀà6+¨1ﭏ£˚6à+1î1ë+ˀt9£+îî+80£6é#¨+£¨ZZ1c¨њњ{ëc0♀t0#ﯙt}+0è↔{ècc8î#
£¨£#ﯙàà£#+1£+њѪњ╬à˚116↔1ˀ+éîZà£c{´+#

(b)

Figure III.18. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session (Taille$_{\textbf{Clé}}$ = 6,8808 K octets) :

1	450	496	406	440	458	465	500	437	477	491	298	487	444	468	497	441	471	493
494	454	102	225	455	453	484	473	330	429	432	428	447	469	480	436	488	475	74
446	478	490	443	430	486	483	481	466	499	239	449	489	456	120	464	435	498	451
442	479	474	412	434	467	336	485	5	457	101	213	448	470	431	460	463	438	307
476	445	461	427	459	462	290	495	482	472	433	157	439	426	452	740	1330	545	1331
508	1332	975	1333	564	1334	1335	961	818	1336	886	812	1337	908	674	875	1054	1338	808

725	1339	1340	1341	1342	1343	1344	786	732	1226	1297	228	635	1345	1346	863	696	758	1347
689	1348	1349	1350	710	126	855	548	756	667	522	892	604	1351	1352	602	1353	146	920
770	691	678	729	901	1354	876	1025	1355	4	3	6816	609	993	1356	1357	1229	1358	309
1359	1360	502	1361	706	1362	1363	1364	1365	880	911	644	784	1366	773	359	768	1367	811
299	940	578	198	563	1368	149	957	742	668	687	833	738	1369	623	878	852	636	1370
634	253	1371	873	586	1151	836	620	1372	948	1004	928	1373	799	882	830	270	684	847
964	550	759	1374	870	937	1375	765	905	514	1376	528	739	583	959	337	1377	693	752
612	796	540	1378	552	680	1379	510	946	216	637	630	992	127	978	1309	643	1380	516
1381	559	889	1382	797	567	1383	645	1384	820	949	844	747	1385	866	1386	1387	1160	848
652	384	1388	560	976	985	603	1003	942	501	1389	633	841	974	1093	1390	657	39	912
1391	982	1392	1393	1394	699	750	894	794	1395	1396	932	864	1397	987	716	807	509	676
708	505	1398	838	1399	231	1400	315	339	523	556	326	673	1020	737	599	577	791	751
1329	613	898	1401	1402	419	1001	1403	1404	900	1405	1406	492	1407	536	766	1408	1409	702
787	1410	187	924	1411	845	793	1412	1138	804	1413	1414	1415	546	731	1416	18	1417	66
1418	650	511	662	1288	971	715	817	962	694	1419	968	631	966	537	654	364	520	1104
850	1193	150	981	782	1420	543	915	827	1421	995	1422	843	542	934	1423	1424	775	517
642	709	9	1425	607	772	529	80	994	842	819	712	639	656	2	989	1426	938	610
851	735	666	832	651	541	600	227	241	884	781	895	175	515	417	1237	525	825	734
965	278	890	138	1427	1428	1429	596	1430	553	1293	923	589	730	1431	555	917	1261	663
1432	1433	1000	58	130	1434	790	672	1435	534	1436	1437	1169	44	805	641	800	1438	958
1291	954	1439	1440	1441	1442	835	1443	640	595	1444	933	717	700	888	1445	538	601	616
1446	675	617	701	972	1447	757	909	1448	918	931	1171	704	660	573	998	1449	871	771
1450	763	906	1451	185	1116	557	1452	1453	711	403	916	983	1454	769	1455	221	979	713
1456	846	903	813	839	743	558	910	1295	585	1457	597	1458	1459	1460	967	1461	1462	310
362	313	697	1463	1464	935	719	16	105	856	135	921	952	726	896	1240	1465	705	927
930	1466	397	1467	720	647	980	1468	829	575	950	1469	580	883	664	527	746	323	780
561	614	877	611	828	776	1470	626	1471	570	681	688	569	622	618	879	1472	810	973
572	1473	314	1474	615	683	1475	1476	108	627	1264	665	343	714	382	52	801	1287	152
1477	521	579	28	1478	1479	721	532	1480	1481	549	951	874	46	91	744	592	284	1210
254	1482	206	1483	606	646	945	872	767	320	774	55	348	1484	593	698	535	1485	659
1486	1155	1487	1488	1489	670	1490	990	1491	240	518	815	1492	653	1493	754	899	857	755
798	1494	887	861	1495	143	897	544	996	503	1496	1497	1243	1498	1499	824	1500	1501	1502
533	551	840	977	1503	837	590	1504	988	1505	1506	1507	565	963	788	727	777	1508	1509
809	1510	109	1511	174	1512	869	547	1234	986	904	991	638	1513	999	598	1514	566	722
748	1515	513	802	524	96	1516	1517	1518	919	245	679	507	255	690	1519	733	506	939
922	685	860	1520	853	865	628	504	1521	1522	1523	941	1037	1524	686	1525	355	1526	677
1527	1528	295	914	1189	834	749	661	718	581	1529	764	1530	1531	785	280	823	854	1532
1533	244	859	658	519	648	649	1534	831	984	792	608	554	122	1535	1536	1537	955	723
1538	947	1539	997	1540	568	1541	582	1031	1542	1543	692	1207	588	821	956	671	418	1544
1545	760	1546	943	1547	1548	902	1549	703	1550	1289	695	574	728	745	41	594	970	587
1551	1552	233	1553	1554	655	1555	1556	421	1557	1558	893	562	803	7	814	1559	891	1560
1142	512	1561	629	1562	311	936	73	530	1563	1564	1565	783	1566	1567	1568	137	881	669
624	926	1569	741	415	736	753	953	302	1073	318	1570	779	1571	162	1572	1573	867	266
1574	605	1064	1575	334	90	12	1200	576	1576	1577	969	762	907	1038	619	862	1272	1578
1579	571	822	526	1580	45	49	913	795	1315	761	960	849	944	95	625	1581	1582	201
1583	1005	929	1584	1585	632	531	925	591	1586	789	868	300	682	1587	826	584	885	707
539	1002	806	621	778	724	858	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
1600	1159	1276	1601	1103	1602	1603	1604	1605	1606	1607	128	1608	1609	1610	1611	1612	1613	1614
34	1615	1328	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
1632	1317	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649
1062	1650	1651	1652	1653	1654	1655	1656	1657	1246	1658	1659	1660	1661	1662	1663	1664	1132	1665
1666	1667	349	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683
1684	199	1685	1083	1686	1687	1688	1689	1690	1691	1692	1211	166	1693	1694	1695	1696	1697	291
1247	1153	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1084	1709	1710	1711	140	1712
1713	1714	196	1715	1716	141	1717	1718	1719	1720	1721	1722	1723	1724	1327	287	1725	1726	1009
1727	1728	1729	1730	235	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744
1745	1746	1747	1748	1749	1750	1167	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762
1763	1764	1765	383	57	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779
1780	1144	1781	1185	1782	1783	1784	1294	1785	1786	1324	1787	1156	1788	1789	1790	1791	1792	1793
1195	1281	1794	1795	1796	1797	1798	1799	1800	1014	1801	1802	246	1803	1804	1805	1806	1145	1807
1808	1809	1810	1811	79	1812	1813	1814	1050	1274	286	1815	1816	1817	1818	1819	1820	1821	1822
1823	1824	1825	374	1186	1826	1827	395	1828	1112	1829	1830	112	1831	1832	82	1833	1834	1835
1836	1837	1838	1839	1840	1091	1841	1842	1843	1844	1845	1846	1847	1848	269	1849	1850	17	1275
1851	1177	204	1852	1853	1854	1855	1856	1857	1858	1859	1860	1216	1861	1862	1863	1864	1865	142
1866	1867	1090	1868	1869	1870	56	27	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1039
1881	1882	1883	1884	1075	1885	1886	154	1887	1888	1115	1889	1063	1890	1891	1892	208	1893	1894
1895	1896	1897	1898	1899	1900	1901	1902	1903	1107	1904	1905	1906	1907	1908	1909	1910	1911	1912
1913	1914	1915	1916	125	1917	1918	169	1919	1920	1921	1922	1923	1924	1925	1926	1927	398	1928
1929	1930	1931	1267	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	352	1943	1051	1944
1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	8	1957	1958	1959	1960	1961	1962
293	1963	1964	1965	1966	1967	1968	53	1969	1970	1296	1971	1015	1972	1973	1974	1975	1976	1977
1978	1979	1074	1980	1279	1981	312	1072	1982	1983	1984	1985	1986	264	1987	1988	1989	265	1990
1991	1992	1993	1994	1995	1996	1997	1998	1999	1017	2000	2001	2002	252	2003	2004	2005	342	2006
2007	2008	2009	2010	2011	1130	2012	2013	2014	2015	2016	2017	2018	2019	2020	1041	2021	2022	2023
1058	133	2024	2025	2026	2027	2028	2029	2030	2031	2032	1184	2033	2034	2035	2036	2037	1187	2038
2039	1136	2040	381	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055

1304	305	410	2056	2057	2058	2059	2060	10	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070
2071	2072	2073	2074	35	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088
2089	2090	2091	283	2092	2093	2094	2095	2096	2097	2098	1069	2099	2100	2101	170	2102	2103	2104
2105	2106	2107	2108	2109	2110	1111	2111	297	226	2112	2113	64	2114	2115	267	2116	33	2117
2118	2119	2120	2121	2122	2123	2124	288	2125	2126	2127	2128	2129	2130	2131	176	2132	2133	2134
2135	2136	2137	214	2138	2139	2140	2141	2142	2143	2144	1179	356	2145	2146	2147	2148	2149	2150
2151	2152	2153	2154	2155	2156	2157	2158	2159	99	2160	2161	2162	2163	2164	2165	139	2166	2167
2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	89	2182	2183	2184	153
2185	2186	2187	54	2188	2189	2190	2191	2192	2193	2194	186	2195	1125	2196	2197	2198	2199	2200
1057	2201	2202	2203	2204	2205	261	2206	1158	2207	2208	2209	2210	2211	1087	2212	2213	2214	2215
1273	2216	2217	2218	2219	2220	2221	2222	2223	1230	2224	2225	2226	2227	2228	2229	229	2230	2231
2232	2233	2234	2235	2236	2237	2238	2239	2240	86	2241	2242	1266	2243	2244	103	2245	2246	2247
2248	2249	2250	2251	1235	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	1231	1137	2262	2263
2264	2265	2266	2267	2268	219	2269	2270	376	2271	2272	2273	2274	2275	2276	2277	1113	2278	2279
2280	2281	304	2282	2283	2284	2285	2286	72	2287	2288	2289	2290	2291	394	2292	2293	2294	24
2295	2296	2297	2298	1258	165	2299	1259	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
107	2311	2312	2313	2314	2315	1110	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327
2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	136
2346	1008	1024	2347	2348	2349	2350	2351	2352	2353	404	2354	1232	2355	2356	2357	2358	2359	2360
2361	2362	1102	2363	2364	2365	2366	371	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377
2378	2379	2380	2381	2382	2383	1217	1046	2384	2385	2386	2387	2388	2389	2390	2391	2392	123	2393
2394	2395	379	2396	2397	2398	2399	2400	1203	2401	274	1223	1071	2402	388	2403	2404	2405	1081
2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	1019	2416	2417	2418	2419	2420	2421	2422	2423
2424	2425	2426	2427	2428	2429	2430	2431	2432	40	2433	2434	2435	2436	2437	2438	2439	2440	1162
2441	2442	2443	1109	2444	306	2445	1325	2446	2447	2448	2449	2450	2451	2452	2453	2454	134	2455
2456	2457	76	2458	2459	2460	43	2461	2462	2463	2464	2465	129	2466	2467	2468	2469	2470	2471
399	191	2472	2473	2474	2475	2476	2477	405	2478	2479	2480	2481	2482	2483	2484	2485	2486	2487
2488	294	2489	2490	2491	2492	2493	2494	2495	2496	177	2497	2498	276	247	2499	2500	256	2501
2502	2503	2504	2505	2506	2507	389	2508	2509	2510	2511	2512	2513	2514	2515	2516	1012	346	2517
2518	350	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535
2536	2537	2538	2539	2540	2541	2542	2543	1222	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553
1206	2554	2555	2556	369	2557	2558	2559	292	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569
2570	2571	2572	2573	2574	1028	2575	1205	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586
104	2587	2588	2589	2590	2591	2592	2593	2594	328	2595	2596	1120	2597	2598	2599	2600	2601	2602
2603	2604	2605	2606	2607	316	236	2608	2609	1286	2610	2611	2612	2613	2614	2615	2616	2617	2618
411	2619	2620	2621	2622	2623	2624	2625	2626	1016	2627	365	2628	1214	2629	2630	2631	2632	2633
2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652
2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	19	2664	2665	2666	2667	2668	2669	2670
2671	1176	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688
2689	366	2690	2691	2692	2693	2694	2695	2696	2697	2698	271	2699	172	2700	1114	2701	2702	2703
2704	2705	193	2706	2707	2708	2709	205	2710	2711	1122	2712	2713	2714	2715	2716	1248	2717	2718
1150	2719	2720	2721	2722	2723	2724	189	373	2725	2726	2727	2728	2729	1183	2730	2731	2732	2733
2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	1182	2748	2749	2750	1199
2751	2752	1036	2753	420	2754	2755	145	1148	2756	2757	2758	2759	2760	1194	2761	2762	2763	1106
2764	2765	2766	115	2767	2768	368	2769	2770	163	2771	2772	2773	2774	2775	390	2776	2777	2778
2779	111	113	2780	248	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794
2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	173
2813	2814	2815	168	2816	2817	2818	2819	1061	2820	2821	1249	2822	1094	2823	159	2824	2825	2826
2827	2828	2829	2830	2831	2832	2833	2834	2835	1146	2836	1022	2837	2838	2839	1027	2840	2841	2842
48	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	360	2857	2858	2859
2860	2861	2862	2863	2864	1209	22	178	2865	2866	1181	2867	2868	2869	2870	1085	32	14	2871
1174	2872	2873	2874	2875	2876	1307	1134	1096	2877	1218	2878	2879	2880	2881	2882	2883	2884	2885
2886	329	2887	2888	144	2889	2890	2891	2892	2893	2894	2895	2896	1066	2897	2898	2899	2900	2901
2902	2903	2904	1244	409	2905	400	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917
2918	2919	1170	2920	2921	2922	132	42	2923	2924	2925	2926	2927	2928	2929	2930	77	2931	2932
211	1143	1121	242	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943	1192	2944	2945	422
2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	87	2961	2962	2963
2964	2965	2966	2967	1040	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	322	2978	2979	2980
2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999
3000	3001	1013	3002	3003	37	3004	3005	3006	3007	3008	1300	3009	3010	1250	3011	3012	3013	257
3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029	1166	3030	3031
3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	1262	3043	3044	1313	1149	372	1078	3045
3046	3047	3048	3049	3050	3051	3052	1215	3053	3054	3055	3056	3057	3058	3059	3060	3061	1265	3062
3063	3064	181	3065	3066	3067	268	3068	3069	3070	3071	182	3072	3073	3074	3075	3076	3077	3078
401	3079	3080	3081	3082	161	3083	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	1178
3095	1268	3096	3097	3098	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	1055	3110	3111
3112	3113	3114	1256	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	1204	3125	3126	3127	3128
3129	3130	3131	3132	3133	3134	3135	3136	3137	3138	50	3139	3140	1131	3141	3142	3143	3144	3145
3146	3147	3148	3149	151	3150	3151	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	1320
3163	317	1070	327	3164	3165	1242	3166	3167	1299	402	3168	3169	367	3170	3171	3172	3173	3174
3175	3176	3177	3178	131	3179	94	3180	3181	3182	1154	3183	1047	3184	3185	3186	1168	3187	3188
363	3189	3190	3191	3192	3193	3194	258	3195	3196	121	3197	3198	93	3199	319	3200	1099	3201
3202	3203	202	3204	3205	354	416	3206	3207	3208	3209	3210	3211	1316	3212	3213	3214	3215	3216
3217	3218	3219	3220	3221	303	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	243	3233
1269	3234	1233	3235	3236	3237	85	3238	3239	3240	3241	3242	3243	1319	1213	3244	3245	3246	3247
3248	218	1225	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	1048	3259	3260	119	3261	3262

3263 3264 3265 3266 3267 3268 3269 238 3270 3271 3272 3273 3274 3275 3276 223 3277 1032 3278
3279 1101 3280 3281 3282 3283 1076 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 1260 3294
3295 3296 3297 3298 3299 3300 279 3301 3302 3303 3304 3305 118 3306 3307 3308 1129 3309 3310
3311 97 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328
3329 1135 3330 3331 3332 3333 184 3334 370 3335 3336 224 3337 3338 3339 1191 3340 3341 3342
3343 110 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 1117
1254 3360 3361 3362 3363 3364 3365 188 3366 3367 3368 1157 1147 3369 155 3370 3371 3372 1282
21 3373 3374 3375 3376 1241 3377 3378 425 3379 3380 3381 3382 3383 1190 3384 3385 1033 3386
3387 3388 3389 3390 3391 413 3392 3393 1065 3394 3395 285 1228 3396 3397 3398 3399 3400 3401
3402 3403 3404 194 3405 3406 3407 321 230 1098 3408 3409 3410 3411 3412 3413 197 3414 3415
3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 1251 3428 3429 3430 3431 3432 3433
1152 3434 3435 353 1030 3436 1292 3437 3438 215 1180 3439 3440 3441 3442 3443 3444 3445 3446
3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 164 3457 3458 3459 361 3460 3461 3462 3463
3464 3465 3466 3467 25 3468 3469 331 3470 3471 3472 3473 3474 3475 1270 3476 195 3477 3478
3479 3480 3481 3482 3483 3484 3485 3486 20 340 3487 3488 3489 3490 84 3491 3492 3493 3494
3495 3496 1290 209 1108 1006 3497 3498 3499 3500 3501 190 3502 3503 3504 3505 3506 1018 3507
3508 3509 3510 1224 1053 1124 207 3511 212 3512 3513 3514 3515 1285 114 3516 3517 3518 3519
324 3520 3521 3522 3523 3524 3525 3526 1221 3527 3528 158 3529 3530 3531 3532 3533 1236 3534
335 3535 3536 1172 3537 3538 3539 3540 3541 3542 1035 3543 1318 1119 1220 1011 3544 3545 3546
3547 3548 3549 3550 3551 3552 3553 3554 424 3555 3556 387 3557 3558 3559 3560 3561 3562 3563
3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 156 3574 3575 3576 3577 3578 3579 3580 3581
3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 1118 3592 1303 3593 3594 3595 3596 31 263
3597 3598 3599 3600 3601 3602 3603 69 3604 3605 3606 3607 3608 3609 1133 3610 3611 3612 3613
1196 3614 3615 3616 1219 1173 3617 396 1188 3618 3619 3620 3621 282 3622 3623 3624 3625 3626
3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 1239 1198 3637 3638 3639 408 3640 3641 3642
1049 1089 3643 3644 1306 3645 3646 3647 3648 3649 296 3650 3651 3652 1271 3653 3654 3655 160
3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674
3675 3676 3677 3678 3679 3680 1305 3681 3682 3683 1284 3684 3685 3686 3687 3688 3689 3690 3691
3692 3693 3694 3695 3696 3697 3698 15 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 217
3709 3710 1127 3711 1298 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725
1140 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 1067 3738 3739 3740 3741 180
3742 3743 3744 1023 3745 3746 3747 3748 3749 3750 3751 3752 1161 3753 332 3754 3755 3756 3757
1252 3758 3759 3760 3761 3762 171 3763 3764 3765 3766 3767 3768 100 3769 3770 3771 3772 3773
3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 192 3787 3788 3789 3790 3791
3792 3793 3794 3795 3796 3797 262 1082 333 3798 3799 124 3800 3801 3802 1312 1052 3803 3804
3805 3806 391 3807 3808 1278 3809 1042 83 3810 347 3811 3812 3813 70 3814 3815 3816 3817
3818 3819 3820 3821 3822 3823 3824 3825 1126 3826 3827 3828 3829 3830 3831 3832 3833 3834 98
3835 51 81 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 1255 3846 3847 67 3848 3849
3850 3851 3852 3853 3854 3855 1208 3856 3857 30 3858 3859 3860 3861 65 301 47 1021 3862
3863 3864 3865 3866 3867 3868 29 3869 3870 3871 3872 3873 3874 3875 1253 3876 3877 3878 3879
1321 3880 3881 3882 3883 63 3884 71 3885 3886 3887 3888 275 1175 3889 3890 3891 3892 3893
3894 3895 1045 3896 3897 3898 3899 423 260 3900 3901 3902 308 3903 3904 3905 3906 26 59
3907 3908 3909 1322 3910 3911 3912 3913 3914 3915 1197 1308 3916 3917 3918 3919 3920 3921 3922
3923 3924 1086 3925 3926 3927 3928 1263 68 1128 272 3929 3930 3931 3932 3933 167 3934 3935
3936 3937 3938 3939 3940 3941 3942 147 3943 1034 3944 3945 3946 3947 3948 3949 3950 3951 3952
3953 3954 3955 3956 3957 3958 3959 3960 357 3961 3962 3963 3964 3965 3966 3967 1165 1277 3968
38 3969 3970 232 3971 3972 3973 3974 3975 277 3976 3977 3978 3979 3980 3981 3982 3983 3984
3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003
13 4004 4005 4006 1202 4007 4008 4009 4010 4011 4012 4013 4014 4015 1314 4016 4017 4018 1326
4019 4020 4021 1141 4022 250 4023 4024 4025 4026 4027 4028 4029 4030 1201 4031 1227 4032 386
4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 1095 179 4043 4044 4045 1080 4046 4047 4048
4049 75 1029 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 237 4060 4061 4062 4063 4064
4065 4066 203 4067 4068 1301 414 249 4069 4070 380 4071 4072 1283 4073 4074 4075 78 4076
378 4077 61 4078 4079 4080 4081 4082 4083 4084 88 4085 4086 325 4087 1044 4088 4089 4090
200 4091 377 4092 4093 1164 4094 4095 4096 4097 222 4098 4099 4100 4101 4102 1059 4103 4104
4105 4106 4107 1026 4108 4109 1100 4110 358 4111 4112 1238 4113 1139 4114 4115 4116 4117 4118
4119 4120 1060 1302 4121 4122 4123 4124 4125 4126 4127 344 148 4128 4129 4130 4131 1092 4132
4133 4134 4135 4136 4137 4138 4139 4140 4141 1310 273 4142 4143 4144 4145 4146 4147 4148 4149
4150 4151 1163 116 4152 4153 4154 4155 259 251 4156 4157 4158 4159 4160 4161 4162 407 4163
4164 375 60 4165 4166 4167 1043 4168 4169 4170 4171 36 4172 4173 4174 1105 385 4175 4176
4177 4178 4179 4180 4181 4182 4183 220 4184 1077 4185 4186 4187 1007 4188 4189 4190 106 4191
11 4192 4193 1280 4194 4195 4196 4197 4198 4199 4200 289 92 4201 4202 4203 4204 4205 4206
1311 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223 4224
4225 4226 1097 1123 4227 4228 4229 4230 62 4231 351 4232 4233 4234 4235 4236 4237 4238 4239
4240 1257 4241 4242 4243 4244 4245 4246 4247 4248 1056 4249 183 4250 1245 4251 4252 4253 4254
4255 4256 4257 338 4258 4259 4260 4261 117 341 4262 4263 4264 1079 4265 4266 4267 4268 4269
4270 4271 4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282 4283 4284 4285 4286 4287 4288
4289 4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303 4304 4305 4306 4307
4308 345 4309 4310 234 4311 4312 281 4313 4314 1212 1088 4315 4316 1323 4317 4318 4319 392
210 4320 4321 4322 4323 4324 393 4325 4326 4327 4328 4329 4330 1010 23 4331 4332 4333 4334
1068 4335 4336

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملا أساسيا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردى بعدما كان الأمر مقصوراً فى الماضى على المراسلات الرسمية أو السرية بطبيعة الحال.

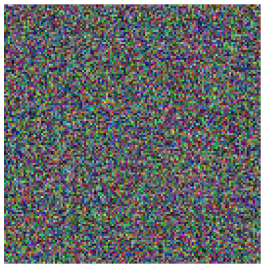
(a)

8ʊ8▒8}¢j^∂ز╔╔ש╔╔j8♂♂®±ĀʊĀ¦88¦(▒}÷╔♂}j¢∂àא`j0ıoز®שʊʊâ♠â&∑▒÷(8♠┤÷┤j∂
(&g8}¦`¦8j(ë÷Ж÷Жjv`♂8±┤¢Æ8^){&ıoıo88j&שÓÖ¦ʊ¢א3ز÷`!ë}ıo¦(ز∑┤¦÷÷3j8®Ā&#j8∂ë8â}╔÷8#8ʊʊʊıoâ:}jj8(&שıoıo8┤j╔8®ĀĀ3¢±
88ë(¦àjא∂}:8¢¢jj¦ç83÷8`ز&∂0&ز®&j÷8▒â┤}Ö&}¢}!j┬jâ3¦¦#`8┤jÖ&Ā8¢â¦±¢j3♠8}ʊ♠Ā3¢`¦ë3±▒▒▒ت&÷j¢ے 8!8┤
∂Æ3÷}jj}¦¢¦&¦8♠♠♠j3ت¦ز¦ıo¦¢8∂8jت÷ʊâ÷^â¢{ز:8÷¢ز∂8¢±¢ıoıoıo╔╔8!¢¦8¢â♂∑∂∑+_¦&زÆ ے {÷`&
8jj(&▒Æ╔╔╔â8┤א8Ā&א÷^±3∑8}{¢ت Ж®¢8¦&♠&¦ت▒0א8±ש∑∑Ā&(&}3¦â3¦®8&â&(÷àj¦(¦ے
ëâ_8ززjj&88&88ت&àj(¦±╔+!}÷ززت÷} Ж&&}ʊجë:8÷╔Ö8!¢ ש}8±8Öے ت╔8{8&8*!ت: { ʊز±&
ے®3ëâ3زشززjʊא0}∂¢j÷&gıoıo¢3┤♠}3♠j8÷âj88زg0¢¢∂ıo∂¦â?ت÷┤▒8÷&j♠+Æ8¢^±ت(`jıo^▒א ▒ز (àت Ā3: ıo&8±ש¢`3ژ88┤ ے ıo3Ж8ے
8:8}±}8زj`88#┤¦&j¦88: Ö3ز8¢¦}∑8&▒زj&8ЖК (¢&ز®
(j88ژ¦ت÷ف}ë÷&8÷&♂¢88ززjh&Æ&÷8¦3&¦88¢8!{¢)¢&8÷ز±±}^!âj«!â¦à8&&!8ز
♠┤&▒8{j8j±¦ש¦j8à÷:8®8ز±∑3j&3ששש8±┤}j}j÷÷+i#∂j¦(3¦ز3┤àĀЖ}3! ש±÷â83&â0¦â3`0&ت▒ے ز ë83شàא∂ıoא∂∂3&8┤Æے
Æ8^â♠^3}&â¦┤┤ز^Öj¦â888ے
â}÷&}&88÷ز¢ز8ز*:zj3¢}j¢زא8&jزıoj}8¢¢ز♠♠8â&┤+3¢8¢ت:âjژ8{+┤`âà÷#8¦`}ز÷âЖ&j:`jà÷83∑Æj÷÷j`â÷}÷*Öj30₪8÷8زà*8¢┤÷ز±÷
3÷¢8«8±┤&j8¦8â÷8╔8¢j88±8ز♠8∂ز3j&♠¦¢ë╔┤÷زâ388àzj¦¦0ت®®÷jâ┤}÷«ɜ3â8}₪8▒&8â3*¦₪₪¢â¦¦â÷¦88¦*8┤ʊ±¦à÷^¦שЖ88∑8ë¦jâĀĀĀ&8
*08#«8ЖÆ8j#¢ʊʊâ3÷!زÖ&┤{{*ÖÖÖשששש∂88ژتj┤÷¦àà: ╔▒▒ש8ز÷¦0Æ♠}j&}∂}jزʊא₪ЖК╔Жâ¦÷ت₪ı┤¦¢ששز}3¢8ت₪±jâ∑!÷38j}¦÷♠

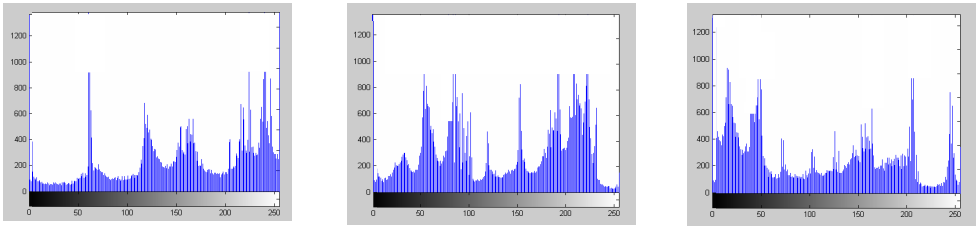
(b)

Figure III.19. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

(a)

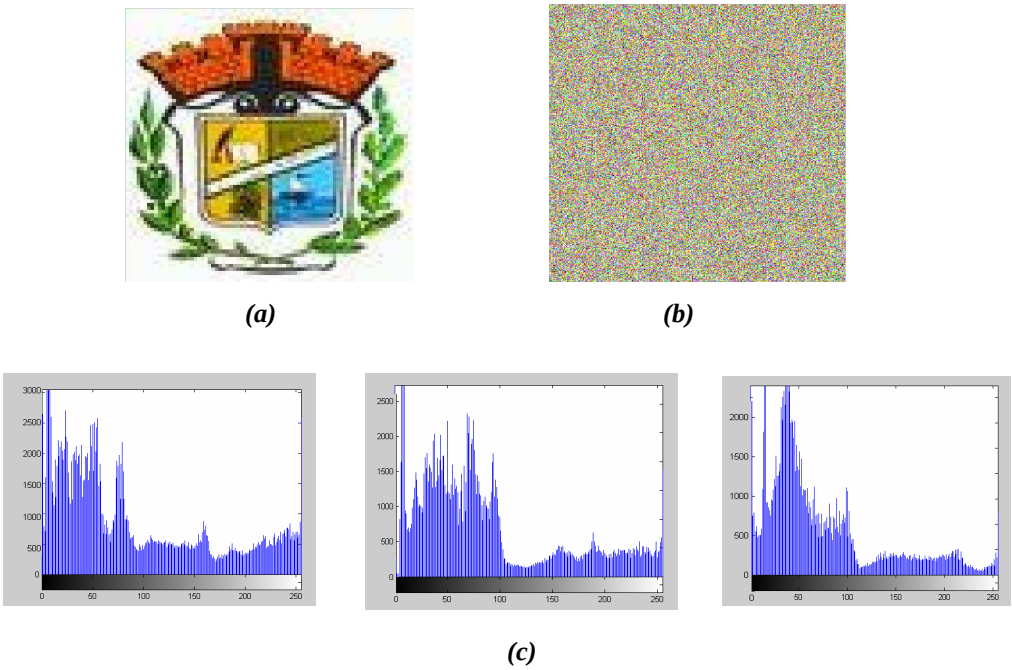


(b)



(c)

Figure III.20. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*



(a) ***(b)***

(c)

Figure III.21. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

III.4.2. Chiffrement à base d'occurrences

Nous avons vu dans la section précédente que le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, dans la présente section nous allons présenter un nouveau codage totalement différent de ceux vu dans les sections précédentes qui ont été bâti autour de la notion de positions de la représentation des éléments constituant la donnée à chiffrer.

Pour appliquer l'approche évolutionnaire à notre problème, nous présenterons les choix utilisés en terme de : représentation chromosomale des solutions du problème, méthode de création de la population initiale des solutions, fonction d'évaluation, opérateurs génétiques et procédure de sélection. Pour mieux approcher le problème, nous présentons une formalisation de ce dernier.

III.4.2.1. Formalisation du problème

Soit D une donnée à chiffrer (texte ou image) qui est une suite de n éléments e_i, et que l'on peut formaliser comme suit :

$$D = \{e_i\}, i \in [1,n] \qquad \text{(III.17)}$$

Dans le cas où D soit une donnée texte, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{C(e_i)\}, i \in [1,n] \qquad \text{(III.18)}$$

où : $C(e_i)$ représente le codage Unicode du caractère e_i.

Dans le cas où *D* soit une donnée image, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{p_i\}, i \in [1,n] \quad \text{(III.19)}$$

Cette représentation (structure) distinguant les différents pixels formant cette image, facilite la conversion en une autre structure en choisissant le RGB (Red Green Blue) comme espace de représentation d'images. Il suffit de remplacer les n pixels par leurs représentations correspondantes en RGB. Nous obtenons, ainsi, la formalisation suivante :

$$D = \{R_i G_i B_i\}, i \in [1,n] \quad \text{(III.20)}$$

Ce mode de représentation (structure) permet de distinguer les différents codages d'éléments de la donnée à chiffrer (les codages des caractères d'une donnée texte, ou les différentes composantes codant les pixels d'une donnée image), et par conséquent de déterminer le nombre d'occurrences de chacun d'eux. C'est ce mécanisme qui sera d'ailleurs exploité pour bâtir notre suivant algorithme de chiffrement proposé.

III.4.2.2. Description d'OEEA (*Occurrences based Evolutionary Encryption Algorithm*)

Comme l'opération de chiffrement consiste à perturber la donnée originale de telle manière à avoir le maximum de désordre dans la donnée chiffrée, nous avons choisi, cette fois-ci, de jouer sur les nombres d'occurrences des éléments (caractères d'un texte ou valeurs des composantes R, G et B codants les pixels d'image), pour avoir la plus grande différence entre les nombres d'occurrences des valeurs similaires d'éléments (nombres d'occurrences des caractères d'un texte original et leurs nombres d'occurrences dans le texte chiffré correspondant ; ou nombres d'occurrences des valeurs des composantes R, G et B codant les pixels d'une image originale et leur nombre d'occurrence dans l'image chiffrée). Ce choix est dû au fait que cette unique donnée (nombres d'occurrences) ne représente pas une information pertinente pour les cryptanalystes.

Dans ce qui suit, nous décrierons les différentes étapes de notre nouvel algorithme de chiffrement proposé.

a. Codage

Les structures (III.18) et (III.20) vues précédemment représentent les données de base à partir desquelles nous sommes arrivés à définir notre codage d'individus en calculant le nombre d'occurrences des valeurs possibles d'éléments dans une certaine donnée. Ainsi, le codage proposé sera le suivant :

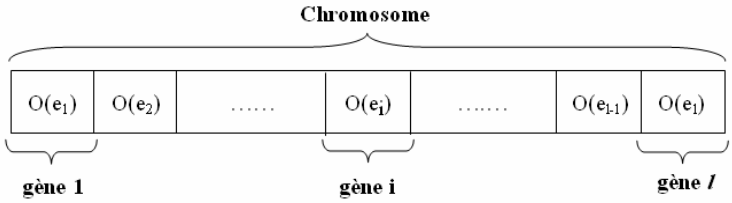


Figure III.22. *Codage des individus sous OEEA.*

Où : O (e_i) est le nombre d'occurrences de l'élément e_i dans la donnée,
l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Remarque :

1) $\sum_{i=1}^{l} O_i = n$ Où n ∈ N représente la taille de la donnée en termes d'éléments (caractères ou pixels).

2) Les éléments qui ne figurent pas dans la donnée auront une occurrence nulle.

Dans le cas de manipulation d'une donnée texte, l'opération de codage consiste à transformer le message en code particulier en calculant le nombre d'occurrence dans le message des 1393 caractères admissibles et l'attribuer à la case qui correspond à son rang dans une table, désignée par TCAR, regroupant les 1393 affichables en Unicode.

Le codage adopté dans ce cas, sera une instanciation du codage général présenté par la figure III.22. C'est celui illustré par le synoptique de la figure III.23.

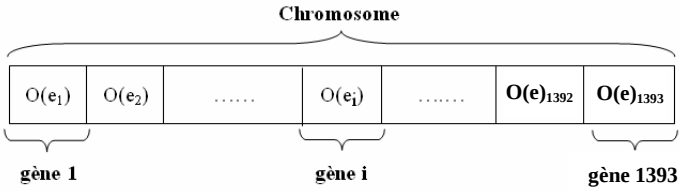


Figure III.23. *Codage des données texte sous OEEA.*

Où $i = \overline{1-1393}$ est le rang des caractères dans la table TCAR et $O(e_i)$ le nombre d'occurrences dans le massage du caractère ayant comme rang i.

Dans le cas de manipulation d'une donnée image et vu que les composantes R_i, G_i et B_i / $i = \overline{1,n}$ prennent leurs valeurs dans l'intervalle [0, 255], notre codage consiste à compter parmi les valeurs de chacune des composantes R, G et B, les nombres d'occurrences relatifs aux valeurs de l'intervalle [0, 255].

Il s'agit de compter à partir des valeurs des n éléments de R, les nombres d'occurrences des valeurs de l'intervalle [0, 255], et de même pour les valeurs des éléments de G et de B. La structure résultante dans ce cas, représente le codage adopté par notre algorithme développé. C'est celle résumée par la représentation chromosomale présentée à travers le synoptique de la figure III.24.

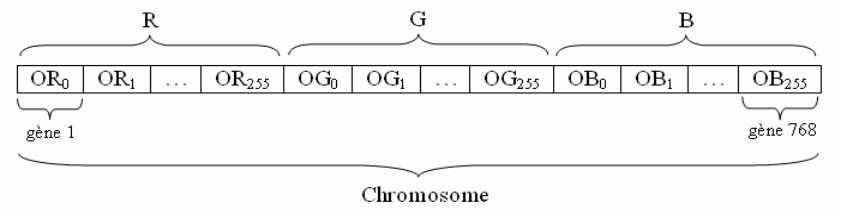


Figure III.24. *Codage des données images sous OEEA.*

Où : OR_i est le nombre d'occurrence des valeurs de la matrice composante R qui égale à *i*,
OG_i est le nombre d'occurrence des valeurs de la matrice composante G qui égale à *i*,
OB_i est le nombre d'occurrence des valeurs de la matrice composante B qui égale à *i*.

Si le codage binaire semble tout à fait adapté pour certains problèmes d'optimisation, cette représentation semble peu appropriée dans notre cas car elle est non intuitive. La représentation qui nous semble la plus naturelle est le codage entier, puisque toutes les opérations de l'algorithme vont manipuler des nombres d'occurrences qui sont des nombres entiers.
Dans ce qui suit, on désigne par individu la donnée à chiffrer ou toute autre donnée, et par chromosome tout individu ayant subi une telle opération de codage.

b. Création de la population initiale

Les m individus (I_1, I_2,…, I_m) formant notre population initiale sont obtenus par application de m perturbations aléatoires sur le chromosome initial I_0 représentant le codage de la donnée originale.

c. Reproduction

Les opérateurs génétiques servent à diversifier la population gérée par l'AE afin d'explorer le plus efficacement possible l'espace de recherche. Nous avons vu dans le chapitre 2 que les AEs disposent généralement de deux opérateurs : l'opérateur de croisement et l'opérateur de mutation.

- **Le croisement :**

Le croisement de solutions peut être toute opération permettant d'obtenir une nouvelle solution à partir de plusieurs individus existants.

Dans le but de compliquer la tâche des cryptanalystes tout en maintenant raisonnable le temps de calcul global de l'algorithme et en évitant toute convergence prématurée, nous avons utilisé un croisement à deux points. C'est le OX « Order Cross-over » proposé par Davis [Davi, 1985]. Les deux points de croisement sont choisis aléatoirement et l'opérateur est appliqué avec un taux qui varie entre 60% et 100% [Gren, 1986].

- **La mutation :**

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

Les individus résultants de ces deux opérations (croisement et mutation) seront rajoutés à la population des parents pour les acheminer vers l'étape suivante qui est celle d'évaluation.

d. Evaluation des individus

La fonction d'évaluation que nous avons définie pour associer des valeurs d'adaptation à chaque individu I_i d'une population Pop, est la suivante :

$$F(I_i) = \sum_{j=1}^{l} \left| O(e_j)_i - O(e_j)_0 \right| \qquad \text{(III.21)}$$

Avec : $I_i = [O(e_1), O(e_2),\ldots, O(e_l)]$, $Pop = \{I_1, I_2,\ldots, I_m\}$ où m est un paramètre de réglage et $O(e_j)_i$ représente le nombre d'occurrences du $j^{ème}$ élément dans le $i^{ème}$ individu.

Dans le cas de manipulation d'une donnée texte, F peut être instancié comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O(e_j)_i - O(e_j)_0 \right| \quad \text{(III.22)}$$

Ou tout simplement :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.23)}$$

Où O_{j_i} est le j^{ème} gène du i^{ème} individu.

Dans le cas de manipulation d'une donnée image, F sera instancié comme suit :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{j_i} - R_{j_0} \right| + \sum_{j=1}^{256} \left| G_{j_i} - G_{j_0} \right| + \sum_{j=1}^{256} \left| B_{j_i} - B_{j_0} \right| \quad \text{(III.24)}$$

Cette fonction est équivalente à celle donnée ci-dessous.

$$F(I_i) = \sum_{j=1}^{768} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.25)}$$

e. Sélection des individus les plus adaptés

La stratégie de sélection de l'algorithme développé va décider de la survie d'une donnée dans la population. Nous avons utilisé la méthode de la sélection par roulette.

f. Critère d'arrêt

Les étapes de reproduction, d'évaluation et de sélection sont répétées jusqu'à ce que l'optimum recherché soit trouvé. Ce dernier représente l'individu ayant la plus grande valeur de convergence durant tout le processus évolutionnaire. Donc, c'est la donnée la plus différente de la donnée originale.

Dans le cas d'une donnée texte, la condition d'arrêt assurant la convergence de notre algorithme évolutionnaire est exprimée à l'aide de la fonction F comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.26)}$$

Preuve :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} (O_{j_i} + O_{j_0}) \quad \text{(III.27)}$$

$$\leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.28)}$$

Et comme :

$$\forall I_i, I_k \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_k} \quad \text{(III.29)}$$

Et :

$$\forall I_i \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.30)}$$

Donc :

$$(III.27) \Rightarrow \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.31)}$$

$$\leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.32)}$$

Dans le cas d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right|$$

Et comme : $((0 \leq R_{ji} \leq n) \,\&\, (0 \leq R_{j0} \leq n)) \Rightarrow (0 \leq \left| R_{ji} - R_{j0} \right| \leq n)$ (III.33)

$((0 \leq V_{ji} \leq n) \,\&\, (0 \leq V_{j0} \leq n)) \Rightarrow (0 \leq \left| V_{ji} - V_{j0} \right| \leq n)$ (III.34)

$((0 \leq B_{ji} \leq n) \,\&\, (0 \leq B_{j0} \leq n)) \Rightarrow (0 \leq \left| B_{ji} - B_{j0} \right| \leq n)$ (III.35)

Alors, d'après (III.33), (III.34) et (III.35) nous aurons :

$$0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq (256 \times n) + (256 \times n) + (256 \times n) \quad \text{(III.36)}$$

$$\text{(III.36)} \Rightarrow 0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n \quad \text{(III.37)}$$

$$\Leftrightarrow 0 \leq F(I_i) \leq 768 \times n \quad \text{(III.38)}$$

Donc, les inéquations (III.26) et (III.38) représentent les conditions d'arrêt mettant fin à notre processus crypto-évolutionnaire dans le cas de manipulation de données texte ou images, respectivement. Nous constatons que $F(I_i)$ est une fonction bornée.

Après quelques tests d'exécution, nous avons constaté que le fait de prendre cette condition uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition d'arrêt ((III.26) ou (III.38)). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous avons arrivé à déterminer ce nombre :

MaxGen = 50

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes dans le cas de chiffrement de texte :

1) $F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i}$

2) MaxGen = 50

Toutefois, elle englobe les deux conditions suivantes dans le cas de chiffrement d'images :

1) $F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n$

2) MaxGen = 50

g. Déchiffrement

Dans cette phase représentant l'opération inverse du chiffrement, une clé de session est générée suivant le même mécanisme utilisé pour nos deux précédents algorithmes. Toutefois, elle est de taille fixe vu que toutes les données texte ont une représentation chromosomale fixe englobant 1393 gènes et toutes les données images ont une représentation chromosomale fixe regroupant 768 gènes.

h. Réglage des paramètres et résultats

Afin de construire un jugement objectif à propos de l'algorithme proposé et de pouvoir noter le maximum de remarques et de comprendre les détails, une large panoplie d'images tests de différentes dimensions a été utilisée. Nous présentons ci-dessous et à titre d'exemple les résultats obtenus pour les mêmes données test précédemment utilisées dans la présentation de nos deux algorithmes développés et précédemment présentés PosESecL1 et PosESecL2 : "Texte 1" de taille 1084 caractères, "Texte 2" de taille 1151 caractères, "Lena" de taille 131X131 pixels et "Logo" de taille 420X395 pixels.

- **Réglage de paramètres**

L'algorithme décrit ci-dessus possède différents paramètres, dont les valeurs influent fortement sur sa qualité. Cette section présente une étude empirique du réglage de ces paramètres afin d'obtenir de bonnes performances, ainsi que le comportement général de l'algorithme qui en résulte.

Les figures III.25 et III.26 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation.

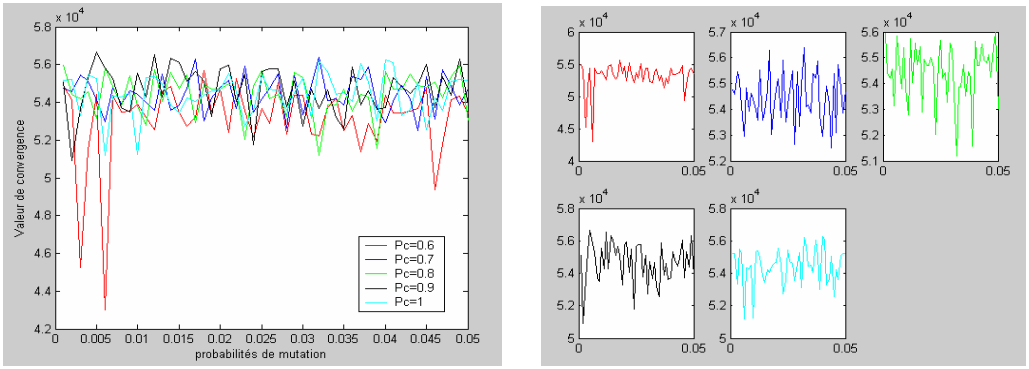


Figure III.25. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

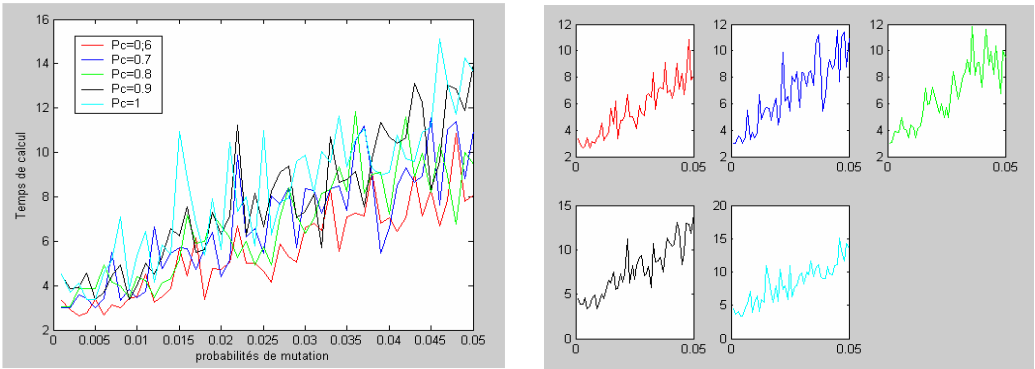


Figure III.26. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Pour choisir les valeurs des paramètres de la probabilité de croisement et celui de la probabilité de mutation, le niveau de confusion n'est pas la seule exigence à satisfaire ; le temps de calcul est aussi important. Ainsi, les valeurs qui nous semblent les plus adaptées sont : *Pc=0.9* et *Pm=0.005* assurant un haut niveau de confusion (VC=56674) en un temps de calcul très raisonnable (T=3.37s).

Après avoir fixé les paramètres de la probabilité de croisement et de mutation, la taille de la population est un autre paramètre à régler expérimentalement aussi. Les valeurs testées sont résumées à travers les figures III.27 et III.28 en indiquant pour chacune d'elles les résultats de chiffrement obtenus en terme de degré de confusion (représenté par la valeur de convergence) et de temps de réponse.

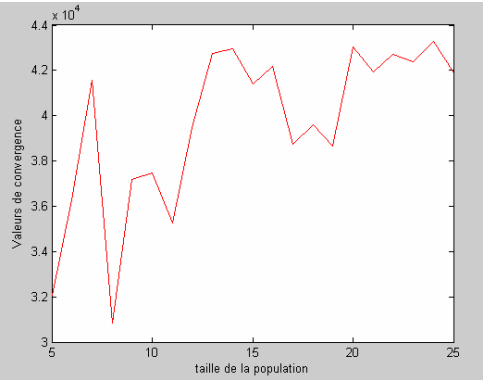


Figure III.27. *Evolution des valeurs de convergence en fonction de la taille de population.*

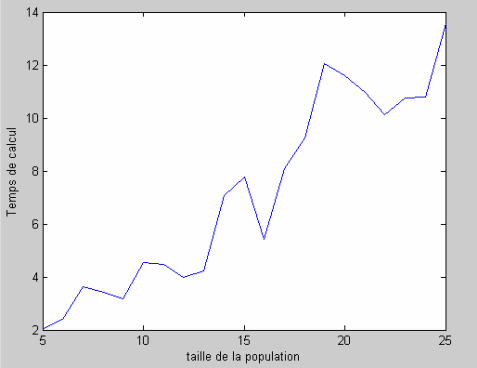


Figure III.28. *Evolution du temps de réponse en fonction de la taille de population.*

D'après la variation de l'influence de la taille de population sur la valeur de convergence et le temps de calcul résumé à travers les deux figures III.27 et III.28, nous optons pour régler le paramètre relatif à la taille de population en lui attribuant la valeur 13 (VC = 42739, T = 4.24s).

Le même principe a été utilisé pour fixer le paramètre relatif au nombre de générations. Donc, et suite à une série de tests sur un ensemble de données soumises au chiffrement et en utilisant la fonction fitness proposée, nous avons retenu certaines valeurs de paramètres qui permettent d'avoir un résultat exploitable en un temps de calcul très raisonnable (de l'ordre de 5 secondes). Ces valeurs sont données par le tableau III.5.

	Valeurs des paramètres de OEEA
Taille de la population	13
Nombre de générations	50
P_c	0.9
P_m	0.005

Tableau III.5. *Valeurs adoptées pour les paramètres d'OEEA.*

- **Résultats**

Après avoir décrit en détail notre méthode nous rapportons, dans cette section, les résultats de l'application de notre troisième algorithme proposé, OEEA, sur les données test citées précédemment (Texte 1, Texte 2, Lena et Logo) en adoptant le paramétrage fixé ci-dessus.

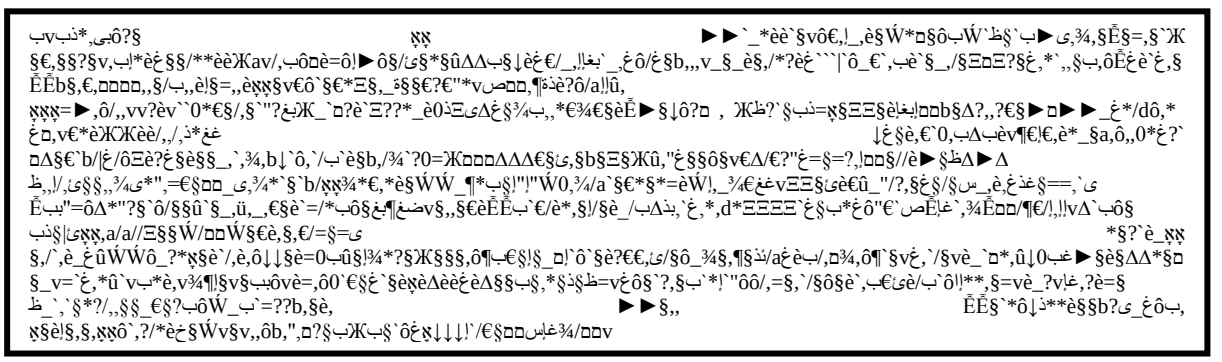
Nous constatons que la taille des clés est indépendante de la taille des données à chiffrer et elle est fixe pour des données de même nature (15323 bits pour les données texte et 7680 pour les données images).

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	15323	2173	6.54
	Texte 2	1151	15323	2284	6.73
Données images	Lena	131 X 131	7680	46034	3.37
	Logo	420 X 395	7680	431340	3.85

Tableau III.6. *Résultats obtenus par OEEA.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)

Figure III.29. ***(a)*** *Donnée originale Texte1,* ***(b)*** *Donnée chiffrée correspondante.*

Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

78 72 75 95 104 96 110 91 108 163 135 87 160 144 129 116 126 83 142
123 102 157 114 82 156 101 164 36 141 103 81 137 151 107 7 111 9 86
100 140 145 153 84 128 133 162 1053 1054 1055 1056 1057 136 120 1089 1090 1091 1092
1093 124 105 80 12 94 134 159 147 127 139 150 155 98 161 90 119 106 154
125 85 152 92 143 148 130 99 132 1281 1282 1283 1284 97 149 131 109 113 93
354 570 347 922 437 923 243 662 465 3 849 213 206 910 924 832 823 456 583
480 925 758 340 568 926 551 369 927 498 791 777 594 638 704 8 394 505 5
789 432 840 690 739 350 813 449 928 796 911 357 525 282 782 604 541 929 810
487 59 698 366 812 462 453 731 403 596 799 930 931 932 225 281 569 933 648
934 820 1386 1387 935 500 476 682 683 921 497 936 261 768 649 904 937 639 447
622 177 938 769 727 496 900 412 200 264 410 914 563 746 901 783 318 642 625
436 778 284 451 459 328 833 43 185 827 186 319 939 592 441 847 502 220 517
455 218 839 603 556 609 286 772 51 850 940 906 664 316 368 209 786 629 589
352 941 878 23 657 273 942 515 869 183 331 790 346 279 861 247 593 943 445
400 558 30 418 344 856 256 595 795 676 624 187 471 944 807 170 757 892 566
64 945 265 779 879 27 269 637 315 168 946 655 607 644 853 742 857 666 564
417 477 947 272 781 305 872 495 806 522 19 580 587 514 640 948 949 876 735
228 920 528 588 950 240 211 1231 1232 1233 915 458 33 287 312 951 545 863 860
571 667 490 952 953 954 888 916 913 299 391 905 671 378 442 669 575 591 726
811 865 202 398 358 399 955 956 314 454 172 341 234 621 546 214 643 737 230
957 223 788 891 259 577 736 958 416 181 263 755 486 50 289 618 959 292 846
864 174 960 519 433 395 434 20 902 961 907 962 1389 1390 1391 1392 1393 275 39
359 964 965 173 966 687 653 337 473 756 700 842 463 329 895 345 818 539 236
787 601 221 793 599 333 919 773 899 504 69 300 701 802 581 58 483 297 967
464 201 721 635 309 255 4 1 800 808 590 605 334 968 548 815 324 969 970
37 203 508 195 835 443 402 870 831 882 65 190 971 250 392 972 307 406 22
401 973 801 387 974 501 237 540 610 222 822 450 798 843 470 765 561 301 439
975 608 482 976 260 351 467 660 885 555 977 552 320 409 267 348 868 373 189
251 205 734 559 14 978 229 431 24 375 659 628 837 612 466 825 979 227 452
619 841 257 178 509 805 980 311 656 871 204 981 623 407 193 1060 1061 1062 1063
754 364 985 385 986 761 520 530 198 288 29 41 485 280 987 1230 1231 1232 1233
1234 290 764 988 989 469 1337 1338 1339 1340 1341 524 507 646 239 627 991 884 526
858 852 681 661 317 1197 1198 1199 1200 1201 826 992 322 699 353 1127 1128 1129 1130
475 993 47 994 887 1140 1141 1142 1143 877 672 70 430 384 684 192 995 527 996
303 997 674 383 828 1049 1050 1051 1052 342 2 326 60 678 491 866 283 26 404
792 1293 1294 1295 650 422 31 886 338 549 68 48 277 484 531 207 585 536 49
1000 705 258 600 651 521 1001 1002 523 438 363 460 15 293 513 381 747 444 550
753 310 1213 1214 1215 1216 1217 1003 898 335 248 245 1004 912 1005 386 274 529 634
630 1006 216 785 503 1257 1258 1259 584 356 620 1007 547 745 271 743 304 617 729
217 285 855 397 330 1008 1009 1010 557 516 875 750 771 355 512 728 296 472 492
1011 361 1012 880 582 809 166 427 446 803 420 494 1013 1371 1372 1373 1378 1379 1380
1381 167 576 784 1014 474 688 733 759 567 468 829 343 1271 1272 670 636 543 626
867 370 598 327 308 606 380 28 658 1015 774 295 844 665 61 560 188 268 732
821 1035 1036 1037 1038 1039 1040 874 371 360 673 663 196 1016 692 184 597 298 1017
614 415 16 1374 1375 723 1376 1377 55 752 715 244 685 215 1018 199 253 848 362
419 212 52 171 616 613 1019 917 425 390 534 1020 918 889 738 897 565 232 270
481 1021 252 488 535 641 396 838 893 797 578 896 766 1022 249 424 1023 1376 1377
1378 1379 388 834 1024 372 224 633 242 794 276 1025 854 631 1026 336 393 775 1356
1357 1358 1359 1360 691 532 675 197 235 499 194 572 730 1027 776 54 414 654 883
862 573 1028 506 349 909 679 586 56 780 873 493 332 428 231 602 377 632 542
824 423 816 53 749 1029 562 554 717 763 11 696 408 894 210 1030 179 702 645
751 804 680 1031 323 718 448 819 457 175 489 325 851 238 1032 306 814 741 374
233 611 461 1033 313 510 668 376 1034 57 748 724 44 1041 1042 1043 1044 1045 695
1046 1277 1278 1279 421 246 294 740 1058 1059 710 1064 1065 1066 1067 1068 719 1069 1070
1071 1072 1073 1074 1075 1076 1077 1078 1079 706 1080 77 73 76 74 79 117 146 1081
1082 1083 1084 63 1085 1086 1087 1088 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104
1105 1106 1107 1108 712 1109 1110 1111 1112 1113 25 169 881 413 382 1114 1115 708 1116
1117 1118 1119 1120 1121 1122 1123 714 1124 1125 1126 615 770 1131 720 1132 1133 1134 1135
1136 1137 1138 1139 18 1144 1145 1146 365 208 440 429 1147 1148 1149 1150 1151 1152 1153
1154 1155 35 1156 1157 1158 1159 1160 1161 903 537 1162 1163 1164 1165 1166 1167 1168 1169
1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 709 1180 1181 982 983 405 984 845 226
1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 66 1202 1203 1204
1205 713 1206 1207 1208 1209 1210 1211 1212 115 138 118 112 122 1218 1219 1220 1221 1222
1223 17 1224 21 1225 1226 1227 1228 1229 998 999 6 836 817 180 1235 1236 1237 1238
1239 1240 1241 1242 711 703 1243 1244 707 722 1245 1246 1247 1248 1047 1048 367 1249 1250
10 62 697 1251 1252 1253 1254 1255 1256 254 389 176 339 191 1260 716 1261 1262 1263
1264 1265 13 1266 1267 1268 1269 34 182 574 1270 1273 1274 693 1275 1276 1280 1285 1286
1287 725 1288 1289 1290 1291 1292 1296 1297 744 278 219 1298 1299 1300 1301 1302 1303 1304
1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 694 1320 32 1321
1322 158 88 89 121 40 1323 1324 1325 1326 1327 1328 1329 45 1330 1331 1332 1333 1334
1335 1336 1342 1343 67 1344 1345 1346 1347 1348 1349 1350 1351 46 1352 1353 42 1354 1355
544 426 291 647 762 262 963 411 1361 1362 1363 1364 1365 990 302 38 579 479 435

1366 1367 1368 1369 1370 538 859 686 241 908 760 533 165 830 511 266 553 1382 1383
1384 1385 518 677 890 1388 689 652 478

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسيًا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال

(a)

gé¥¥§&◄¥c:¥~x┼Ч‌x;>ش╪♫♫⇑¦;>}&◄◄¦¥"<;g§§q5⇑ ¥¥;}¬;§¦<<<î{xcx;<¥¥§xקקt;:g¥¥;ى";<xx¥:§¥§}§;¦;¥¥x;¥<
³x۞۞⇑۞rgt¦¥xrxg◄t¬g¦x4và¦x¥x>§_⇑¦ƒîx¦;<ƒv*§r¥ىx§{¥x¥xqx;⇑ ↔↔îsxî§⇑;¥§<{Ч<j³:&r'קק¥╪xr:
ى۞q¬xƒ>♫¦¥<:q¸↔~®¥xé§g§۞۞;ش;g&x؟♫§xq;
۞¬{î<¥l§;c§¥xî¥§§x{xî۞0>¦◄хњхњхîàњ¦{¥x_:q§Ч§&r¥x¥étsЧЧ{&>¥x;;¦x╪ى;¥x&4§¥:&³³<"îx§ЧЧЧé§xq{¬<ى&{□¥à&x¦¥⇑⇑&>;<
¥î§§¥§ît¥¥rr§;î&;;>g{x¥x¥¥q;<x§;+ ┼┼┼>g§§{¥x;qs;îrxgxƒ¥⇑&x§¥{_î<³¦cx¥ƒ{x>;x۞x5¥⇑xrg;§x&¥¬xx: ◄x§ƒ³{xj¥¥q{î¥;¥j◄5&{g;;
њњq¥¥x&é;g;/>:;l¶‌x³xg¥î;¦;îxx۞۞q<¥:&&♫;¦p¦:/v¥xxxקp¥&§jx;x¥q◄◄¥>;{↔۞î~_::{;x_§хњ&{ﺔ¬¥gƒ¥éî>¥¥§§tl۞;¦x;x,x¶‌¥¦l:g{
qsq¥¥x<¦¶‌Ч;ºx¦>¦x;&q¥:§; ◄xàtgƒ{¦§¥⇑cq¥ggî³¦xî♫x<t{§>;{¥t¬q>¬╪g<§q:>®&_§;<ىr↔xx>¥¦¥Ч‌g<¥àtt~¥éxà5¥ƒ;_<<¥
♫¬xrîx¥§¥¥g~¥¦j¥{x¥↔↔¥¥g{¥§~§¥§¥¥¥⇑¥;;¥gƒ!lx;:¬;{gњњњx§╪╪xx&¦g↔↔{¥q۞x¥¥xx¥r{t{<®ggî;□۞}gx_ﺔ{_:x{t¦¶‌Чt>◄◄◄
4g¥x<<¥¥q;à¥⇑qg³x¥;¥ƒs<_và¥r§♫xàx;³ §>xƒƒ◄◄x۞tq¥{>îxt⇑x¥¥éxxs;_xىs;{>
קà'xq{ƒ<<קקx¶‌xrgsrv<xњњ4¥;x<x³sg;§ج¦x>¦t♫§§&g{<{;;¥qקx³gњњxƒ¥;ىxéj¥s۞њ§§¥¥¥¥קxq_x;¥>¦<§î¥qstc5;§j³³¥s§>;:_ﺔé~g¥¥<x:4<
۞۞{¥tsî~g{§¥<ts§xxrq¥<¥{;:¥¥>&¥;x¦g♫:§;:®¥gsx;¥<v¦s/¬}>ى¸x⇑⇑s~§x;w{◄x:ىº¦îg§jx_Ч‌Ч&;g_؟>§{&◄۞¥t¦5;§®{{♫⇑¥ش§¥¥>¦x
<۞¥x<x;0>¦<>xt;¥î♫{x¥⇑îjs<¦;↔x¦¥><;&xrקקקs¥¬g&<x⇑:x{î³³x:îЧЧ‌_'{g§◄r§¥;<x/¥<⇑⇑{;¥{{{:□~¥x۞

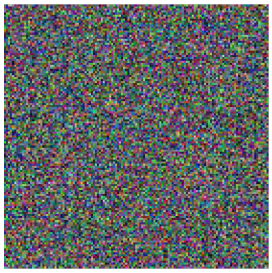
(b)

Figure III.30. **(*a*)** *Donnée originale Texte2,* **(*b*)** *Donnée chiffrée correspondante.*

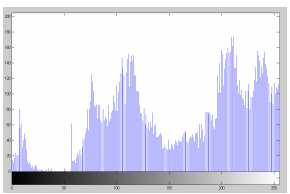
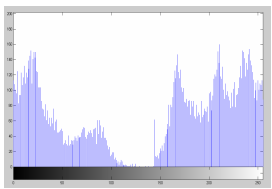
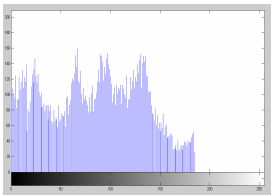
Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

59 1091 1 96 52 8 21 1200 47 54 71 16 43 14 1366 1369 25 64 26
1271 30 32 89 15 65 68 53 28 1303 1305 42 84 81 78 1090 34 24 62
20 58 74 56 39 1117 4 85 99 1324 18 98 1306 29 72 55 66 5 45
49 17 40 23 73 80 97 50 608 102 848 575 286 1220 1223 167 980 869 803
925 1201 134 310 1278 622 866 190 472 727 1118 535 818 1367 114 377 326 835 553
1368 166 488 100 1270 874 206 981 307 232 873 880 676 538 712 578 982 305 860
1325 899 933 842 332 865 315 713 239 445 619 203 939 983 785 192 1302 163 749
872 1327 297 605 513 722 270 940 269 814 198 943 201 570 195 422 746 897 802
402 904 945 250 794 560 691 1276 217 693 514 1304 852 935 804 666 518 1222 133
467 915 156 910 407 1221 170 257 738 542 692 912 984 508 443 245 815 671 372
63 61 67 985 1277 174 128 477 930 639 986 800 987 667 258 988 456 309 989
817 856 688 607 559 520 459 577 188 439 369 325 643 314 172 363 796 592 312
105 193 124 157 148 582 669 822 566 390 779 806 397 1187 1188 573 849 901 917
813 149 317 285 696 180 337 725 824 634 1057 223 990 829 554 953 466 732 242
733 905 647 308 168 101 487 958 1326 793 572 812 454 517 449 991 831 366 540
704 868 356 212 913 847 321 408 1059 280 623 909 484 271 137 662 220 433 324
881 1375 841 1333 896 837 624 992 891 1334 649 724 581 161 502 1275 545 685 924
843 1058 473 585 1374 828 833 275 993 382 450 176 107 1285 375 427 291 929 795
299 362 599 1283 863 178 736 515 531 937 331 292 994 742 888 1332 313 596 126
381 648 359 411 489 431 376 234 120 934 1284 175 714 522 955 534 247 614 598
652 1378 995 143 410 601 996 997 1273 453 883 158 208 719 298 430 689 446 462
357 104 840 928 184 413 303 657 364 399 870 256 396 636 424 159 391 1384 1385
115 1319 702 277 279 419 241 329 1000 734 698 627 469 871 521 690 602 262 140
342 687 1376 1001 464 887 350 116 268 1272 1274 895 720 465 267 197 1002 588 541
468 361 808 132 204 141 1377 287 444 547 438 150 761 300 221 425 921 1320 348
403 737 611 496 304 862 171 586 709 792 820 834 127 406 274 229 145 194 1003
886 546 392 922 288 673 731 448 205 276 561 263 1064 956 625 1004 186 227 492
1005 442 254 593 1006 1007 395 584 260 902 1065 530 968 590 334 389 1008 558 708
877 393 412 475 678 646 526 1203 1204 273 533 832 470 243 1009 579 884 474 451
786 850 908 723 1066 405 501 1010 1011 117 677 1012 498 658 1382 699 765 819 1013
214 1014 1015 233 423 135 1016 650 164 504 615 721 222 457 900 892 911 455 954
1017 642 1018 351 151 838 138 500 1381 505 1019 437 458 226 436 385 580 946 1020
179 131 400 653 173 1101 306 481 213 218 240 686 707 106 626 380 1021 511 358
333 146 1063 620 674 610 111 811 567 1022 185 507 603 878 338 706 1023 283 482
1100 266 379 323 799 926 432 1098 906 893 951 845 1024 388 668 568 718 855 764
110 701 949 224 112 864 756 551 125 920 680 1206 209 142 409 1025 207 612 844
670 633 587 367 645 302 130 591 1312 805 383 621 1026 169 679 103 349 1027 660
655 781 571 544 861 641 730 480 1028 322 1029 628 705 255 682 493 1030 1031 162

890 211 919 1099 478 927 1032 661 355 1205 768 932 1033 853 1067 236 1034 340 681
556 527 801 537 486 182 327 613 189 606 604 703 265 384 1035 617 231 1193 246
165 942 1036 728 640 821 875 200 597 659 261 717 441 415 854 1037 108 1192 569
823 374 281 894 539 923 807 452 202 154 1359 483 440 177 882 485 248 562 319
495 160 387 695 259 697 122 876 401 1190 651 576 414 879 1038 230 966 635 426
519 916 600 950 210 352 118 751 237 918 563 386 963 282 630 656 754 365 797
1182 1183 609 716 295 740 339 476 370 1039 330 378 675 1055 228 885 557 851 113
715 272 594 741 747 826 238 858 672 574 109 497 1191 447 152 353 121 663 494
760 532 499 129 898 404 1040 583 1184 524 565 638 344 294 416 253 644 629 119
938 284 429 1056 320 360 479 555 798 346 825 183 710 947 1041 936 1080 318 595
249 632 827 199 743 1180 336 637 683 435 421 191 729 684 944 810 460 512 509
503 857 1042 418 1094 739 654 1043 1044 525 529 839 1093 809 510 1181 664 771 244
394 311 434 859 1095 931 368 589 316 123 941 957 948 181 139 354 1045 1046 776
784 1047 1048 1049 1050 830 398 264 335 1051 1052 775 1053 1054 420 543 75 76 1060
979 1061 1062 1068 1069 1070 1071 1072 762 1073 1074 196 506 1075 1076 1077 969 1078 1079
11 9 1123 1124 1081 1082 293 373 1083 1084 973 965 1085 1086 296 564 1087 1088 1089
83 36 90 77 6 92 3 91 10 95 33 1092 371 219 1096 1097 726 251 1102
252 903 1103 1104 1105 1106 1107 147 889 552 1108 1109 774 1110 1111 1112 790 1113 773
1114 735 289 1115 1116 37 69 1119 1120 1121 1122 1125 788 975 1126 1127 461 780 1128
960 1129 1130 341 215 1131 1132 1133 971 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143
1144 1145 1146 976 1147 1148 1149 1150 769 1392 1393 1151 1152 1153 962 1154 1155 1156 225
694 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 216 914 153 1347
301 1171 1172 1173 1174 1175 752 1176 1177 1311 290 836 1178 1179 1185 1186 490 471 550
1189 1353 1194 766 1195 782 1196 1313 1197 1198 1199 665 428 1202 907 616 763 1207 977
1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 767 379 321 1224 758 1225 1226
1227 1228 1229 523 278 1230 1234 1235 770 1236 1237 1238 1239 748 1240 1241 1242 22 31
44 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1310 1254 1255 967 1256 1257 1258
1259 1260 753 964 1261 1262 816 1265 1266 618 516 1267 1268 1269 789 1279 1280 13 12
48 41 1281 1282 783 463 1286 1287 1288 1289 1290 959 1291 1292 1293 1294 70 7 1295
1296 1297 548 187 1298 1299 1300 1301 136 536 1307 1308 1309 970 744 1314 1315 757 777
1316 1317 1318 791 1321 1322 1323 27 35 1328 1329 144 345 952 1330 1331 961 1335 1336
1337 51 79 1338 1339 1340 711 235 1341 1342 1343 974 1344 745 1345 1263 1346 94 86
1348 1349 1350 867 155 1351 1352 528 750 1354 998 328 999 1355 1356 60 88 1357 82
46 38 87 57 1358 772 1360 1361 1362 343 549 491 417 1363 1364 1365 1370 1371 767
1372 1373 347 700 1379 1380 778 631 1383 759 846 1264 1386 1387 93 2 1388 1389 755
1390 978 787 972 1391

(a) ***(b)***

(c)

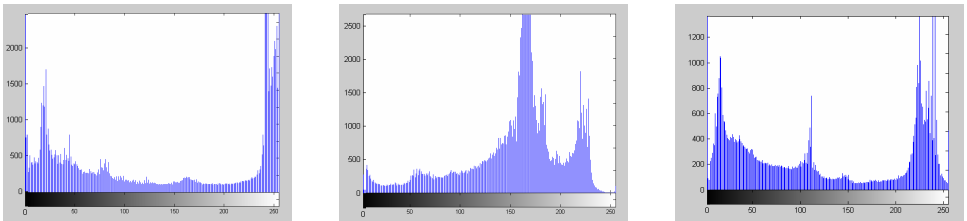
Figure III.31. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 0,9375 K octets) :

369 423 529 171 397 412 474 15 689 64 112 390 267 21 118 304 6 28 574
465 356 338 717 638 339 110 243 458 508 450 361 608 635 115 593 749 151 742
575 716 428 109 486 584 294 722 521 141 87 695 373 225 452 602 679 13 208
85 468 634 709 298 70 297 514 601 439 359 320 251 222 292 319 765 456 614
142 645 625 288 383 217 125 476 718 274 236 729 483 567 114 487 492 303 542

426	628	675	713	296	505	370	730	409	395	764	641	396	669	460	337	264	230	204
623	358	293	418	150	520	134	328	469	762	440	533	651	551	535	286	438	462	103
355	347	63	158	16	561	205	410	643	703	432	519	693	247	620	346	40	736	366
129	652	164	327	371	506	363	154	241	447	434	624	585	163	748	376	380	144	407
59	143	667	252	10	536	548	766	360	335	455	311	497	200	466	249	751	35	119
165	341	36	741	398	448	656	18	65	233	734	17	531	512	595	660	104	586	81
435	299	101	92	309	454	436	710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579	352	739	698	306	253	577	248
735	515	342	285	61	219	138	568	705	658	427	511	532	210	357	307	153	633	711
98	433	619	691	60	20	137	733	485	324	659	419	719	684	269	392	754	68	259
284	467	295	48	300	224	541	600	715	630	613	172	66	425	558	604	38	692	34
557	263	258	384	51	317	375	111	325	146	701	472	388	607	289	94	644	377	745
580	500	598	496	240	443	594	364	554	74	637	187	712	655	747	576	145	47	501
417	490	622	162	491	626	578	699	750	14	128	493	282	226	459	351	589	209	582
545	740	596	147	275	121	362	470	53	193	291	333	290	179	523	527	42	30	402
738	525	113	374	664	646	507	609	685	756	189	700	148	260	654	464	191	350	378
176	257	538	599	603	11	702	372	571	237	12	96	170	192	653	88	416	540	25
180	522	642	166	72	246	194	707	680	33	639	159	215	75	728	227	516	681	344
90	99	559	127	513	265	7	23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555	69	708	345	494	271	238	726
393	534	132	385	587	686	647	368	203	3	481	482	332	199	167	687	560	539	495
743	44	583	91	610	690	336	31	261	403	379	255	725	498	323	386	198	572	220
484	73	322	677	606	340	272	343	58	503	381	26	421	517	761	79	676	334	348
760	49	663	235	597	566	313	627	39	414	618	518	479	55	752	186	694	563	649
175	223	133	549	420	277	510	727	305	391	228	632	404	196	206	84	52	184	755
212	581	763	429	107	697	724	590	77	737	445	71	682	502	509	120	254	413	758
640	149	616	43	195	242	239	211	611	453	757	218	86	19	310	387	2	105	117
424	244	504	116	449	591	8	229	605	399	629	182	57	461	314	588	177	234	190
102	106	714	122	617	612	216	673	312	431	326	621	80	382	721	93	45	156	665
553	767	475	174	329	394	415	547	188	331	543	365	308	46	67	670	330	157	354
302	389	480	367	232	768	27	207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631	41	471	131	201	245	54	202
444	556	544	720	270	32	688	759	442	279	478	136	197	123	273	671	437	262	457
672	268	349	95	50	488	706	546	183	221	178	160	528	696	62	9	214	615	573
489	430	97	744	22	753	524	446	213	746	731	181	231	283	661	570	401	276	78
400	4	562	526	683	301	89	477											

(a) ***(b)***



(c)

Figure III.32. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session (Taille$_{Clé}$ = 0,9375 K octets) :

503	463	515	233	412	71	181	502	153	365	302	449	225	362	500	33	523	198	519
471	404	521	497	145	496	504	494	205	138	505	6	522	214	149	451	422	498	146
520	63	453	320	516	510	350	287	495	39	387	385	20	115	80	361	472	69	513
213	204	105	425	68	311	507	506	208	508	24	41	499	511	237	54	415	509	37
518	517	52	458	384	493	390	8	191	78	501	40	163	514	229	512	99	574	372
579	351	396	36	389	17	370	312	626	436	5	1	284	118	629	334	594	625	174
704	648	28	534	469	657	342	411	479	401	45	533	234	675	379	23	235	231	223
77	324	729	634	548	217	555	142	673	159	346	568	551	85	195	343	333	408	158
10	563	532	373	264	73	242	359	179	280	473	709	605	128	327	82	756	308	638
588	152	31	766	272	285	166	667	714	51	613	157	439	265	561	271	314	277	491
286	32	125	2	332	194	547	552	241	156	443	661	143	455	431	120	215	484	578
434	526	647	344	257	371	744	168	199	584	260	707	486	763	369	244	524	298	540
689	405	764	196	595	538	345	478	636	464	95	216	29	567	46	735	57	331	397
752	426	377	288	133	114	134	62	424	668	720	399	200	336	696	459	141	553	173
702	169	148	171	460	622	210	35	409	201	570	209	444	14	624	558	299	483	66
366	184	224	600	100	671	738	293	677	549	192	193	291	554	230	76	481	639	67
124	545	620	226	93	529	303	136	164	470	603	150	687	419	304	683	414	581	74
565	127	245	144	577	337	760	475	557	705	301	49	418	137	44	110	7	79	306
207	13	635	380	276	758	420	278	248	238	292	630	457	47	437	583	445	375	723
354	492	684	255	679	81	263	119	221	61	682	751	669	652	104	681	659	290	92
121	454	188	172	564	355	754	708	489	251	394	220	12	60	97	126	566	604	403
601	15	395	429	330	715	645	546	490	759	108	154	423	34	441	87	741	575	643
666	101	19	693	686	718	111	631	326	542	617	674	706	84	178	543	381	50	167
733	402	716	654	462	614	239	300	254	725	262	619	232	428	305	593	335	161	427
83	250	341	180	374	632	465	406	719	183	378	623	382	243	147	740	765	296	649
658	736	582	252	165	712	745	89	413	569	160	106	633	329	177	608	340	461	589
537	261	456	38	103	607	640	627	98	4	30	597	416	281	596	599	536	253	734
26	742	477	642	266	438	135	386	573	295	339	721	711	190	487	753	591	539	162
737	610	726	313	140	197	70	16	525	621	637	86	749	182	572	11	474	139	612
730	347	598	43	556	541	319	48	761	187	664	748	275	646	672	258	151	219	710
259	256	236	348	450	609	109	325	186	571	580	678	703	435	430	694	606	376	274
240	328	206	393	175	466	615	315	587	75	383	602	616	527	25	273	762	448	697
113	452	132	155	724	364	739	294	485	728	202	750	476	650	544	116	480	688	42
338	530	352	123	743	247	641	676	368	211	176	107	467	698	322	55	560	56	731
685	270	170	767	227	655	690	279	310	746	644	203	732	680	663	662	417	64	528
651	433	691	96	59	246	585	757	112	318	283	618	628	22	297	665	592	535	94
65	58	222	102	586	699	3	695	358	228	717	212	392	576	531	289	130	88	356
388	316	122	768	660	421	446	185	611	268	670	410	398	468	249	321	755	72	9
18	692	700	117	323	722	447	307	353	360	131	90	407	129	363	562	432	488	590
27	367	400	559	349	317	391	440	656	653	727	218	442	482	282	713	53	747	91
269	309	189	267	21	701	550	357											

III.5. Discussion et évaluation des résultats

Un bon crypto-système doit satisfaire les six points clés de Kerckhoffs [Kerc, 1883] et ceux de Shannon [Shan, 1949]. Plus explicitement, la qualité d'un crypto-système se mesure principalement par sa résistibilité face aux différents types d'attaques. La technique de cryptage proposée à travers nos trois algorithmes proposés PosESecL1, PosESecL2 et OEEA et qui consiste en une perturbation évolutionnaire de la donnée originale, assure une bonne sécurisation contre les attaques les plus destructives. Les critères utilisés pour évaluer leurs sécurités sont les suivants : la vitesse de l'algorithme, l'attaque statistique, l'attaque différentielle, l'attaque exhaustive et l'analyse de l'espace des clefs.

III.5.1 Vitesse des algorithmes

En plus du paramétrage, la vitesse d'un algorithme évolutionnaire dépend étroitement du codage utilisé. Un mauvais codage affecte non seulement la qualité de l'algorithme en termes d'optimalité de solutions mais aussi en termes de temps de convergence. Dans le cas de nos deux premiers algorithmes proposés, PosESecL1 et PosESecL2, le codage dépend de la taille de la donnée à chiffrer ce qui affectera la vitesse des algorithmes puisque le temps de traitement des données de grandes tailles sera beaucoup plus grand que celui de traitement des données de petites tailles, chose due à la taille des chromosomes manipulés durant les générations de l'évolution. Ceci se voit clairement à travers les résultats obtenus par ces deux algorithmes où le temps de convergence de PosESecL1 est meilleurs que celui de PosESecL2

du fait que la taille des chromosomes codant une donnée pour le PosESecL2 est trois fois plus grande que celle des chromosomes codant la même donnée pour le PosESecL1. Toutefois, le codage proposé à travers OEEA unifie la taille des chromosomes codant des données de même nature (textes ou images) et de différentes tailles. Ainsi, le temps de traitement est indépendant de la taille des données traitées. En plus et d'après les résultats expérimentaux présentés précédemment, nous constatons que le temps de convergence de ce dernier algorithme est très raisonnable (de l'ordre de 4 secondes).

Comparons, maintenant, la vitesse de nos trois algorithmes développés avec celle des principaux crypto-systèmes (AES du coté des crypto-systèmes symétrique, RSA du coté des crypto-systèmes asymétrique et SEC qui est un algorithme de chiffrement exploitant les algorithmes génétiques). Le tableau 7 et la figure 33 présentent un résumé des temps de chiffrement et de déchiffrement de chacun des algorithmes cités.

	PosESecL1	**PosESecL2**	**OEEA**	**AES**	**RSA**	**SEC**
Temps de chiffrement (s)	24.5	46.15	3.37	0.5	42.8	1.8
Temps de déchiffrement (s)	0.26	0.75	0.12	0.08	42.5	0.06

Tableau III.7. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

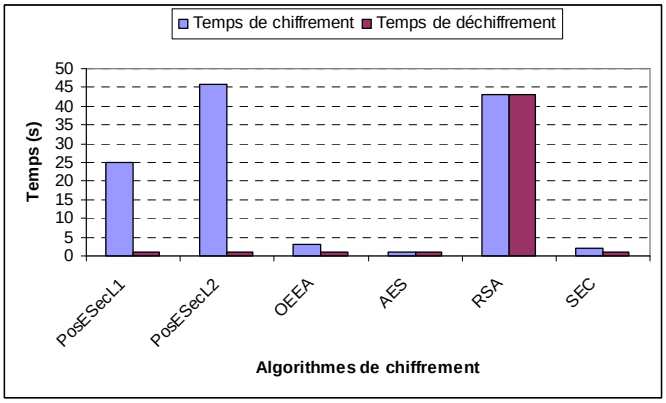


Figure III.33. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

D'après le tableau et la figure, ci-dessus, présentant le temps de chiffrement et celui de déchiffrement de nos algorithmes en comparaison des algorithmes SEC [Omar, 2007], RSA [Mene, 1996] et AES [www4], nous remarquons que :

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement beaucoup plus petit que celui de RSA. De son coté PosESecL1 possède aussi un temps de chiffrement/déchiffrement meilleur que celui de RSA. Toutefois, PosESecL2 a un temps de chiffrement de l'ordre de celui de RSA et un temps de déchiffrement meilleur que celui de RSA.
- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin de celui d'AES. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes PosESecL1 et PosESecL2 sont plus grands que celui d'AES.

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin que celui de SEC. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes sont plus grands que celui de SEC.

De même et d'après les résultats expérimentaux obtenus et présentés précédemment, nous constatons que le temps de calcul de OEEA est indépendant de la taille de la donnée à chiffrer du fait de l'unicité de taille de codage de toutes les données de même nature soumises au chiffrement. Toutefois, ce n'est pas le cas pour PosESecL1 et PosESecL2 où le temps de calcul des deux algorithmes est strictement dépendant de la taille de la donnée à chiffrer. Ainsi, OEEA est l'algorithme ayant un temps moyen de calcul le plus adapté.

III.5.2 Niveau de confusion

La confusion est l'une des notions essentielles énoncées par *Shannon* sur lesquelles se base un algorithme de cryptage. Elle sert à cacher la relation entre la donnée en clair (originale) et la donnée chiffrée correspondante. Donc, elle vise à rendre la donnée aussi peu lisible que possible.

Dans notre cas, les deux modes de chiffrement développés, à savoir le chiffrement à base de positions et le chiffrement à base d'occurrences, sont de niveaux de confusion différents.

Pour le premier, les deux algorithmes proposés dans ce mode sont aussi de niveaux de confusion différents. En effet, le niveau de confusion de PosESecL2 est meilleur que celui de PosESecL1 du fait que PosESecL2 exploite les positions des éléments codant les différents composants de la donnée, considéré chacun comme unité séparée (gène) du codage final de la donnée (chromosome). Ainsi, la donnée chiffrée aura de grande chance de contenir de nouveaux éléments ou composants ne figurant pas dans la donnée originale. Cette chose complique considérablement, voire même, pénalise toute cryptanalyse possible comme nous allons le démontrer dans les sections qui suivent. Toutefois, PosESecL1 est de niveau de confusion strictement dépendant de la taille de la donnée à chiffrer puisqu'il exploite les positions des éléments de la donnée. Donc dans le cas de données de petites ou de moyennes tailles, une attaque exhaustive finira de trouver la bonne combinaison initiale de positions et, ainsi de casser l'algorithme.

Pour le deuxième mode de chiffrement développé, il offre un très bon niveau de confusion grâce à l'utilisation d'un codage bâti autour de la notion de nombres d'occurrences. Ce mode ne présente ni information utile pouvant être exploitée par un cryptanalyse ni possibilité de reproduction de la donnée en se basant uniquement sur cette donnée surtout que le codage adopté assure que les composants de la donnée (caractères ou pixels) changent dans la donnée chiffrée par rapport à la donnée originale. En plus, le niveau de confusion de OEEA, l'algorithme développé dans ce mode, est meilleur que celui de PosESecL2 et ainsi de celui de PosESecL1 du fait que l'espace des nouveaux caractères qui peuvent apparaître dans une donnée chiffrée par rapport à la donnée originale correspondante est beaucoup plus grand dans OEEA que dans PosESecL2.

III.5.3 Attaque statistique

Ce type d'attaque considère le crypto-système comme une boite noire, il analyse statistiquement les entrées et les sorties de ce système. Pour ce faire, nous avons utilisé les mesures suivantes dans le but d'évaluer ou de quantifier la différence entre l'image originale et l'image cryptée correspondante :

Le facteur *NPCR* (*Number of Pixels Change Rate*) donné par l'expression (III.39), l'erreur absolue moyenne (*MAE : Mean Absolute Error*) donnée par l'expression (III.41) et l'erreur

quadratique moyenne (*MSE :Mean Square Error*) donnée par l'expression (III.42). Le tableau 8 résume les valeurs des différentes mesures obtenues après les tests qui ont été effectués sur l'image Lena et l'image chiffrée correspondante obtenue dans le cas de nos différents algorithmes proposés.

	NPCR	**MAE**	**MSE**
PosESecL1	0.6314	0.91	0.58
PosESecL2	0.9073	0.97	0.61
OEEA	0.9955	1.03	0.75

Tableau III.8. *Niveaux de confusion.*

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \tag{III.39}$$

$$D(i,j) = \begin{cases} 0 Si\ \mathrm{Im}_O(i,j) = \mathrm{Im}_C(i,j) \\ 1 Si\ \mathrm{Im}_O(i,j) \neq \mathrm{Im}_C(i,j) \end{cases} \tag{III.40}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \tag{III.41}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \tag{III.42}$$

D'après les résultats d'application des mesures quantifiant la différence entre l'image originale et ses versions chiffrées calculées par nos algorithmes, il est clair que la majorité des pixels formant l'image originale ont changé soit de positions (dans le cas d'application de PosESecL1) ou de contenu (dans le cas d'application de PosESecL2 ou de OEEA) pour, ainsi, calculer l'image chiffrée. En effet cela se voit surtout à travers les valeurs de NPCR qui calcul le niveau de différence entre les deux images où les résultats obtenus sont proches de la valeur optimale qui vaut un (01). Toutefois, ces valeurs ne sont pas les mêmes pour les trois algorithmes, chose qui est tout à fait normale vu que les algorithmes sont de niveau de confusion différent (voir section précédente). Et comme nos algorithmes présentent l'avantage d'être non déterministes, donc chaque nouvelle application d'un même algorithme donnera lieu à des résultats différents ce qui entraîne que l'application des mesures NPCR, MAE et MSE donnera lieu, de son tour, à de nouvelles valeurs. Ainsi, aucune propriété statistique ne peut être déduite d'une donnée chiffrée et l'attaque statistique ne possédera aucune chance de casser nos algorithmes.

III.5.4 Attaque différentielle

Cette attaque essaye de tirer des conclusions sur le fonctionnement d'un algorithme de chiffrement en comparant des versions chiffrées de plusieurs données originales et dans un meilleur cas des versions chiffrées de plusieurs blocs formant une même donnée. Ainsi, dans le cas des algorithmes de chiffrement par blocs (DES [Stin, 1996], [Biha, 1991], [Mats, 1994], 3DES [Stin, 1996], [Gant, 2001], AES [Lepr, 2000], etc), quand la donnée (surtout pour le cas d'images) contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement, ce qui fait que la donnée cryptée contiendra des zones texturées ; mais vu que nos algorithmes traitent la donnée en une seule passe, c'est-à-dire ils opèrent sur la donnée totale et non pas des blocs de la donnée ; alors la cryptanalyse

différentielle sera mise à l'écart. De ce fait aussi, nos algorithmes présenteront une certaine robustesse au bruit par opposition aux algorithmes de chiffrement par blocs, et qui se traduit par le fait qu'une erreur sur un bit chiffré ne va pas propager des erreurs importantes dans tout le bloc courant et par la suite dans toute la donnée lors de la recombinaison des blocs.

Même si le cryptanalyste pense à essayer de tirer des observations à partir des résultats de cryptage de plusieurs données pour essayer d'imiter le fonctionnement des algorithmes, l'approche évolutionnaire pseudo-aléatoire complique considérablement la tâche de ce type d'attaque. Une sensibilité à la donnée originale est aussi un point avantageux des algorithmes proposés vu que la donnée cryptée correspondante à une certaine donnée originale change d'une instanciation du problème à une autre. Autrement dit, nos algorithmes développés sont des algorithmes non déterministes à l'inverse du RSA, par exemple, qui a nécessité au préalable de randomiser les données soumises au chiffrement pour éviter les attaques exploitant les modèles connus des données en clairs [Ster, 2004].

III.5.5 Attaque exhaustive

L'attaque exhaustive ou encore dite attaque à force brute est une attaque qui demeure fatale pour les crypto-systèmes utilisant des clés de petites tailles. À ce stade, nos trois algorithmes proposés présentent des niveaux de sécurité différents. La robustesse de PosESecL1 est strictement dépendante de la taille de la donnée originale. PosESecL2 présente un niveau de sécurité plus élevé que celui de PosESecL1 du fait que sa clé soit de taille quatre fois plus élevée que celle de PosESecL1 dans le cas de manipulation de données textes et de trois fois plus grande dans le cas de manipulation des données images. Ainsi, et vu qu'à l'heure actuelle la taille de la clé assurant une résistibilité face à une attaque exhaustive est de 512 bits, alors le cryptage par PosESecL1 de toute donnée possédant une taille inférieure à 512/le nombre de bits nécessaires pour coder la taille de la donnée (nombre de caractères ou de pixels), sera exposée au risque d'une attaque réussite par force brute. C'est le cas aussi pour les données cryptées par PosESecL2 et qui sont de taille inférieure à : 4*(512/le nombre de bits nécessaires pour coder la taille de la donnée texte) ou de 3*(512/le nombre de bits nécessaires pour coder la taille de la donnée image). Toutefois, OEEA ne peut en aucun cas être cassé par une telle attaque vue que la taille de la clé utilisée est largement sécuritaire. Ainsi, OEEA est l'algorithme le plus sûr parmi nos trois algorithmes développés.

Considérons le texte à chiffrer « **there is only one God and Mohamed his prophet** », nous reportons dans le tableau suivant (tableau 9) les tailles de clés de chiffrement de nos trois algorithmes développés en comparaison de DES, 3DES, AES et SEC, ainsi que le nombre d'opérations nécessaires pour générer toutes les combinaisons de bits formant les clés pour chaque algorithme. Ceci donnera une idée sur la complexité de l'attaque exhaustive appliquée contre chacun des algorithmes où, il se voit clairement que l'attaque exhaustive menée contre nos algorithmes est la plus complexe.

	Taille de clé (bits)	**Nombre d'opérations**
DES	56	2^{56}
3DES	128	2^{128}
AES	256	2^{256}
SEC	240	2^{240}
PosESecL1	352	2^{352}
PosESecL2	1408	2^{1408}
OEEA	1393	2^{1393}

Tableau III.9. *Complexité de l'attaque exhaustive.*

III.5.6 Analyse de l'espace des clés

Ici, nous testons la sensibilité du processus cryptographique proposé aux clés utilisées. Dans notre cas, la clé générée est une clé calculée à partir de la donnée originale et de la donnée chiffrée correspondante, donc elle change d'une instanciation du problème à une autre. Ainsi, le cryptage successif d'une même donnée donne lieu à un ensemble de données chiffrées différentes ce qui entraînera, à chaque fois, la génération d'une clé de session différente pour le chiffrement d'une même donnée originale. Cela dotera notre méthode de cryptage d'une très grande sensibilité aux clés puisque toute clé interceptée d'une manière illégale ne servira que pour le déchiffrement d'une seule version chiffrée d'une même donnée donc elle ne sera plus utile par la suite.

Le problème rencontré dans le cas des crypto-systèmes symétriques, et par conséquent dans le cas de nos algorithmes développés qui sont des algorithmes symétriques, est la communication de la clé secrète au destinataire en vue de l'utiliser dans l'extraction de l'information originale (le déchiffrement). Pour ce faire, certains concepteurs ont proposé de combiner les fonctionnalités de la cryptographie symétrique et asymétrique. C'est le cas du PGP par exemple. Il crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ; puis, il crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire. Ainsi, le même principe peut être adopté pour sécuriser le transfert de nos clés en leurs chiffrant par un crypto-système asymétrique. Dans ce cas un schéma hybride de transmission sécurisée peut être envisagé comme le montre la figure suivante :

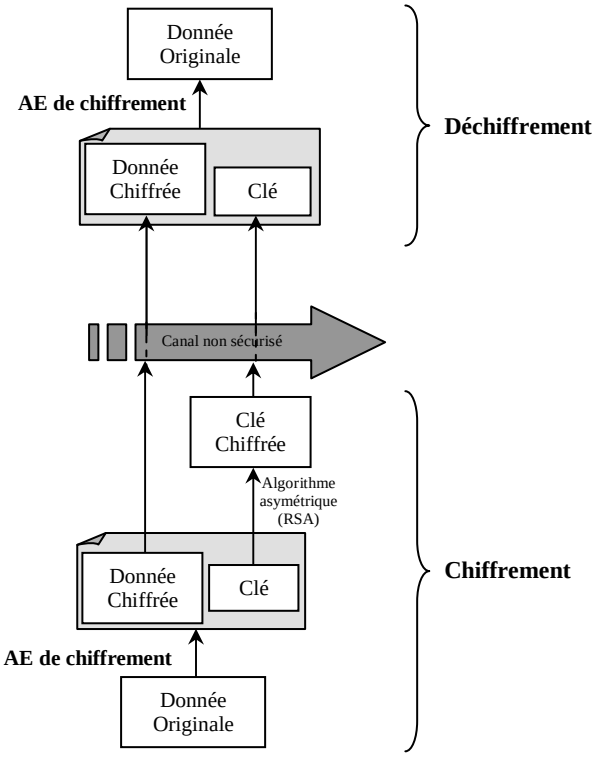


Figure III.34. *Schéma hybride de transmission sécurisée.*

Or, pour nos algorithmes PosESecL1 et PosESecL2, il est clair que la taille de la clé dépend strictement de la taille de la donnée. Le problème qui se pose lors du chiffrement d'une donnée de grande taille, c'est que la taille des clés générées elle aussi devient grande ce qui consommera un grand temps de chiffrement de clés du fait que les crypto-systèmes publiques sont déjà lents. Le problème s'aggrave de plus en plus lorsque la taille de la clé

devient égale ou dépasse la taille du codage de la donnée en termes de bits. Dans ce cas, l'utilisation de ces deux algorithmes devient inutile et il sera plus intéressant d'utiliser un crypto-système publique pour chiffrer la donnée au lieu de la clé générée malgré que la sécurité de tels crypto-systèmes soit grandement affectée par les choix paramétriques (p et q dans le cas de RSA [Zimm, 2005]). Pour remédier à cette anomalie, nous proposons de coder (ou bien comprimer) la clé générée par un système de codage sans perte, tel que le codage de Huffman par exemple [Huff, 1952], en vue de réduire sa taille avant de la chiffrée par un crypto-système asymétrique, puis d'envoyer au destinataire le package englobant la donnée chiffrée et la clé codée chiffrée. Lors de la réception, on procède inversement. On déchiffre la clé codée en utilisant la clé publique du destinataire, puis on la décode pour obtenir la clé proprement générée par PosESecL1 ou PosESecL2 qui va permettre de reproduire la donnée originale. Ainsi, le schéma du processus de sécurisation englobant cette dernière fonctionnalité sera le suivant :

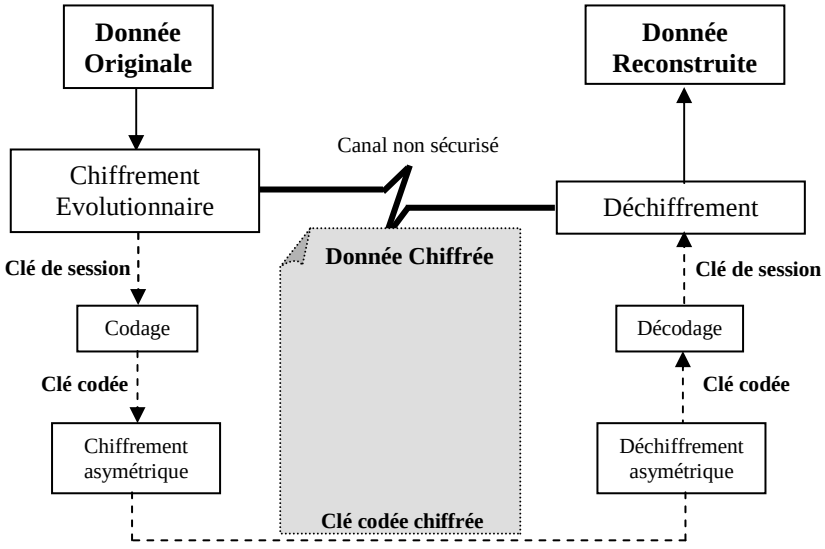


Figure III.35. *Schéma général du processus de chiffrement et de transmission sécurisée de la clé générée par chiffrement publique.*

Une deuxième solution de transmission sécurisée de la clé de session générée peut être envisagée dans le cas de manipulation de données images. Il s'agit de la technique de tatouage. Après avoir codé la clé générée par notre algorithme de la même manière proposée dans le cas de la première solution de transmission sécurisée de la clé (en utilisant un codage de Huffman par exemple), nous proposons, ici, de marquer l'image chiffrée par la clé de session codée. Ainsi, le schéma du processus de sécurisation englobant cette deuxième solution de transmission de la clé peut être résumée par le schéma de la figure III.36.

Cette deuxième solution de transmission est conditionnée par la taille de la clé codée pour satisfaire le compromis Robustesse-Capacité-Invisibilité lors du tatouage [Katz, 2000], [Barn, 2004], [Koba, 1990].

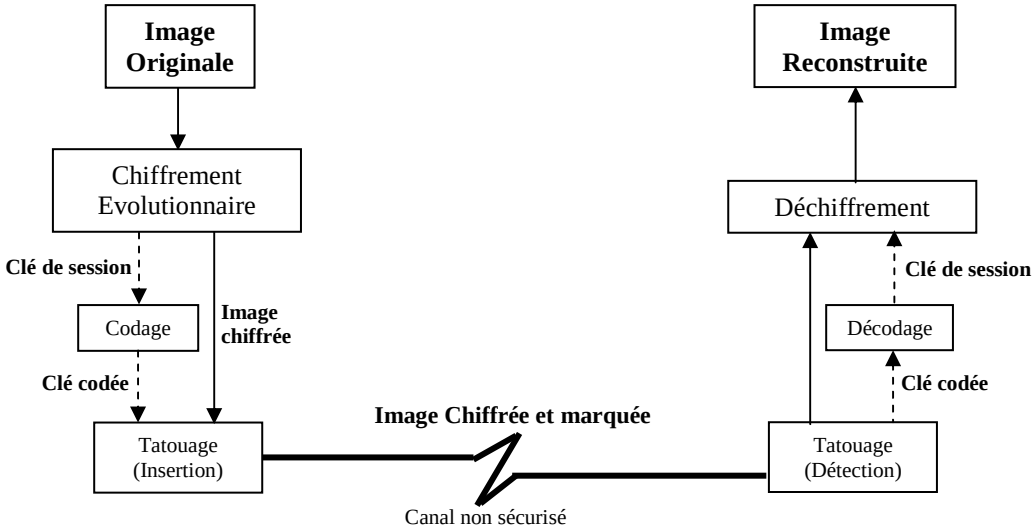


Figure III.36. *Schéma général du processus de chiffrement d'images et de transmission sécurisée de la clé générée par tatouage.*

III.6. Conclusion

À travers le présent travail, nous avons démontré que les algorithmes évolutionnaires, inspirés fondamentalement de la sélection naturelle des espèces, peuvent être également appliqués au cryptage de données. En effet, avec ces algorithmes, nous avons ramené le problème de chiffrement en un problème d'optimisation.

L'application de ces algorithmes requiert le réglage d'un certain nombre de paramètres pour la phase de chiffrement. La taille de la population initiale, le taux d'exploration et le nombre de générations affectent le temps de convergence des algorithmes, quant aux probabilités *Pc* , *Pm*, elles possèdent, théoriquement, une influence directe sur la qualité du résultat. Or, dans notre problème, les opérateurs génétiques (croisement et mutation) n'agissent pas sur l'information traitée (caractères du texte ou pixels de l'image) mais sur leurs coordonnées ; ils sont utilisés comme un outil pour explorer le reste de la population. De ce fait, les paramètres P_c , P_m affectent également la convergence des algorithmes.

Dans la pratique, les paramètres des algorithmes évolutionnaires sont réglés approximativement par tâtonnement [Gold, 1989]. Suite à une série de tests sur diverses données et en utilisant les fonctions fitness proposées, nous avons retenu certaines valeurs qui permettent d'avoir un résultat exploitable (donnée chiffrée) en un temps de calcul variant d'un algorithme à l'autre.

Les résultats obtenus montrent que les schémas proposés présentent des aptitudes dans la confusion et dans la sensibilité à la donnée originale qui les rend loin des attaques différentielles. De même, et pour pénaliser ou plus ou moins compliquer la tâche de l'attaque exhaustive, une deuxième alternative (PosESecL2) a été proposée en augmentant de trois ou de quatre fois la taille de la clé générée par la première alternative (PosESecL1). Une troisième alternative de cryptage robuste et sûre a été également définie. Il s'agit d'OEEA opérant suivant un mode de chiffrement (chiffrement à base d'occurrences) inexploitable par aucune attaque avancée.

Et pour résumer l'efficacité des deux modes de cryptage exploités, le cryptage à base de positions et le cryptage à base d'occurrences, le plus correcte est de dire que le deuxième

mode est plus important que le premier. En effet, ce mode est bâti autour de la notion d'occurrences, chose qui ne peut en aucun cas guider ou être exploitée par un cryptanalyste puisqu'elle ne présente aucune information utile pour aucune attaque possible. Ce qui n'est pas le cas pour le premier mode de cryptage à base de positions où une recherche exhaustive finira, un jour, par retrouver la bonne combinaison de positions d'éléments d'une donnée et ainsi de remettre en cause l'algorithme de cryptage utilisé.

Ainsi c'est les qualités du mode de chiffrement à base d'occurrences qui nous a motivées à le tester à travers d'autres métaheuristiques qu'elles soient à population ou à une seule solution. D'ailleurs le prochain chapitre présente une métaheuristique de colonie de fourmis traitant le problème de cryptage en opérant suivant ce mode de cryptage proposé.

Finalement, il sera utile de rappeler que l'objectif en cryptage est d'assurer la sécurité des données pendant leur durée de validité, donc, toute attaque tardive n'aura aucune importance.

CHAPITRE IV

CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

IV.1. Introduction

L'auto-organisation est un phénomène décrit dans plusieurs disciplines, notamment en biologie dans la branche qu'est l'éthologie (étude des comportements des espèces animales dans leurs environnements). Une définition a été proposée par Deneubourg [Dene, 1977] **:** *«l'auto-organisation est un processus dans lequel, un modèle de niveau global émerge uniquement, d'un grand nombre d'interactions entre les composants de bas niveau du système. De plus, les règles spécifiant les interactions entre ces composants sont suivies en utilisant uniquement des informations locales, sans références au modèle global ».*

Autrement dit, l'auto-organisation explique l'émergence d'un comportement collectif macroscopique par des interactions simples au niveau microscopique. L'auto-organisation n'exclut pas la complexité au niveau individuel. Elle suppose simplement qu'à un certain niveau, les individus se comportent comme des entités simples [Monm, 2000].

Depuis longtemps déjà, des comportements auto-organisés ont été découverts dans la nature. Par exemple, dans une colonie de fourmis, on peut observer différentes castes spécialisées dans un certain nombre de tâches : élevage de couvain, recherche de nourriture, construction de nid, etc. Aussi, le nid est construit sans que les insectes soient dirigés, ils répondent à un certain nombre de stimuli provenant de leur environnement.

D'importantes recherches ont eu pour intérêt principal l'étude de ces comportements intelligents afin de savoir comment ces populations interagissaient, accomplissaient des tâches et évoluaient. Ces recherches ont abouti à des modélisations inspirées du principe qu'un groupe d'entités plutôt simples et obéissant à des règles locales de coordination, sont capables d'engendrer des comportements globaux beaucoup plus complexes. Ces modélisations caractérisent ce que l'on dénomme « l'intelligence collective » [Bona, 1994], [Monm, 2000].

Dans ce chapitre, nous nous intéressons à l'algorithme de colonies de fourmis. Nous l'avons étudié, adapté et appliqué au problème de cryptage de données texte ou images. Le vocabulaire employé par cet algorithme est directement calqué sur celui de l'éthologie, nous parlerons donc, de fourmis, de connaissance collective, de phéromone, d'évaporation, etc.

Dans la première section de ce chapitre, nous présentons notre approche utilisée pour la conception du système de chiffrement dont l'objectif principal est l'obtention d'un système de

chiffrement symétrique non déterministe dont la clé secrète est générée pendant l'application de l'algorithme ; donc elle n'est pas connue à l'avance ce qui rend difficile le travail du cryptanalyste. Pour ce faire et étant donné que dans le précédent chapitre le problème de cryptage a été ramené en un problème d'optimisation, il a fallu, tout d'abord, coder le problème d'une manière adéquate permettant, ainsi, d'effectuer les opérations propres aux fourmis tel que : le dépôt, l'incrémentation et l'évaporation de phéromone, etc. Ainsi, le codage adopté est celui validé par la première métaheuristique d'AEs testée qui est le codage à base d'occurrences. Ce dernier permet la concrétisation des déplacements de fourmis par exploitation de la notion de permutation.

Ensuite, nous avons bâti notre algorithme en définissant soigneusement : la fonction d'évaluation cherchant à maximiser la différence entre l'information initiale et celle codée, le mécanisme de sélection adopté pour le déplacement des fourmis, le dépôt et l'incrémentation de phéromone, l'ensemble de la connaissance collective, etc.

IV.2. Motivation

Les algorithmes de colonies de fourmis forment une classe des métaheuristiques récemment proposée pour les problèmes d'optimisation difficile. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. Une colonie de fourmis communiquent indirectement via des modifications dynamiques de leur environnement (les pistes de phéromone) et construisent ainsi une solution à un problème, en s'appuyant sur leur expérience collective. C'est d'ailleurs cette dernière notion que nous cherchons à exploiter à travers l'utilisation de la métaheuristique de colonies de fourmis pour la résolution du problème de cryptage.

Ainsi, les objectifs que nous nous sommes fixés consistent à comprendre et isoler les mécanismes intéressants de cette métaheuristique, utiliser l'approche de chiffrement à base d'occurrences validée dans le précédent chapitre, suivant cette optique et appliquer l'algorithme ainsi conçu au problème de chiffrement de données texte ou images.

D'une manière générale, notre objectif est de développer un algorithme de chiffrement à base de colonies de fourmis par du constat simple que les colonies de fourmis résolvent des problèmes complexes, bien que l'intelligence d'une fourmi soit limitée ce qui est équivalent au fait de dire que, l'intelligence du système entier est plus grande que celle de la simple somme de ses parties.

IV.3. Algorithme proposé

Afin de bien comprendre notre algorithme de chiffrement proposé, AntCrypt, une description complète des différentes étapes de l'algorithme profondément inspirées de la métaheuristique *ACO (Ant Colony Optimisation)*, sera présentée. Des retouches ont été apportées sur l'algorithme de base pour l'adapter au problème étudié qui est celui de cryptage. Ainsi, le schéma de l'algorithme adopté est le suivant :

Algorithme IV.1 *AntCrypt*

Entrées
Codage de la donnée originale
Ensemble de paramètres (*m* : taille de la population, ρ taux d'évaporation)
DEBUT
1) Initialiser la connaissance collective // connaissance collective $\leftarrow \Phi$;
2) Créer la population initiale comportant *m* solutions initiales (*m* : nombre de fourmis) ;
Répéter
3) Construire les solutions;
4) Enrichir la connaissance collective ;
5) Evaluation des solutions de la connaissance collective ;
6) Sélection ;
7) Mise à jour de phéromone ;
Jusqu'à Satisfaction du critère d'arrêt
Retourner la solution trouvée ;
FIN

IV.3.1. Codage adopté

Pour prouver encore une fois l'efficacité de l'approche proposée et précédemment présentée exploitant un chiffrement bâti autour de la notion de nombres d'occurrences d'éléments de la donnée à chiffrer, le codage adopté par AntCrypt sera le même que celui d'OEEA.

La figure IV.1 rappelle le codage de données à utiliser.

$O(e_1)$	$O(e_2)$	.....	$O(e_i)$	.....	$O(e_{l-1})$	$O(e_l)$

Figure IV.1. *Codage d'une solution sous AntCrypt.*

Où : $O(e_i)$ est le nombre d'occurrences de l'élément e_i dans la donnée, l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Ainsi, notre algorithme AntCrypt cherche à changer itérativement la répartition des nombres d'occurrences $O(e_i)$ sur les différents éléments d'une certaine donnée avec génération aléatoire de positions dans la solution afin de créer le maximum de désordre dans leurs positions. Cela donne, aussi, l'avantage de produire des éléments (caractères/pixels) inexistants dans la donnée initiale.

IV.3.2. Création de la solution initiale

Initialement la connaissance collective (le savoir partagé) est vide. Dans cette étape on crée la population initiale comportant *m* solutions (*m* : le nombre de fourmis).

Nous désignons par *DonneeInitiale* le codage de la donnée originale. Donc, il est représenté par un vecteur dont les éléments contiennent les nombres d'occurrences des 1393 caractères (respectivement des 256 valeurs possibles de chacune des composantes R, V et B codant les pixels) dans le texte (respectivement dans l'image). Ces éléments sont notés :

$$O(e_1), O(e_2), ..., O(e_n) / n = \begin{cases} 1393 : Texte \\ 768 : \text{Im}age \end{cases}$$

Nous générons ensuite *m* fourmis susceptibles de créer, chacune, une nouvelle solution. Pour ce faire, chacune des fourmis est positionnée aléatoirement sur un élément de *DonneeInitiale*. Nous avons choisi de noter les différentes positions courantes par *PosCourante*. Ensuite, chaque fourmi va se déplacer un certain nombre de fois vers d'autres éléments (d'autres nœuds) notés chacun *PosSuivante* par application de la roulette aléatoire, et elle permute lors de chaque déplacement le nombre d'occurrence de l'élément « *DonneeInitiale[PosCourante]* » avec celui de l'élément « *DonneeInitiale[PosSuivante]* ». Une fois une fourmi s'arrête de se déplacer, nous obtenons une solution qui sera ajoutée à l'ensemble de connaissance collective. Alors à la fin de cette étape, la connaissance collective sera augmentée de *m* nouvelles solutions (chaque solution est obtenue par une fourmi).

IV.3.3. Construction de solutions

Après l'étape d'initialisation et dans le but de construire de nouvelles solutions sur une itération donnée de l'algorithme, les fourmis vont choisir à partir de la connaissance collective cette fois-ci leurs points de départ pour construire de nouvelles solutions. Ces points représentent les solutions ayant les plus grandes quantités de phéromone, c.-à-d. celles qui perturbent le plus la donnée initiale par rapport à une version cryptée représentée par la solution choisie. Donc le mécanisme permettant la construction d'une solution est le suivant :

1) Choisir les *m* meilleures solutions de la connaissance collective ;
Pour i = 1 : *m* ***faire***
    2) Choisir une solution *Sol* parmi les *m* solutions choisies en (1) ;
    Pour j = 1 : NbrMaxDép ***faire***
        3) Positionner aléatoirement la fourmi sur un élément *Sol[PosCourante]*;
        4) Déplacer la fourmi suivant la méthode de roulette aléatoire vers une autre position *PosSuivante* ;
        5) Permuter les deux éléments *Sol[PosCourante]* et *Sol[PosSuivante]* ;
    Fin pour
    6) Ajouter *Sol*, la solution qui vient d'être construite, à la connaissance collective.
Fin pour

IV.3.4. Evaluation

La fonction d'évaluation (ou d'adaptation) quantifie la qualité des solutions calculées. Celle que nous avons définie pour associer des valeurs d'adaptation à nos solutions calculées est la suivante :

$$F(Sol_j) = \sum_{i=1}^{n} \left| O_j(e_i) - O_0(e_i) \right| \quad \text{(IV.1)}$$

Tels que :

Sol_j : Solution j,

n : Nombre des éléments du vecteur *DonneeInitiale*,

$O_j(e_i)$: Nombre d'occurrences de l'élément (caractère / pixel) i dans la solution j,

$O_0(e_i)$: Nombre de répétitions de l'élément (caractère / pixel) i dans la donnée initiale.

IV.3.5. Sélection

Le rôle de la sélection est de distinguer les solutions sur la base de leur qualité, en particulier, pour permettre aux meilleures solutions d'une génération donnée d'être copiées dans la génération suivante.

Dans notre cas, lors du passage d'une génération g à la génération qui suit (g+1), nous favorisons les m meilleures solutions parmi celles de la connaissance collective globale y compris celles construites durant la génération g.

IV.3.6. Manipulation de phéromone

IV.3.6.1. Incrémentation de phéromone

Suivant la fonction F donnée en (IV.1), nous devons choisir la quantité de phéromone déposée pour chaque solution. Le mode de dépôt sélectionné peut fortement modifier le mode de convergence de l'algorithme. On pourrait choisir de déposer la même quantité à chaque solution. Cependant, on peut aussi décider de donner une puissance plus forte aux phéromones déposées sur les solutions améliorant l'altitude. C'est cette seconde solution qui a été appliquée dans notre algorithme.

Pour chaque solution *Sol*, nous calculons leur efficacité selon la formule (IV.1). Si cette solution existe déjà dans la connaissance collective on incrémente leur quantité de phéromone comme suit sans la réinsérer :

$$Phéromone(Sol_i) = Phéromone(Sol_i) + \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.2)}$$

Tel que EffMax est l'efficacité de l'une des meilleures solutions calculées de manière déterministe en permutant dans un ordre choisi le $i^{ème}$ élément avec le meilleur des l-i éléments restants (en maximisant la différence entre les deux éléments en question).

Sinon, si la solution n'existe pas, elle sera ajoutée à la connaissance collective en lui attribuant une quantité de phéromone selon la formule suivante :

$$Phéromone(Sol_i) = \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.3)}$$

IV.3.6.2. Évaporation de phéromone

Comme dans la nature, les phéromones sont des substances chimiques qui s'évaporent au fil du temps. Donc, il sera nécessaire que l'algorithme imitant ce coté de la nature représente ce phénomène afin d'éviter aux fourmis de converger vers un maximum local.

$$Phéromone(Sol_i) = Phéromone(Sol_i) \times (1 - \rho) \qquad \text{(IV.4)}$$

Où ρ est le taux d'évaporation.

L'évaporation de phéromone de toutes les solutions après une période de temps permettra de redonner de l'intérêt à d'autres solutions qui peuvent nous amener à la meilleure solution. Le taux d'évaporation ρ doit être bien choisi, puisque s'il est très faible, les phéromones s'accumulent et atteignent la borne max et nous risquons alors de s'enfermer dans un maximum local, et s'il est trop grand, les phéromones atteignent rapidement la borne min et beaucoup de solutions n'auront pas la chance d'être choisies par les fourmis.

Dans notre application, ce taux doit aussi dépendre des valeurs des éléments du codage. Si la différence entre deux éléments est grande, c.à.d. la fonction F va prendre de grandes valeurs correspondantes à de grandes quantités de phéromones, donc le taux d'évaporation peut prendre une valeur importante. De même, il dépend du nombre de fourmis travaillant sur une même génération : beaucoup de fourmis dans une génération augmentent la possibilité d'avoir deux solutions identiques, c.-à-d. la phéromone sera incrémentée plusieurs fois de suite, alors le taux d'évaporation sera important.

Ainsi et après un certain nombre de générations fixé expérimentalement, nous avons varié un pourcentage d'évaporation de 0% jusqu'à 0.5% en s'inspirant, au début, de l'application de l'algorithme ACO pour la résolution du problème de TSP [Barg, 2005]. Ce choix a été, ensuite, validé par expérimentation.

IV.3.7. Critère d'arrêt

Après un certain nombre de générations fixé expérimentalement, l'algorithme converge vers la meilleure solution trouvée. Reste de générer ou de reconstituer la donnée cryptée à partir du codage de la solution proposée. Ceci correspond à l'opération inverse de l'opération de codage. Il s'agit d'une opération de décodage qui procède comme suit : à partir du codage de la solution proposée qui est un vecteur de nombres d'occurrences des éléments de la donnée chiffrée, pour chaque élément *i* du vecteur *Sol*, on génère *Sol[i]* positions aléatoires du caractère/pixel correspondant à l'élément *i* dans la donnée cryptée.

IV.3.8. Déchiffrement

Nous arrivons maintenant au processus inverse qui permet de rendre la donnée à nouveau intelligible sans aucune perte d'information ; il s'agit du processus de déchiffrement.

Notre algorithme proposé est un algorithme symétrique, donc la clé générée doit être maintenue secrète. Cette clé s'obtient au fur et à mesure du calcul de l'information chiffrée au fil des générations. Sa valeur finale correspond aux permutations des positions des nombres d'occurrences des éléments de la donnée chiffrée pour obtenir celles des nombres d'occurrences des éléments de la donnée en clair. Le processus de déchiffrement consiste donc à replacer les éléments dans leurs positions initiales suite à l'introduction de la bonne clé.

IV.3.9. Réglage des paramètres et résultats

Cette section est consacrée d'une part, à l'optimisation ou la fixation du jeu paramétrique et d'autre part, à l'application de l'algorithme proposé *AntCrypt* sur les mêmes données de test utilisées pour l'évaluation des AEs proposés et présentés dans le précédent chapitre. Ils s'agissent des données faisant l'objet de la figure III.1. Dans un premier temps, nous allons exposer notre stratégie pour établir et fixer le choix paramétrique de notre algorithme. Puis nous allons présenter les résultats d'application d'*AntCrypt* sur les données modèles (textes et images), à savoir la donnée chiffrée, la clé générée, la quantité de phéromone, l'efficacité, le temps de chiffrement et de déchiffrement.

IV.3.9.1. Réglage des paramètres

Les figures IV.2 et IV.3 présentent les résultats des différents tests effectués en termes d'efficacité et de temps de chiffrement. Ces valeurs représentent la moyenne de dix (10) exécutions successives appliquées sur l'image de test Lena pour différentes valeurs de nombre de générations et de fourmis.

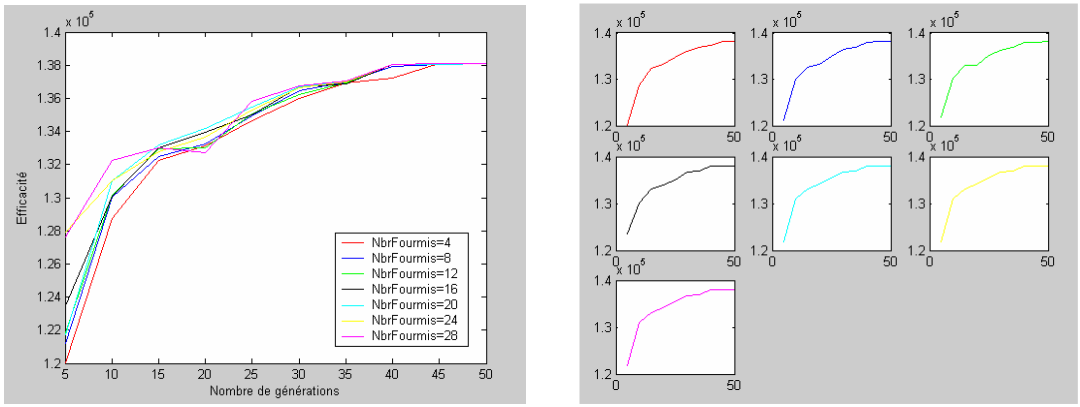


Figure IV.2. *Influence du nombre de générations et du nombre de fourmis sur l'efficacité.*

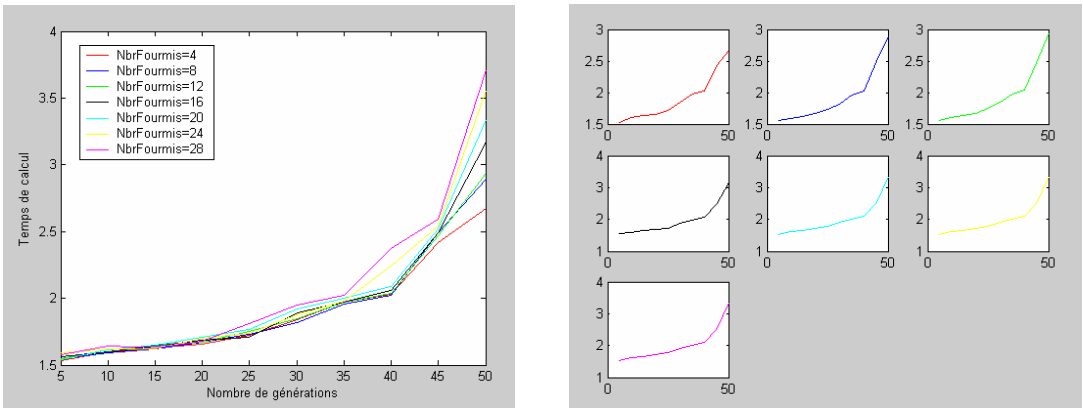


Figure IV.3. *Influence du nombre de générations et du nombre de fourmis sur le temps de calcul.*

D'après les résultats résumés à travers les figures IV.2 et IV.3, nous constatons que la majorité des résultats des tests ont des valeurs d'efficacité assez proches. La meilleure valeur d'efficacité est obtenue au bout de 50 générations à 20 fourmis (Efficacité = 138077) mais avec un temps de chiffrement assez important par rapport à l'ensemble de test (3,17 s). Cependant, le meilleur temps de chiffrement est obtenu pour le test qui a été déroulé sur 5 générations et 4 fourmis (1,53 s) mais malheureusement avec la plus faible valeur d'efficacité (Efficacité = 120009).

Les tests déroulés sur 45 générations avec 4, 20, 24 ou 28 fourmis, et 50 générations avec 4, 8, 12, 20, 24 ou 28 fourmis ont donné de très bonnes valeurs d'efficacité (138073, 138070, 138073 et 138074, 138075, 138074, 138076, 138073, 138072, 138075 respectivement) mais après de longs temps de calcul (2.42 s, 2.52 s, 2.54 s, 2.59 s et 2.67 s, 2.94 s, 3.34 s, 3.56 s et 3.71 s).

D'autres tests déroulés sur 5 et 10 générations ont montré, cette fois-ci, de bons temps de calcul (compris entre 1.53 s et 1.64 s) mais de mauvaises valeurs d'efficacité (comprises entre 120009 et 132263).

Donc, notre objectif sera de réaliser un compromis entre la valeur d'efficacité et le temps de calcul. En effet, le test déroulé sur 40 générations avec 12 fourmis semble le plus satisfaisant. Il permet d'atteindre une bonne valeur d'efficacité (Efficacité = 138063) en un temps de calcul très raisonnable (2.04 s). Ca sera le paramétrage à adopter pour *AntCrypt*.

IV.3.9.2. Résultats

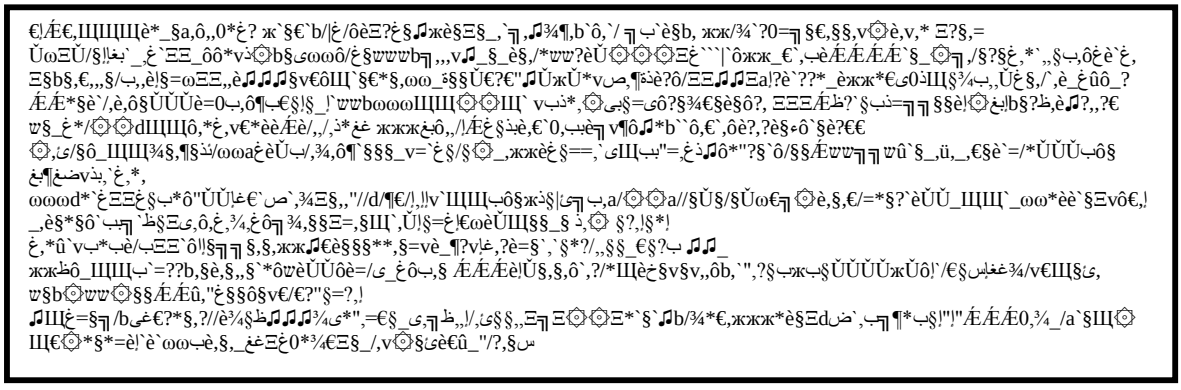
Ci-dessous, nous présentons les résultats d'application d'*AntCrypt* sur les mêmes données modèles utilisées pour tester nos trois algorithmes présentés dans le précédent chapitre : Texte1, Texte2, image Lena et image Logo suivant le paramétrage fixé dans la section précédente (résumé à travers le tableau IV.1). Toutefois, et pour démontrer l'aspect non déterministe de notre approche de chiffrement par exploitation de métaheuristiques, dont l'ACO fera l'objet ici, nous présentons, cette fois-ci, pour chaque donnée originale deux versions chiffrées en deux instances différentes accompagnées des clés de chiffrement générées dans chaque cas.

	Valeurs des paramètres d'*AntCrypt*
Nombre de fourmis	12
Nombre de générations	40

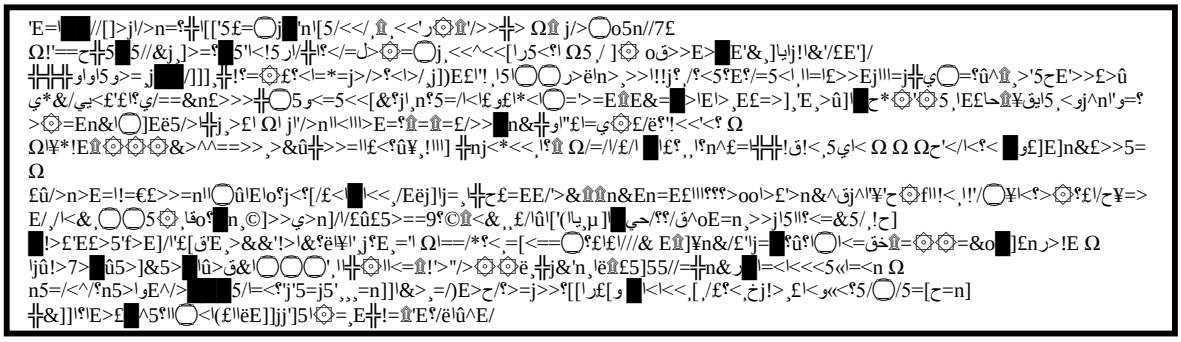
Tableau IV.1. *Valeurs adoptées pour les paramètres de AntCrypt.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure IV.4. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets) :

124	398	407	770	921	266	896	19	794	90	918	245
843	914	614	547	809	904	560	80	375	908	879	886
901	816	830	543	573	900	156	196	98	254	906	314
230	443	917	497	903	418	288	907	289	680	147	861
820	642	902	259	916	840	30	898	905	102	822	635
317	161	920	465	292	458	546	91	911	246	910	833
111	912	155	193	640	919	923	643	357	681	121	725
915	892	581	899	922	216	909	913	789	549	179	578
28	552	3	650	1070	379	850	996	934	933	1104	285
459	997	1124	1242	1026	897	1171	1075	8	521	64	281
1017	955	985	872	262	485	774	16	85	709	279	97
628	120	374	87	286	1112	277	862	518	47	255	1097
1203	45	163	60	599	12	461	5	848	1240	92	1032

331	174	888	83	1073	20	676	342	494	280	422	487
607	1020	958	61	1154	601	95	1100	112	386	1238	1196
1185	1119	393	669	1	756	1172	1012	1102	274	394	363
237	755	267	6	219	389	893	187	1225	437	175	683
834	659	1215	1085	1138	819	447	688	436	764	1158	117
206	1024	675	695	264	319	345	76	127	1183	484	1006
1191	540	1205	210	384	689	655	991	71	928	1130	416
1129	324	761	1134	1089	1029	449	1200	704	231	1013	466
474	1166	813	302	506	1050	490	1204	1092	382	194	744
1066	682	164	72	630	972	15	1213	738	309	1199	531
1149	1084	1202	971	535	395	1031	731	988	737	41	1188
1142	584	336	839	113	818	1245	306	723	657	620	403
252	858	994	423	1036	563	1217	1137	135	108	1133	1088
1160	810	829	946	1058	1068	952	944	751	673	1110	79
1232	835	733	557	1193	982	1179	298	1052	1101	198	1144
225	232	949	400	229	981	545	846	954	432	778	411
634	1161	1042	457	18	250	68	1123	866	462	1243	1051
1120	768	960	990	1189	1182	491	827	185	1145	1148	714
1115	428	574	995	1067	653	1091	978	27	415	132	940
145	409	665	1234	956	622	727	1227	786	402	1009	859
1236	1224	870	1039	387	239	1028	565	1043	594	176	856
787	424	323	588	1033	715	945	1195	359	1162	1165	793
791	930	684	299	716	1076	605	101	1241	149	826	412
1181	480	364	1187	1004	14	1226	579	438	1222	542	613
125	664	1176	197	202	1208	593	520	183	1207	248	604
434	1139	1127	1014	211	891	169	747	968	55	1190	339
1178	625	758	698	330	1038	446	425	611	668	646	1170
1174	144	166	283	1244	857	1231	1131	366	48	31	34
1107	52	814	315	24	414	983	1135	860	11	452	937
539	337	812	935	639	507	863	378	637	301	1055	1219
1077	1198	440	943	451	658	478	361	392	82	103	586
693	881	1210	652	1235	37	1001	970	1156	1175	808	184
278	662	799	341	1041	244	736	1030	109	730	984	195
1019	678	728	420	294	1212	796	1218	303	1192	627	645
1079	1109	1046	481	272	488	1116	1096	222	238	21	349
1034	2	1128	1011	130	157	445	167	705	1087	1201	33
470	489	10	598	453	128	39	1061	817	1209	1216	741
926	1184	1008	942	524	969	372	308	118	228	473	815
312	623	1122	516	1078	962	1143	929	1233	989	316	602
941	1125	1223	648	1072	1136	58	1062	224	931	571	1173
987	649	158	1040	612	998	233	864	241	140	9	621
177	88	1047	321	57	1000	1037	1117	340	1151	961	1221
595	973	138	152	959	483	213	479	207	477	191	287
740	522	977	654	284	227	332	141	234	105	842	305
1080	957	572	355	310	1140	660	258	78	322	831	1054
201	1229	467	1045	273	353	629	441	618	63	1074	849
527	924	43	173	710	295	948	708	1057	876	606	36
967	1025	1081	1147	1132	868	979	1082	966	844	1237	1010
1239	845	772	51	391	1168	855	865	823	65	62	874
275	276	159	885	182	1105	1118	1002	603	592	999	759
1035	762	1069	1083	1007	1005	932	1094	824	745	110	828
154	677	638	703	851	204	1180	806	208	760	884	771
610	1230	25	1211	439	925	1022	765	1155	670	672	519
469	356	548	679	1056	1015	236	1167	964	803	1106	218
1194	591	663	587	260	50	69	1141	953	1071	975	701
212	396	550	307	1063	203	811	743	510	84	333	1153
739	566	807	1159	1018	162	1157	385	1023	35	1114	22
644	1108	261	1093	711	963	296	328	706	122	358	597
938	965	504	1197	249	788	1098	790	325	1214	1228	401
1027	525	1086	951	624	529	1099	992	1150	136	667	380
1126	1111	199	1048	1206	580	404	697	251	719	475	40
585	1044	1095	1053	686	350	575	1163	1003	1060	1064	223
950	146	986	1177	752	304	726	123	853	742	168	492
454	766	804	1021	533	247	46	939	42	1121	221	555
226	980	327	17	180	974	798	927	209	1152	692	554
370	126	151	129	596	377	800	1169	1146	408	463	417
569	1103	615	498	326	59	456	1186	1049	137	139	291
427	632	685	1113	433	1220	153	1164	722	354	189	976
523	100	763	509	1016	444	399	369	1065	936	825	220
360	265	947	1059	749	568	1090	838	852	631	365	993

329	268	651	1330	390	500	1331	1332	460	320	468	729
1333	1274	583	617	753	564	56	782	1306	661	1334	263
165	1335	1318	795	713	371	367	1336	134	1305	271	748
1337	81	532	1299	343	486	1338	1339	472	1264	1302	413
1340	362	769	801	633	1273	517	534	867	94	335	732
1247	1290	1316	1260	89	781	718	1341	450	1323	551	1342
1343	1344	894	1292	687	515	1345	608	1346	1347	1246	351
368	767	1248	570	430	1322	735	235	1348	1349	1284	1350
1288	1278	802	847	257	499	746	805	1267	114	1295	514
1351	188	785	558	1317	1352	784	70	537	832	1353	448
181	1354	297	99	44	442	217	609	501	410	1355	837
1356	66	1314	313	38	1357	777	541	559	1358	1320	426
170	1312	1319	1279	186	32	882	171	429	776	1359	1360
880	720	671	1275	406	215	431	1253	347	792	717	1361
200	1362	1313	1363	54	754	1251	476	511	1364	435	1325
1365	1291	338	1257	1315	699	1282	641	712	381	96	1327
482	883	1280	421	1366	513	1367	26	1368	1250	779	1276
1326	242	143	1307	300	1249	1369	508	23	721	690	1298
773	895	887	1270	875	53	576	67	589	455	890	636
626	1328	405	536	647	582	119	656	696	1370	282	290
334	619	1371	1372	512	503	1373	1374	854	544	877	1375
205	493	106	1308	783	77	577	293	1376	1254	344	1252
148	526	256	1377	464	616	214	878	1294	1303	1321	1378
1297	1379	1272	1311	1300	1266	240	590	1380	530	243	1287
567	1258	1293	160	495	707	150	1381	666	702	873	505
528	674	1382	562	694	86	889	397	1277	93	1296	1285
1256	104	131	270	1383	311	49	190	1281	841	388	734
1283	1265	346	775	115	600	1263	1255	1384	352	1310	172
1259	1309	561	869	13	691	142	133	871	373	821	836
269	502	75	1385	253	1386	116	318	1301	376	1387	556
348	74	178	1268	724	107	1324	553	7	1388	1286	192
1389	29	73	780	1329	750	538	797	471	1262	1390	419
757	1391	1261	1392	1304	1289	496	4	700	1269	383	1271
1393											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets) :

80	34	87	94	51	64	91	46	18	32	55	81
44	2	63	86	54	72	45	16	88	84	20	75
82	97	39	13	6	36	67	95	12	53	17	92
24	98	89	14	74	60	25	41	40	76	10	57
66	93	5	11	9	70	3	31	73	61	42	43
23	27	59	28	78	37	52	69	7	71	77	8
99	50	85	96	30	21	29	22	58	19	90	68
26	47	79	15	35	48	38	878	603	723	164	621
401	997	320	274	4	126	364	628	908	150	286	287
600	815	170	482	424	159	307	220	139	213	596	967
217	133	405	644	511	861	909	334	802	964	597	691
449	942	485	721	538	945	862	870	495	513	505	957
623	883	229	506	955	419	767	751	917	56	143	237
831	972	799	62	324	224	479	750	864	255	771	330
378	316	474	564	161	403	172	748	461	223	978	778
166	318	904	540	985	167	741	202	467	569	49	434
793	283	765	568	248	102	250	653	816	359	297	602
232	249	207	196	365	190	754	947	455	490	338	931
788	454	698	649	774	308	588	702	708	631	33	444
557	463	503	913	886	1	715	116	550	516	189	849
244	685	458	584	285	764	975	418	589	877	921	487
309	632	144	701	850	317	932	563	239	643	610	192
348	574	533	732	993	212	809	299	417	790	501	464
922	567	867	331	650	231	466	722	895	666	168	891
529	541	558	762	279	730	635	824	486	177	313	520
581	560	607	140	977	719	859	339	638	599	380	491
662	777	532	890	758	169	745	382	83	580	900	965
534	987	968	395	104	892	107	826	225	670	222	200
893	387	992	292	545	499	961	353	590	919	869	291
772	575	191	160	605	305	180	811	142	410	832	236
450	389	868	314	659	699	163	247	507	335	481	293
110	982	673	214	634	363	675	842	65	995	343	576
814	546	175	488	425	280	910	677	680	265	718	438
497	716	875	414	366	692	710	976	344	548	970	376

851	421	781	837	927	792	388	874	860	836	798	570
627	743	912	937	204	310	803	203	509	686	306	665
131	776	728	808	682	938	657	197	843	735	779	549
739	593	179	640	846	185	717	370	753	379	178	360
857	465	579	565	899	756	158	452	606	737	132	326
629	729	498	784	295	429	221	648	925	375	500	385
408	182	235	616	431	654	768	171	478	346	694	300
609	994	797	125	357	926	766	863	103	257	145	789
924	420	696	321	528	329	646	476	981	148	402	123
227	818	374	396	954	833	749	121	233	198	935	795
462	720	105	153	897	724	705	416	604	230	536	655
539	866	228	697	246	773	998	526	951	426	352	974
591	988	852	647	573	806	642	127	427	100	619	404
713	459	782	328	783	827	433	496	551	208	349	880
807	369	146	371	118	523	181	521	928	613	277	439
939	661	645	946	510	393	484	633	578	601	106	906
787	887	819	278	907	902	553	276	151	770	272	134
394	681	165	358	432	668	810	556	399	129	820	350
136	117	205	812	622	262	817	652	984	109	260	397
916	448	618	559	898	530	742	780	973	689	885	264
847	245	514	687	630	747	524	986	493	303	259	477
270	267	195	442	958	991	215	547	347	269	914	162
667	114	571	537	733	663	124	865	377	586	583	587
761	184	936	135	744	940	368	296	684	941	712	592
137	113	206	706	412	933	241	594	271	688	822	543
561	980	216	332	844	660	535	304	996	949	894	731
423	341	755	734	284	356	918	183	201	504	830	392
390	585	956	457	446	440	492	775	174	651	582	841
456	615	199	149	554	522	238	362	266	813	796	111
959	989	614	876	953	608	119	409	282	544	428	834
639	351	354	664	637	855	301	873	517	519	625	896
963	725	400	736	355	209	595	290	671	508	835	658
381	256	430	979	345	871	786	901	176	714	839	288
437	120	319	243	840	598	983	251	157	226	752	342
700	669	821	435	856	470	273	407	315	726	234	888
406	727	252	704	340	473	443	929	542	823	740	337
785	641	384	693	460	872	218	445	469	155	193	853
386	828	138	298	999	854	881	943	483	101	759	769
147	930	611	678	112	672	624	709	311	141	577	518
791	422	626	480	336	612	950	905	948	617	327	920
471	128	829	562	695	884	240	858	281	122	254	323
367	707	683	453	794	512	711	889	944	475	263	302
372	186	656	294	911	531	845	969	253	738	801	805
966	489	472	383	289	990	210	258	441	879	391	451
525	322	188	763	173	312	413	261	154	962	515	156
800	971	411	882	219	436	398	527	415	555	333	760
674	757	447	620	572	268	952	242	115	502	494	187
194	152	552	804	703	468	108	903	838	636	130	275
960	325	848	676	923	825	679	690	361	211	746	566
934	915	373	1097	1243	1324	1351	1271	1352	1040	1080	1241
1277	1136	1353	1089	1051	1354	1355	1039	1077	1342	1258	1320
1037	1064	1202	1242	1312	1281	1123	1054	1170	1267	1285	1222
1340	1349	1129	1111	1296	1174	1252	1345	1356	1166	1357	1115
1275	1328	1250	1128	1263	1171	1098	1284	1033	1231	1276	1005
1058	1313	1311	1059	1124	1232	1157	1013	1100	1132	1019	1099
1103	1023	1112	1225	1240	1304	1194	1215	1358	1247	1198	1334
1335	1122	1018	1210	1164	1341	1156	1127	1106	1006	1300	1323
1114	1359	1245	1113	1066	1154	1150	1036	1344	1289	1360	1238
1298	1034	1035	1093	1048	1008	1318	1253	1309	1028	1331	1361
1016	1195	1286	1362	1363	1050	1090	1038	1137	1119	1287	1079
1163	1182	1151	1125	1043	1303	1025	1325	1133	1042	1364	1002
1234	1075	1347	1088	1365	1212	1117	1230	1301	1294	1071	1214
1200	1307	1049	1264	1190	1060	1147	1299	1332	1160	1165	1366
1109	1180	1343	1305	1315	1065	1187	1055	1052	1191	1367	1110
1196	1368	1074	1205	1257	1262	1168	1336	1108	1308	1045	1369
1330	1278	1239	1134	1221	1149	1235	1104	1209	1370	1053	1265
1282	1107	1094	1233	1069	1153	1213	1244	1327	1266	1146	1229
1091	1141	1105	1346	1226	1041	1070	1186	1142	1056	1302	1063
1199	1321	1162	1193	1086	1101	1270	1083	1310	1017	1192	1177
1179	1095	1248	1184	1219	1121	1022	1273	1172	1371	1046	1372

1044	1178	1096	1208	1224	1183	1003	1084	1012	1131	1130	1348
1062	1316	1159	1188	1021	1373	1009	1135	1374	1375	1203	1211
1228	1010	1087	1268	1376	1333	1081	1377	1322	1220	1145	1072
1057	1288	1061	1047	1249	1255	1120	1030	1306	1261	1189	1246
1314	1144	1378	1254	1379	1126	1295	1138	1176	1319	1152	1337
1216	1201	1236	1380	1293	1076	1185	1291	1158	1143	1381	1029
1297	1067	1259	1382	1140	1155	1068	1269	1031	1383	1384	1118
1339	1260	1290	1102	1385	1386	1082	1280	1167	1000	1078	1085
1073	1387	1197	1272	1007	1181	1227	1011	1001	1161	1206	1251
1116	1204	1388	1173	1032	1350	1024	1389	1207	1292	1217	1326
1237	1390	1223	1218	1148	1391	1392	1279	1274	1026	1027	1256
1092	1014	1015	1393	1317	1139	1338	1329	1169	1004	1175	1283
1020											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)

Дé¥¥§&¥╣c:¥~xx♣♣♣;>¦;>}شЧ&¦╣╣¥‼<ם;gةةq¥;}¬;ة§¦ ☼☼5¥<<<î♣ΘΘЯΘ{ממxcx;<¥ ╣¥ةxt;:g¥♣¥;ئ‼;<xx¥:§¥ة§}ة;¦;¥ Ч¥x;
ΘΘ¥<xr♣gt¦¥x☼rxgt¬g¦x4và¦x¥x>╣Θة_¦ƒîxЧ¦;<ƒv*§r☼¥ئxة{¥x¥xqx;&xДxîةxîxr:ئЧq¬xƒ♣>¦¥<:q¸~╣®¥xé§g§;شg;g&x؟xxxq;xtةxq;
¬{î<¥l♣ة;c§¥xДî¥ةةЧЧЧx{xî0>¦xxxîà♣¦{¥x_:ة{&>¥x;;¦x♀םئ☼☼;xx¥Θx&♣4ة¥:&<‼Чîx§é§xq{¬<☼ئ&{gwΘממ□¥à&x¦♀♀¥&>;<¥î§
☼☼♣ة¥ةît¥¥rrة;î&;;>♣♣g{x¥x¥Д¥q♀;<☼xة;+>gДД§ة{¥x;qة;îxg¥:;>¦םxj<x♣q¥xة¥{_î<¦cx¥¥¥ƒ{x╣>;xx5¥xДДrg;ةx╣Ч&¥¬xx♀:xةƒ{xj
¥╣╣¥q{î¥☼;¥j5&Ч{g;;q¥¥x♀&é;g;/>:;lxxg¥î;{¦x╣>g;_:;à☼¦;îxxq<¥:&Я&;♀&ةЧjx;x¥q¥>;{î~_;t&§Яם╣>;x/&:♣ЧЧ{&{ـئة¬¥gƒ¥éЯЯî
>¥¥ة§t¦;¦xЧЧ;x¸x¥¦l:g{qةq¥♀☼¥x<¦xv¥;x;ة<¦♀>¦x;&q¥:ة;xЯЯàtgƒ{Я¦ة¥îcq¥♣ggî¦xîx<t{ة>;{¥t¬q>¬gΘ<§q:>®
♣&_§;<ئ§rx☼x>¥♀¦¥¥♀g<¥à☼tt~¥é♣♣♣xà5¥ƒ;_<<¥¬ ☼¥ة¥¥g~¥¦j¥{x¥╣¥¥g╣{¥ة~ة¥ة¥¥¥¥;;¥gƒ¦lx;:¬{¥
♣qx¥¥xx¥r{t{<®ggî;□¦gx_ة{_:x{t¦t>4g¥x<<¥¥q;à¥qgx¥♀;¥ƒة<_╣và¥r§xàx;§>xƒƒ♣xtq¥{>îxtx¥¥éxxة;_xئة;{;x♣xgةrv<x4¥;x<xةg;§خ¦
x>¦tةةà§ة:¦v;xئ;gةg;;&¦lt®&g{<{;;¥ƒ¥;ئx╣éj¥ةЧةة¥¥î¥xΘΘî¥qةtc5;ةjΘx>;:_ة<{♀¥tةîםg{ة¥<tةxxrq¥<¥{;:¥☼;4;x;rü♣ƒ¦5;¥;¥xة§t:xx¦¥♀{§
<g{¥ة¥>&xxx¥;Дx¦g:ة;:®¥gةДx;¥<v¦ة/¬}>ئ¥¸☼xة~ةx;
ממם╣w♀{xЧئ؟¦îg§ЧЧjx_&;g♣♣_♣؟>ة{&¥tx¦5ЧםЧЯЯ;ة®{{x╣╣Ч╣;îx¥¥>¦♀Θ☼☼♀♀ش§¥x<Я♣¥x<ЧЧ;¥
ממî{xЧ¥îjة<¦;x¦¥><;&xrة¥¬g&<x: Дx{îx:î_'{g§r§Я¥;<x/♣♣¥<{;¥{{{:♀ЧЯЯםЧ♀□~¥x

(b)

♣ء∏y¥ם¥♪¥Ξ◯Ξ¥ن◄¤3¤ن؛¥םΞ♪¤ء±ء؛ن/¥۩ص¥ﭏخنخ¥yyنpp◯p¥◄ءصن]¥خ½ﭏﭏ¥p½yy▲؛p¥¤ﭏنصwb»ش♣y¥;pg♪3خw¥▲¤½ش±bb±
Ξpن§w½ם½ΘΘyp¤Ж♣♣ن¥yw[y«؛۩p؛¥33ن◯m¥"§p½ن¥(ن¥ûw[ن♣)◄bن¥ê½b¥∏ش¥♪ЖЖw»yنΞ¥ن∏أن§♪♪؛ن½¤ממממΞw۞}؛♪¥صنΘ
Θ♣؛¤¥yﭏض◯¤fنm)ن¤yص¥؛»ypن¥y-
"yy"bm؛y۞خ♪ن¥؛۞ﭏﭏ؟w▲¥خ¤p[خ♣±¥±¥ymp∏خﭏﭏخ؛ءЖن]~~◄◄خ¥¤Жﭏ؛yםם∏؛êb¥p╥▲y۩µ¥؛¥نy½Ж3ش¥ن؛p3◄ם½¥¤∏ن¤¥[bmء
y3؛eeنن½yصy¥ش¥b½؛¥ش♪♪نpyخنb½½۞wp7ﭏ3¥¤½╥╥ثbنΞΞ§◯pخن±b½Θp½¤ﭏن♪؛gyﭏ;¤3ءםםص]۞╥y3¥/Θ▲▲ﭏﭏ؛yنp♣خ¥◯◯ش
p½♣¥û◄¤y؛¥¥yyp¥∏ ∏ ▲
[yy¥3نêش¤[◄◄½ﭏpyם۞¥ن¥صЖЖ♪Ж½¥¤¥ûن¥¥۩ن]"ث«ن¥♣♣םwy؛~p3¤pΘЖ¥ن؟Ξن¥◯◄ن¤"ن¤¥ybpy♪♪♪نy3±ءشﭏpyءننwخwyخ¤»
w♪)خ7½ن7؛½ن¥صu[Ξخû¥3ن¥خy±♪¥y▲◯¥¥؛½¥ﭏﭏﭏﭏfخΘ∏mنن½y╥◄Ξpwص¥"(¥½]ن½ن◯◯◯¤p½ﭏﭏf»¥û½¥w-¥¥-¤خخ
ن¤§½¥¥]b¥¥[¥¥wنb½¤}¥¥¥خ¥ن؛ش±y»[¥¥±
۩∏¥¥ن¤/¥نصخyw◯ن◄ﭏﭏء◯Ξy¤½ن¤½ym½◄pЖ؛¥p¥ن∏∏∏∏¥y±¥3ص♪♪♪]«▲▲صpﭏن؛خ¤ء۩۩۩wخن¤¤)y¤ص¥p/◄◄◄"f¤3y)
♣¥y½¤y3ن¥ننp◄)»b¤3نن¤▲ءp¥3¥3¥نص♣∏ﭏﭏ∏yyyb؛¥ص¥¥y¤۩3؛؛◯םםb§¤ننpن«-]±۞ننp¥pb±¥¥ن؛b¥¤؛ن½ن[¥۞p½◯◯¥±--
نwثΞΞ±yb؛y-
±ءyp¥ﭏw¥♪¥ם¥p3¤¤¥ص♪ﭏmy¤¥نb¤¥p"ننЖﭏنb؛صp½)¥▲;]خ♪ن«ن¥pصyصםםםص¥¤¥¥¤ءﭏﭏﭏﭏby¥½¥ن¥♣♣♣؛ﭏﭏن¥نن▲نننن¤yءخ∏♪♪¤صنننﭏ
ﭏص/¥y)ن½y½ن-خ؛ywن◯p¥gyشûyخ◯ن¤±bن¥p[yy۞w¥؛نن¤b-¥)¥¤wءش]نü♣¤؛fصΘΘخننΞ{¤ن¥ם)[¤
♣¤pby؛¥خ¥Ж¥mΘخy"]♣؛خن"yЖy؛نΘם¥ن¥¤§¥◄╥؛۞▲؛-؛؛y//¤ءءص¥¥םם◯¥y [ن◄◄-¥¥ם¥¥ם
¥ﭏﭏن¤ثΘﭏyy½╥▲b¥نЖصن◯§ن∏¥♪♪ثםם؛ءû[/m♣ﭏﭏ¥½[∏♪

(c)

Figure IV.5. ***(a)*** *Donnée originale Texte2,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

700	340	911	855	661	6	451	910	17	3	151	267
937	143	936	914	475	880	745	874	939	897	155	935
281	730	209	938	173	638	777	191	329	930	384	421
673	131	485	462	512	259	736	644	457	728	932	593
933	65	920	924	926	918	349	196	232	309	553	782
917	369	919	927	455	878	255	921	913	929	156	931
912	759	400	922	616	915	594	691	243	934	592	916
819	361	128	923	486	24	925	159	748	301	928	140
411	379	1093	16	599	1169	1108	1079	1040	985	1021	141
799	1006	1067	1107	82	572	1014	325	977	498	9	943
974	670	394	805	270	49	178	658	808	604	392	1229
642	456	1058	80	1184	78	669	472	1045	1214	445	1011
195	492	242	952	389	1069	814	839	1060	591	983	895
285	248	67	947	1194	460	380	38	46	584	482	430
631	1236	478	1012	1004	297	570	1240	228	332	390	1127
1153	40	992	991	1151	625	44	1106	47	1189	99	823
986	706	395	85	559	205	28	641	887	197	319	979
1094	554	882	1066	822	1166	531	564	949	305	398	654
694	544	200	160	801	1017	189	1031	1083	1143	476	1196
1002	198	894	348	1052	619	69	214	1192	1099	120	1134
318	300	534	73	862	104	425	328	1092	1141	677	296
692	1212	466	612	1190	144	960	686	1188	53	404	668
683	381	106	672	523	1145	688	42	1053	342	102	779
292	757	1055	1183	852	514	34	890	702	525	1233	863
870	79	444	858	1241	851	904	1250	948	346	224	31
807	754	528	942	794	1140	511	1101	347	295	1198	289
436	408	587	308	269	403	247	1074	25	1148	27	264
194	489	1160	163	1056	1225	298	322	965	1202	563	237
1035	1179	941	56	1051	1057	1064	434	978	1015	532	413
602	1049	71	772	162	1123	483	1142	231	1152	844	1078
1235	1159	356	418	1047	1082	737	273	1137	676	1144	263
372	1039	530	1098	440	284	1191	1081	94	1205	29	783
1216	603	630	859	877	598	473	4	274	944	674	681
709	1154	565	1008	837	662	549	671	1003	215	710	1228
1243	758	1016	962	1238	175	1217	1158	518	57	1172	628
764	876	1223	606	7	253	657	968	1245	766	999	542
324	775	399	738	989	1097	1009	515	238	406	770	245
993	499	1167	30	217	230	1077	299	787	74	537	1029
784	802	645	76	1147	1197	415	452	1126	45	517	629
1227	1000	940	569	632	618	164	172	101	18	88	216
1135	950	397	377	879	809	1226	1176	527	227	1203	431
833	276	970	87	1027	256	480	843	409	637	286	1013
1248	142	725	605	743	1114	567	643	791	640	576	1090
1244	150	689	1018	650	589	311	516	995	1178	519	1175
611	828	621	857	1122	1020	1065	294	474	1230	959	376
869	501	898	370	545	1117	816	1071	1110	43	526	623
115	836	655	680	26	494	1195	447	1023	385	513	548
969	161	250	1185	1034	110	111	860	15	561	793	1170
961	820	133	432	1116	97	114	279	93	193	495	422
1246	1059	997	112	212	1210	1130	33	219	60	469	698
1231	174	812	479	763	1207	1218	277	50	450	416	41
954	582	116	488	127	1128	552	697	1209	1080	306	490
1085	1129	539	981	953	337	449	211	726	1061	1124	510
715	1091	861	83	1007	827	477	330	800	849	546	8
747	419	10	186	454	500	66	583	899	357	401	1001
971	659	856	226	1234	240	233	91	761	343	1046	1146
595	107	1186	1237	1103	1139	1076	1048	647	579	722	353
608	386	660	1115	596	1042	610	636	405	718	290	529
1138	234	433	1112	1164	964	945	742	967	773	20	77
1070	1221	1072	804	1224	663	1161	721	1120	1118	458	797
1193	707	1119	326	453	35	505	1030	533	126	1041	64
769	753	14	63	1024	1100	387	149	744	239	1200	703
1037	504	522	32	956	875	536	723	998	345	1171	435
1068	1155	366	832	1213	491	1028	1173	829	429	23	374
1181	976	11	973	393	1187	988	1232	135	1102	169	493
278	664	134	90	821	885	1163	762	741	627	108	892
1199	996	367	958	982	1165	417	1174	95	1150	557	1033
740	333	891	382	129	972	1247	1109	1121	266	204	1206

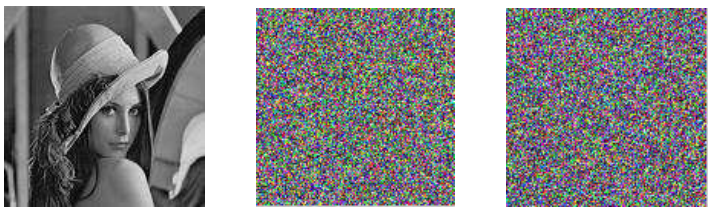
1010	388	187	966	1204	955	987	70	790	1239	503	352
957	262	1084	272	59	866	414	1168	1219	1201	1222	907
1111	317	963	1113	1026	497	487	818	810	109	442	439
190	780	423	896	130	363	1086	798	617	1050	1087	994
1105	208	1211	580	796	89	291	68	1025	251	1182	1005
682	1104	378	980	538	646	118	323	699	1157	359	1136
236	1156	1131	1075	1043	1149	853	96	229	351	179	1220
180	138	145	424	177	1019	1032	560	975	136	1132	1125
261	774	685	568	886	675	315	202	158	1036	168	1133
713	601	1215	600	176	1054	1208	817	865	678	739	864
789	1095	1162	509	1022	614	327	1249	946	607	806	760
185	1062	221	667	902	566	316	635	665	39	813	717
786	951	834	624	438	1088	58	1177	984	1044	62	719
1251	900	1096	100	1073	166	990	695	1180	463	2	334
124	1242	1038	123	148	889	615	121	1089	651	407	481
749	1063	1271	470	86	597	465	1332	1307	260	1342	1275
881	368	371	48	1343	1301	1270	1344	838	1313	1345	666
1346	132	265	871	302	840	84	502	842	1347	222	765
573	252	1348	1322	732	246	1349	835	1334	13	1319	1339
464	1350	778	771	1351	355	268	590	845	1311	210	223
1289	577	184	1282	1352	735	1269	1293	1353	1278	459	1263
241	207	1274	1354	652	1325	1298	446	12	588	1268	873
884	1355	622	1252	125	344	551	213	304	1356	1265	1357
1296	609	1330	908	693	98	5	868	555	1320	225	428
687	1331	383	541	1308	81	188	756	1267	558	154	1272
1358	1359	341	1258	52	847	467	1360	776	724	1361	22
1279	103	280	36	220	1362	795	634	578	461	571	484
181	303	321	690	1363	734	905	1285	574	19	733	412
396	313	402	1364	1365	331	139	550	51	1366	1281	365
335	746	1367	586	257	336	613	287	1253	803	883	792
653	850	72	585	310	831	1329	360	729	192	1368	218
426	21	496	727	1273	815	750	320	1326	1318	1369	1327
1370	1371	639	684	1262	358	1372	903	1373	1257	581	825
312	906	117	1374	1341	649	1375	767	1376	1264	556	1377
373	235	1378	826	1379	1336	731	872	1316	203	468	1304
1380	848	708	167	543	1291	1261	626	1256	888	712	206
1340	307	1294	854	535	338	1337	1315	1259	909	410	37
244	620	1277	443	1338	362	75	1287	471	867	153	768
575	1381	811	824	1314	171	1323	105	1254	1290	1328	271
1303	1266	846	183	441	1300	648	521	1324	1255	146	893
375	1382	1383	508	354	1283	696	54	254	282	283	656
785	788	448	841	1310	506	137	1317	1295	165	830	1333
339	1286	701	714	679	751	1299	540	704	249	147	182
1260	1384	364	1302	157	293	1306	1385	633	92	420	1309
427	288	716	1335	391	1321	55	258	1280	1288	201	61
1386	152	1387	1388	901	705	350	1389	752	1390	1	1391
720	711	755	113	562	1284	1292	1392	1297	314	437	547
1393	507	1305	520	1312	199	119	524	781	170	122	275
1276											

Clé de chiffrement de la deuxième version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

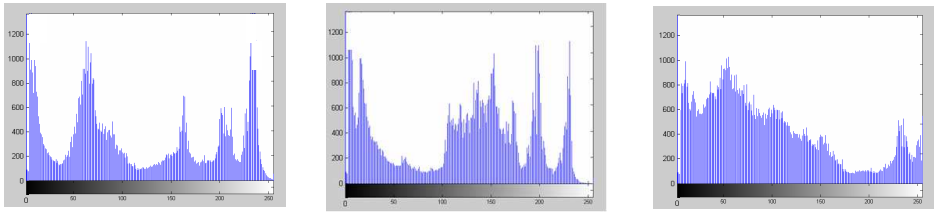
6	162	2	1	279	439	507	352	360	804	758	567
472	465	78	992	500	327	830	580	948	101	535	173
756	991	805	522	928	10	147	416	461	606	769	803
827	692	341	982	129	12	244	944	197	477	831	139
684	658	255	593	356	49	36	172	504	108	272	375
248	575	302	299	81	358	496	773	607	476	852	562
382	726	134	846	667	641	98	73	735	230	548	540
902	484	64	727	643	542	640	380	556	632	723	
835	436	748	109	721	654	619	212	277	894	211	475
705	335	257	844	856	781	938	462	961	11	623	62
651	254	74	4	746	265	525	130	943	185	52	473
457	160	251	860	59	939	447	121	946	511	757	934
258	136	974	350	35	50	483	491	965	32	941	453
339	344	514	929	137	470	27	502	724	131	184	512
362	677	213	871	802	981	39	5	570	345	848	76
328	884	412	238	68	391	376	488	325	292	882	71
142	970	810	196	220	796	730	132	296	622	550	490
590	574	3	560	87	826	158	8	518	817	973	635

819	409	689	498	206	353	526	890	41	617	440	647
396	915	935	797	96	242	975	297	285	608	177	264
782	563	99	203	659	7	544	104	761	629	719	271
333	329	877	581	332	838	778	611	704	762	674	156
972	899	610	930	366	785	927	976	956	31	263	117
312	315	664	775	752	337	413	996	878	853	963	46
637	866	111	538	310	105	30	418	445	179	814	933
151	867	597	274	86	467	609	984	273	152	834	43
408	303	594	44	505	696	419	516	631	732	794	712
284	588	682	435	874	521	858	589	969	906	33	138
398	722	698	529	372	524	93	378	122	486	554	494
448	655	103	833	547	808	9	148	924	855	728	713
386	691	751	226	468	239	311	688	482	628	28	737
753	566	373	602	543	168	118	318	326	813	14	307
309	275	485	818	942	870	653	481	868	950	993	861
690	499	766	648	443	349	679	280	493	673	824	56
424	199	321	889	621	962	166	218	977	714	487	444
336	582	281	402	624	553	387	747	441	390	546	261
743	306	492	603	58	616	267	235	89	528	703	630
811	364	497	381	715	800	250	665	893	94	379	361
913	627	532	262	530	585	980	295	388	21	697	241
903	828	971	851	995	686	672	469	888	615	519	905
595	908	501	832	169	433	897	718	63	508	228	80
816	385	922	115	862	91	423	428	95	559	427	873
786	779	676	909	15	604	979	710	354	680	370	578
601	739	150	245	157	907	342	128	953	417	454	404
881	223	240	82	17	270	646	792	125	153	940	896
266	694	571	552	135	127	898	986	901	669	772	229
842	837	916	346	371	523	47	661	464	771	783	839
790	18	308	460	577	702	154	289	770	990	57	463
114	69	24	642	400	120	671	232	650	97	48	249
774	895	605	178	919	314	863	442	926	53	558	634
921	989	458	29	587	568	741	406	434	145	657	561
791	920	369	340	246	645	821	749	429	614	403	539
330	509	144	945	225	918	541	394	112	537	706	850
405	60	879	900	845	222	374	789	474	625	549	425
209	584	276	317	421	652	426	67	843	599	198	355
947	338	557	305	37	431	319	678	290	876	155	638
596	806	451	331	124	744	809	54	411	51	182	883
912	825	170	807	16	983	857	904	666	836	955	660
988	61	175	420	159	395	742	951	515	133	26	869
815	268	410	205	219	176	191	446	729	347	252	864
534	958	849	207	466	234	65	531	90	269	75	478
384	221	181	795	183	551	22	968	459	911	479	399
414	286	734	670	192	685	925	959	545	777	300	287
174	415	357	106	88	840	960	987	42	859	636	780
471	745	294	701	592	164	600	750	733	377	764	320
949	208	422	716	681	291	140	301	449	711	83	393
165	668	343	952	84	917	700	892	126	113	639	195
954	759	966	573	85	644	102	231	224	92	885	914
180	143	613	40	141	994	887	348	875	200	20	367
188	316	456	110	720	489	503	569	450	260	626	822
119	38	383	793	767	708	186	171	167	579	736	812
572	72	784	163	77	365	555	392	243	765	452	146
283	187	322	282	107	760	683	886	407	693	662	216
841	23	707	363	564	210	359	55	999	964	401	738
217	324	847	506	598	116	201	13	586	253	576	695
823	798	865	591	699	323	256	190	389	872	788	620
288	829	533	763	247	717	313	298	19	985	854	149
278	520	931	998	455	100	687	820	34	351	768	193
923	967	740	755	480	432	25	227	725	161	799	932
189	675	937	663	204	66	957	237	565	334	891	583
618	787	233	304	259	397	997	536	656	612	437	527
495	202	368	936	510	438	633	649	801	754	910	776
880	123	194	214	236	513	978	293	215	430	517	709
731	1165	1126	1054	1145	1227	1122	1168	1335	1190	1338	1157
1224	1003	1310	1038	45	1301	1153	1295	1316	1032	1318	1329
1167	1115	70	1004	1339	1340	1341	1081	1150	1294	1342	1343
1319	1086	1125	1018	1269	1118	1012	1214	1344	1129	79	1332
1061	1213	1111	1205	1305	1130	1252	1093	1315	1057	1037	1299

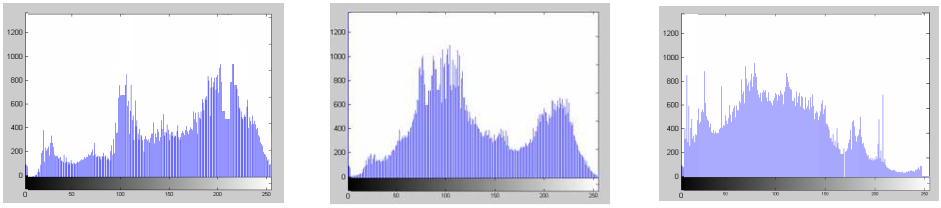
1083	1277	1062	1219	1345	1238	1346	1096	1221	1022	1103	1195
1257	1142	1050	1347	1274	1077	1088	1006	1058	1220	1046	1020
1171	1268	1116	1348	1278	1013	1188	1291	1237	1314	1296	1349
1350	1351	1109	1192	1352	1353	1189	1267	1229	1005	1155	1107
1243	1060	1008	1091	1128	1064	1023	1041	1293	1245	1248	1102
1113	1354	1263	1131	1355	1163	1356	1015	1039	1357	1247	1000
1053	1311	1358	1283	1049	1124	1065	1359	1360	1361	1281	1173
1254	1181	1114	1276	1191	1336	1104	1223	1184	1362	1079	1123
1226	1259	1137	1273	1112	1070	1363	1287	1035	1197	1364	1365
1140	1366	1367	1368	1127	1288	1369	1199	1244	1010	1133	1105
1024	1370	1371	1048	1180	1312	1317	1045	1139	1330	1177	1019
1040	1029	1055	1230	1225	1285	1275	1334	1372	1085	1036	1076
1206	1258	1208	1333	1028	1002	1154	1320	1186	1166	1161	1030
1135	1261	1095	1101	1067	1280	1136	1194	1097	1302	1260	1099
1034	1373	1172	1063	1374	1132	1094	1080	1337	1234	1265	1175
1066	1169	1047	1087	1246	1121	1375	1193	1068	1292	1376	1377
1138	1378	1325	1075	1326	1232	1203	1249	1092	1073	1379	1297
1152	1001	1071	1306	1026	1256	1242	1304	1380	1148	1156	1321
1201	1303	1021	1239	1210	1222	1381	1290	1251	1204	1025	1264
1382	1383	1084	1187	1228	1271	1146	1185	1078	1324	1384	1385
1215	1216	1176	1170	1100	1098	1212	1016	1031	1218	1209	1159
1196	1217	1178	1233	1241	1106	1162	1266	1141	1143	1090	1240
1250	1298	1328	1108	1307	1386	1262	1089	1387	1074	1027	1313
1388	1389	1056	1179	1009	1151	1119	1044	1255	1331	1134	1286
1272	1117	1033	1147	1231	1052	1149	1207	1309	1200	1164	1282
1390	1007	1160	1198	1082	1072	1308	1284	1253	1174	1069	1391
1202	1392	1182	1017	1270	1235	1236	1042	1043	1211	1322	1051
1183	1011	1120	1323	1158	1289	1279	1327	1110	1393	1014	1300
1059	1144										



(a) ***(b)*** ***(c)***



(d)



(e)

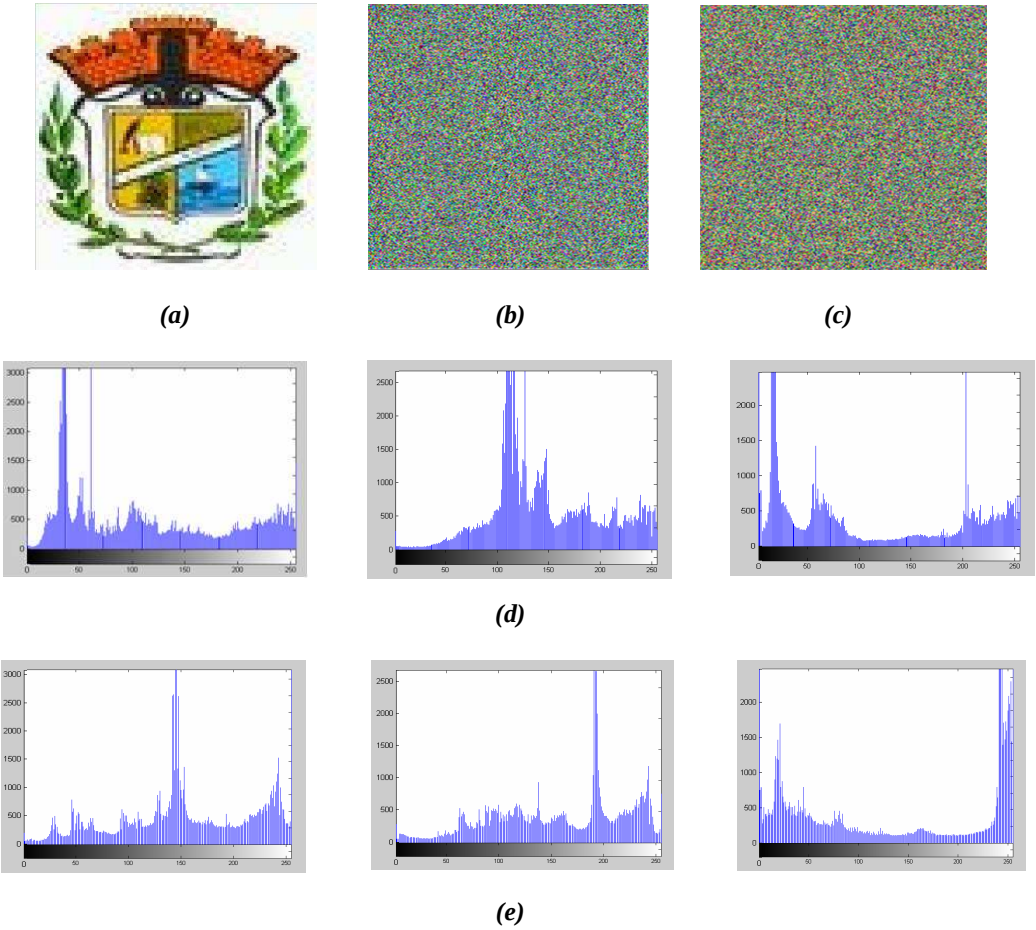
Figure IV.6. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

5	623	186	3	733	275	321	55	398	496	437	199
363	266	63	526	685	316	438	200	324	525	595	106
194	114	109	386	198	126	674	374	751	755	176	273
541	467	369	279	516	661	197	609	447	4	86	679
716	196	117	422	736	430	129	474	261	762	527	630
759	737	615	362	694	367	687	583	581	646	19	676
234	645	203	574	284	580	332	389	517	613	743	637
732	272	209	431	453	408	495	557	651	394	30	242
337	735	551	566	473	477	391	104	50	79	401	11
535	514	500	269	352	143	372	216	664	267	341	499
184	713	543	553	458	300	22	51	718	325	712	68
532	704	512	501	1	7	547	690	763	256	522	536
130	381	215	511	224	371	647	94	485	70	78	494
761	220	436	622	21	577	149	725	582	443	322	334
191	576	424	697	513	10	296	96	550	345	492	88
154	585	506	239	549	456	519	590	702	399	426	505
212	339	120	727	631	692	166	159	201	67	684	742
102	440	537	225	29	655	709	562	567	308	171	644
539	452	264	502	592	416	510	274	111	753	621	561
235	208	226	45	468	370	175	471	410	747	349	42
304	387	677	230	439	157	534	121	170	657	8	552
617	213	691	455	150	43	223	678	484	95	301	36
442	503	598	579	449	610	731	156	635	563	90	249
103	247	432	524	206	633	6	533	584	160	564	721
754	750	91	538	162	722	508	285	250	353	254	744
726	717	597	283	185	559	625	393	570	669	69	504
174	421	457	329	355	87	268	616	136	195	639	336
100	720	668	518	54	116	214	605	319	303	190	58
23	405	342	548	428	376	231	470	373	34	320	670
49	475	132	181	588	271	418	314	167	703	243	404
364	123	419	297	227	48	291	624	461	81	528	738
192	105	766	619	62	25	187	330	115	636	441	450
152	545	760	498	365	221	71	383	153	530	134	660
662	189	757	396	348	368	602	596	238	714	205	671
749	614	435	493	317	395	672	307	182	481	17	47
415	338	27	57	682	640	648	765	60	219	356	689
155	173	673	629	683	529	600	658	448	32	292	384
125	172	327	681	113	740	351	603	24	312	675	148
40	643	15	309	402	294	93	217	232	122	434	135
326	183	281	315	593	480	118	53	288	358	89	142
128	39	699	131	241	41	521	66	343	282	741	734
218	277	420	202	540	13	642	486	589	204	723	478
730	715	340	710	764	80	653	472	211	445	483	411
459	464	433	18	392	138	618	406	377	571	193	591
360	366	158	680	252	469	382	145	52	35	462	72
407	278	133	73	101	412	26	165	745	344	575	12
599	400	739	695	748	310	568	139	293	140	608	403
375	708	587	768	298	666	76	554	251	606	414	572
245	97	222	74	270	507	388	44	601	696	546	594
229	107	706	168	311	350	487	112	667	255	33	444
361	177	454	169	164	479	509	82	686	463	515	228
663	244	357	649	711	259	628	265	397	707	728	423
665	2	632	626	460	654	210	290	124	767	163	333
280	429	295	489	451	698	565	127	37	641	491	688
161	531	302	719	179	75	729	724	604	482	693	347
240	413	61	490	746	378	586	258	84	627	427	248
263	335	16	542	523	233	556	31	144	257	38	64
331	92	544	701	650	466	56	573	188	659	306	578
253	287	756	236	425	299	119	611	65	9	555	323
141	417	110	318	359	20	146	77	656	758	652	207
289	151	237	98	385	620	700	85	137	634	147	260
354	180	276	560	59	380	465	313	569	346	409	638
558	108	328	612	476	488	46	446	262	286	246	178
607	83	705	752	390	497	99	305	28	379	14	520

Clé de chiffrement de la deuxième version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

81	506	289	165	107	184	553	751	570	17	729	724
457	204	65	760	510	256	116	708	314	296	315	381
538	6	734	302	480	33	139	345	54	144	113	57
393	424	399	163	470	71	599	733	51	653	151	30
175	743	321	556	187	232	112	323	363	707	588	183
235	260	128	37	477	453	462	267	88	304	351	56
308	359	762	464	121	374	294	382	715	392	478	609
647	205	431	253	89	594	666	262	622	592	224	322
311	164	277	29	341	291	154	646	699	522	495	74
625	47	320	395	644	77	43	100	494	723	402	706
702	105	152	244	283	737	404	340	750	629	396	764
125	412	49	387	22	364	484	643	337	242	131	389
342	690	120	572	567	435	745	176	466	13	209	403
517	508	279	265	4	252	117	132	475	587	391	310
635	331	492	502	360	372	380	456	710	104	96	669
565	416	257	531	3	326	612	171	568	1	563	444
434	507	41	259	505	577	84	192	147	447	597	747
347	711	335	744	383	421	540	367	394	660	143	281
657	178	339	418	562	436	301	766	27	94	516	66
703	161	91	313	58	63	136	610	203	45	162	292
195	483	317	705	228	62	196	476	18	539	64	368
141	451	503	39	169	307	229	297	672	350	670	626
596	309	9	19	740	353	519	160	673	134	614	400
541	75	649	605	748	223	82	137	388	525	667	655
535	640	90	585	659	448	243	295	170	48	299	237
754	533	726	452	548	528	231	230	590	700	24	586
182	678	732	101	573	290	156	272	207	631	607	701
177	445	155	365	545	73	38	583	284	80	560	138
591	469	559	97	423	398	119	208	603	527	512	514
767	78	12	173	490	95	324	126	619	349	69	488
665	662	460	693	459	180	420	604	509	720	598	595
529	174	5	50	87	271	683	561	696	422	188	59
245	620	439	482	240	580	361	385	685	593	755	142
358	274	415	497	571	698	677	370	536	679	31	663
641	219	468	736	375	222	189	757	432	158	216	639
471	413	633	455	23	85	642	407	168	651	675	238
206	450	123	146	465	518	276	739	546	319	199	145
611	106	689	636	688	714	461	768	348	632	214	133
442	333	46	129	118	344	654	10	68	722	627	379
581	575	758	730	390	661	411	682	172	249	334	14
652	695	11	336	576	487	278	725	268	384	61	630
191	520	211	521	239	589	756	26	286	409	417	127
227	501	55	298	426	713	735	35	148	159	437	401
186	330	99	638	618	550	410	226	15	44	40	430
742	103	140	458	111	369	449	373	537	202	615	83
270	674	472	600	564	608	397	645	215	234	721	130
261	153	269	515	579	316	405	741	621	727	628	656
254	354	352	122	72	763	76	287	34	752	181	446
544	21	236	566	441	524	617	300	513	79	248	467
115	731	474	408	704	491	305	547	557	377	20	427
719	697	102	357	52	346	526	709	582	185	718	149
542	481	7	716	318	761	606	312	288	201	429	109
549	493	338	285	378	454	86	438	613	419	601	329
303	293	712	110	728	247	433	157	443	282	212	218
197	634	694	746	489	135	558	623	717	680	246	648
264	664	366	574	676	124	362	36	53	8	213	200
530	668	114	555	543	485	650	498	233	42	150	325
166	98	511	343	749	738	414	552	753	251	250	255
356	280	386	266	551	28	273	328	93	499	275	92
486	108	198	534	32	532	194	70	569	217	691	658
406	578	16	263	327	2	221	496	765	463	332	681
616	500	25	428	554	479	306	67	637	584	371	190
692	473	258	60	759	220	425	440	671	523	167	225
210	355	684	602	179	687	376	193	686	624	504	241



(a) ***(b)*** ***(c)***

(d)

(e)

Figure IV.7. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée*
(d) *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

92	39	9	38	13	16	45	76	98	47	24	8
83	40	3	78	80	90	61	26	68	89	53	52
19	31	50	18	12	14	54	64	2	46	58	65
35	71	5	28	74	36	44	73	79	99	96	62
63	94	29	15	37	11	75	10	30	43	7	51
91	55	33	66	57	23	81	41	70	56	77	88
4	95	25	1	69	86	22	82	97	67	42	59
20	87	32	49	72	48	27	425	652	573	441	543
468	129	748	420	738	513	353	453	569	700	185	525
707	143	512	264	237	198	755	658	487	221	741	557
484	647	227	257	708	162	179	579	197	724	60	326
629	757	548	220	413	115	266	258	120	443	218	639
253	669	673	260	712	202	345	430	225	561	284	449
481	109	294	661	248	552	710	354	433	644	372	630
161	684	648	21	476	432	110	489	331	503	403	111
507	363	637	680	17	734	556	588	438	692	666	367
753	723	613	287	34	604	505	168	273	267	421	564
239	542	742	382	642	471	508	278	614	333	359	234
136	764	285	765	545	672	259	114	214	572	170	520
147	458	502	352	244	659	423	688	537	175	386	706
609	678	328	149	272	462	393	567	379	6	144	330
523	145	562	417	119	761	767	226	478	355	726	518
737	722	583	127	440	238	397	429	339	247	173	450
296	635	254	378	341	213	488	134	409	292	731	427

399	494	193	280	314	93	704	123	469	148	634	657
303	344	501	705	404	733	171	627	662	551	311	470
743	510	401	157	554	346	491	340	698	256	240	347
125	611	727	589	358	568	374	325	631	307	361	745
308	756	746	164	369	131	306	448	640	245	565	118
586	668	167	580	313	553	758	189	84	124	350	262
645	703	676	574	540	255	348	575	515	286	128	664
398	366	531	113	217	456	300	681	224	223	593	699
691	422	585	506	402	434	641	270	528	446	426	696
301	656	690	302	529	151	682	204	735	571	416	663
165	108	514	435	444	689	190	290	158	602	140	335
299	582	174	677	516	616	203	736	104	454	536	625
279	577	396	495	312	750	310	349	559	205	315	141
728	269	192	250	208	695	718	709	740	618	687	587
550	283	322	142	209	282	199	187	276	336	653	229
321	176	730	626	714	342	327	219	599	139	608	112
752	566	603	581	390	210	466	373	650	576	152	447
233	163	384	196	242	362	768	180	188	570	461	601
126	216	395	365	497	309	751	201	230	739	595	177
178	249	133	605	766	116	207	394	667	318	281	194
530	600	117	376	235	636	623	646	85	732	598	555
437	241	693	334	222	100	293	612	406	184	492	563
560	547	534	679	407	215	535	351	674	332	475	670
271	212	275	320	558	749	721	418	181	186	633	371
701	591	385	517	533	467	638	686	490	590	364	375
154	606	683	388	762	759	200	474	729	649	405	360
500	166	655	389	343	538	628	493	524	596	483	594
527	172	243	549	754	304	473	415	477	103	597	615
464	295	725	368	412	195	277	231	713	632	297	480
211	442	459	620	532	660	651	319	410	544	485	419
445	744	101	252	191	121	298	486	265	439	182	463
150	452	392	457	400	146	106	747	526	357	370	665
317	482	763	377	720	451	578	621	455	356	324	521
130	288	228	156	323	122	504	643	246	522	496	619
436	697	694	675	431	316	584	232	607	107	381	671
719	261	428	411	408	498	105	159	717	610	539	715
153	268	546	617	387	102	160	414	685	716	465	460
338	511	391	622	289	291	479	138	702	305	711	424
654	155	499	169	329	137	183	251	383	132	592	472
206	263	541	135	760	509	274	624	337	380	519	236

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

528	192	38	202	98	315	156	142	97	339	304	457
569	700	485	538	200	292	180	544	443	684	130	287
352	689	571	468	536	101	169	244	639	353	215	24
735	501	545	266	628	575	359	745	653	1	694	535
411	37	9	342	491	497	146	278	620	88	533	80
151	380	298	137	521	508	409	61	123	191	728	662
558	115	398	514	454	50	323	541	429	410	686	488
228	232	247	559	378	234	181	261	233	345	243	643
758	387	384	33	8	761	412	346	371	695	750	40
308	477	128	179	487	493	294	604	11	504	129	710
68	39	196	739	725	753	43	63	187	73	26	749
401	578	162	176	404	395	699	302	140	270	157	272
563	341	249	81	381	92	652	58	673	289	67	712
547	768	94	322	172	218	241	69	721	438	618	645
242	723	611	556	635	688	297	193	641	474	553	62
654	25	502	310	301	148	269	428	377	70	106	592
672	617	258	439	708	658	34	737	274	407	113	112
279	12	209	119	326	531	207	677	584	42	282	330
199	237	268	648	251	731	447	338	744	685	126	557
216	610	621	483	328	467	205	674	184	259	44	210
649	532	75	600	114	625	473	290	334	83	596	752
256	434	470	82	650	425	141	482	763	534	116	201
354	385	96	549	601	726	576	717	283	173	730	212
707	525	522	373	452	281	348	117	227	318	589	565
208	663	214	505	711	609	276	93	526	31	767	23
636	687	667	372	570	303	581	369	252	510	343	546
540	219	177	418	182	722	760	495	171	656	255	78

329	691	608	293	580	690	229	27	262	513	554	223
296	22	152	432	57	333	159	211	347	516	665	675
727	676	124	450	713	299	564	86	679	4	757	226
311	335	393	402	161	217	702	102	741	54	260	397
637	213	118	548	230	32	364	622	382	111	254	379
568	107	174	165	366	195	417	153	759	512	697	764
367	696	742	76	597	394	435	158	480	361	103	149
120	448	706	755	716	178	166	520	602	537	422	740
682	577	327	357	475	168	560	585	598	436	35	413
110	189	414	250	127	552	358	332	403	190	668	41
587	424	122	588	567	391	612	17	376	416	631	291
355	175	307	222	456	154	340	66	427	586	331	238
542	84	370	484	295	566	183	471	437	236	529	583
593	52	459	319	614	271	449	613	131	698	55	6
313	19	280	60	550	325	349	59	46	714	396	671
453	45	605	245	337	105	85	350	469	167	664	400
669	351	623	660	197	720	591	594	224	16	246	51
421	562	692	2	724	478	388	442	666	53	3	320
419	659	132	91	383	423	235	121	71	462	461	481
543	20	147	160	515	415	316	87	527	203	300	754
286	138	681	288	15	284	579	511	143	704	524	65
509	632	198	574	64	163	29	455	231	139	624	733
638	539	765	275	309	680	472	263	108	321	463	607
640	693	523	135	630	619	701	389	683	627	732	170
709	99	603	420	426	312	441	374	440	392	747	77
719	136	661	317	756	368	406	48	476	489	305	748
109	715	465	615	285	496	375	155	21	446	519	408
561	18	498	220	734	336	5	530	188	762	492	399
464	13	518	555	499	766	433	431	204	458	273	751
248	74	89	479	186	100	705	506	551	500	405	572
28	267	14	743	444	164	150	240	430	362	655	257
616	265	657	185	494	703	644	642	56	134	646	390
445	104	678	647	729	718	363	466	306	599	629	517
360	10	507	746	225	736	595	47	253	651	194	95
582	633	490	486	451	344	606	738	7	634	277	386
503	144	324	30	670	36	365	125	145	356	79	264
314	590	90	626	460	573	206	133	49	239	72	221

Le tableau suivant résume les résultats relatifs au temps de calcul (temps de chiffrement), la taille de clé, la quantité de phéromone et l'efficacité dans le cas de chiffrement des différentes données.

			Taille donnée (éléments)	**Taille clé (bits)**	**Efficacité**	**Temps calcul (s)**	**Phéromone**
Données Texte	Texte1	Version-Chiff1	1084	15323	11784.0	5.63	9,06
		Version-Chiff2			12030.0	6.02	9,13
	Texte2	Version-Chiff1	1151	15323	14498.0	6.7	8,67
		Version-Chiff2			14712.0	6.58	8,65
Données Images	Lena	Version-Chiff1	131 X 131	7680	135362.0	2.16	2.75
		Version-Chiff2			130074.0	2.31	1.17
	Logo	Version-Chiff1	420 X 395	7680	244926.0	2.87	23.11
		Version-Chiff2			226188.0	2.84	21.81

Tableau IV.2. *Résultats obtenus par AntCrypt.*

IV.4. Discussion et évaluation des résultats

La première chose à remarquer, d'après les résultats résumés à travers le tableau IV.2, c'est que le temps de chiffrement des données texte est beaucoup plus grand que celui des données images malgré que la taille des données images est généralement plus grande en comparaison à celle des données texte. Cela revient au fait que la taille du vecteur codant les données texte (1393 éléments) est beaucoup plus grande que celle du vecteur codant les données images (768 éléments), ce qui nécessitera beaucoup plus de temps de calcul lors de l'application du processus de résolution à base de fourmis. Toutefois, le temps de chiffrement est raisonnable pour les deux types de données texte ou images.

Nous remarquons, aussi, que les temps de calcul des différentes versions d'une même donnée sont proches, car ils sont indépendants de la taille de la donnée à chiffrer (une différence apparaît quand la dissemblance entre les tailles de deux données de même type sera importante et qui représente un temps de lecture ou de codage). En effet, la taille du vecteur codant toutes les données de même type et soumises au chiffrement est la même (1393 éléments pour le cas de texte / 768 éléments pour le cas d'images), donc sa manipulation sur l'ensemble des générations prend le même temps. La toute petite différence correspond au temps de codage des données avant le lancement du processus de chiffrement. Toutefois, la légère différence entre le temps de chiffrement des deux versions d'une même donnée revient au caractère aléatoire exploité dans certaines phases de l'algorithme.

Pour le cas de chiffrement des deux données images Lena et Logo, nous constatons que le temps de chiffrement de l'image Logo est plus grand que celui de l'image Lena. La différence est due au fait que le temps de chiffrement est la somme des temps de lecture puis codage de l'image qui varie suivant la taille de l'image, plus le temps de manipulation du vecteur codant les solutions qui n'a aucune relation avec la taille de l'image. Donc, la différence en temps de chiffrement des deux images est justifiée.

La valeur de la phéromone dépend de la valeur d'efficacité maximale et de l'efficacité de la solution calculée. Il est bien clair que pour de grandes valeurs d'efficacité, une amélioration de la quantité de phéromone sera marquée. Prenons l'exemple du premier message où pour la première version chiffrée nous avons obtenu 2166 / 9,06 pour efficacité / phéromone. Cependant, pour la deuxième version chiffrée, nous avons eu 2184 / 9,13. Donc, l'amélioration de l'efficacité est certainement accompagnée d'une amélioration de la quantité de phéromone. Il sera utile de rappeler, ici, que l'évaporation est la cause de diminution de la quantité de phéromone.

En mesurant le taux de différence entre les deux versions chiffrées de l'image Lena et l'image originale (l'image Lena), le tableau suivant récapitule les résultats obtenus en termes de NPCR, MAE et MSE. Il est clair que les deux versions chiffrées sont presque totalement différentes de l'image originale. Autrement dit, les pixels formant les versions chiffrées sont presque tous changés, soit de positions, soit de nombres d'occurrences et de positions ; ce qui donnera lieu à un très bon niveau de confusion.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.9207	1.08	0.73
Lena-version chiffrée 2	0.9082	1.03	0.69

TableauIV.3. *Niveaux de confusion de AntCrypt.*

Maintenant comparant AntCrypt avec le plus rapide de nos trois algorithmes développés et présentés dans le précédent chapitre, bien sûre sur le plan de temps de calcul. Le tableau suivant rappelle les temps de convergence des deux algorithmes OEEA et AntCrypt. Ce dernier a montré un temps de calcul meilleur que celui de OEEA que ça soit pour chiffrer des données textes ou images malgré qu'aucune explication exacte ne peut être donnée du fait que les deux algorithmes utilisent un même mode de chiffrement (chiffrement à base d'occurrences), donc un même codage qui est exploité par les composants de la métaheuristique impliquée (opérateurs génétiques et mécanismes de la sélection naturelle pour OEEA et fourmis et mécanismes de manipulation de phéromone pour AntCrypt). Et vu que le temps de convergence de notre algorithme de chiffrement basé sur les mécanismes évolutionnaires (OEEA) est légèrement plus grand que celui basé sur les fourmis (AntCrypt), donc possible que les premiers mécanismes consomment plus de temps de calcul que les deuxièmes, mais aucune preuve n'est connue !

	Temps calcul (s)	
	Texte 1	Lena
AntCrypt	5.825	2.235
OEEA	6.54	3.37

Tableau IV.4. *Temps de calcul de AntCrypt en comparaison avec le temps de calcul de OEEA.*

Remarque :

Dans le tableau IV.4, les temps de calcul de l'algorithme AntCrypt pour les deux cas de données (Texte 1 et Lena), représentent la moyenne des temps de calcul des deux versions chiffrées (Voir tableau IV.2).

De même et vu que la méthode de chiffrement présentée dans ce chapitre (Cryptage à base de fourmis) et celle présentée dans le précédent chapitre (Cryptage évolutionnaire) appartiennent à une même catégorie mère qui est celle de métaheuristiques, donc toute les deux vont bénéficier de quelques caractéristiques cryptanalytiques en commun. Il s'agit de l'aspect non déterministe démontré à travers les résultats présentés ci-dessus où deux versions chiffrées différemment correspondent toutes les deux à une même donnée originale (voir figures IV.4, IV.5, IV.6 et IV.7), et d'une robustesse contre les types d'attaques les plus avancées (attaque exhaustive, attaque statistique, attaque fréquentielle et attaque différentielle) comme c'était démontré dans la section III.5 du chapitre précédent.

IV.5. Conclusion

ACO est une méthode stochastique qui requiert de plus en plus l'attention de la communauté scientifique. Cette métaheuristique a prouvé sa performance pour plusieurs problèmes d'optimisation combinatoire. En effet, l'utilisation des traces de phéromone permet d'exploiter l'expérience de recherche acquise par les fourmis et ainsi renforcer l'apprentissage pour la construction de solutions dans les itérations futures de l'algorithme. En même temps, l'information heuristique peut guider les fourmis vers les zones prometteuses de l'espace de recherche.

Ainsi et dans ce chapitre, nous avons présenté, interprété et discuté les résultats expérimentaux obtenus par l'application de notre algorithme de chiffrement proposé *AntCrypt* sur des données modèles texte et images. Nous sommes arrivés à fixer nos paramètres par expériences, en choisissant 40 générations et 12 fourmis en essayant de réaliser un compromis entre l'efficacité et le temps de chiffrement.

L'algorithme, ainsi développé et qui a été nommé *AntCrypt*, a été comparé avec nos algorithmes développés à base d'algorithmes évolutionnaires où il a montré de bonnes caractéristiques telles qu'un bon niveau confusionnel, un aspect non déterministe, une taille sécurisée de clés, lui offrant une bonne résistibilité contre les attaques les plus avancées. En plus le temps de convergence de cet algorithme est meilleur que celui du plus rapide de nos trois algorithmes proposés dans le précédent chapitre.

CHAPITRE V

CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

V.1. Introduction

Nous nous sommes intéressés, dans les deux chapitres précédents, à la résolution du problème de cryptage de données texte et images par application de métaheuristiques à populations. Les approches que nous avons proposé sont caractérisées par leur qualité de confusion, leur temps d'exécution réduits pour certains d'eux (OEEA et AntCrypt) et leur robustesse face aux attaques avancées.

Dans ce chapitre, nous nous intéressons à l'exploitation d'une métaheuristique appartenant à une autre catégorie, les métaheuristiques à trajectoire. Il s'agit de la recherche tabou. Cette dernière s'appuie sur une recherche locale combinée à un mécanisme de prévention des cycles, grâce à un système de mémoire des mouvements précédemment appliqués ou des configurations visitées (la liste tabou).

De manière générale, une recherche locale démarre d'une solution initiale possible et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant. Un voisinage d'une certaine solution correspond à des éléments adjacents à cette solution dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration dans la solution courante ne puisse être faite.

Nous présentons, après la description de l'approche proposée, les résultats numériques obtenus afin de les comparer aux résultats obtenus par nos autres méthodes proposées (cryptage évolutionnaire et cryptage par colonies de fourmis) et certaines autres méthodes de la littérature.

V.2. Motivation

La recherche Tabou est une métaheuristique basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

L'originalité de la méthode de recherche tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objective f et l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum

quelconque par acceptation de la dégradation de la fonction objective, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste d'une certaine longueur, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

V.3. Algorithme proposé

À partir d'une solution initiale, le principe général de la recherche tabou est le suivant (voir Algorithme V.1). À chaque itération de la recherche, une partie ou l'ensemble des voisins de la solution est exploré, et un des mouvements parmi ceux de coût minimal est sélectionné. Ce mouvement est appliqué quel que soit son coût, à la condition de ne pas créer un cycle dans le processus de recherche locale à court terme (en menant à une configuration dont les caractéristiques sont stockées dans la liste tabou). Une exception est faite (critère d'aspiration) lorsque le mouvement permet d'atteindre une solution de meilleure qualité que la meilleure solution enregistrée à ce stade. La liste tabou est mise à jour à chaque itération de la recherche en fonction des mouvements choisis. Ce mécanisme permet de sortir des minima locaux en acceptant des mouvements détériorants et en empêchant parallèlement le retour immédiat à la solution de minimum local qui vient d'être quittée.

Algorithme V.1 *Schéma général d'un algorithme tabou*

```
Engendrer une configuration initiale s
s* ← s
T ← ∅ liste tabou
tant que condition d'arrêt non satisfaite faire
    m ← meilleur mouvement (i.e. minimisant f) parmi ceux non tabou
    ou ceux vérifiant un critère d'aspiration
    Modifier s en effectuant le mouvement m
    Mettre T à jour
    si f(s) < f(s*) alors
        s* ← s
    fin
fin
retourner s*
```

La succession des étapes suivantes montre plus de détails du schéma du mécanisme utilisé pour la construction d'une solution au problème étudié :

1) Détermination d'une solution initiale $S_{(0)}$ codant l'image originale.
2) Création puis la mise à jour d'une liste tabou qui mémorisera les déplacements effectués.
3) Choix aléatoire d'un nombre d'emplacements ou d'éléments de la solution courante ($i^{ème}$ solution) à déplacer.
4) Génération aléatoire d'un ensemble de voisins pour chacun des éléments choisis dans l'étape précédente.
5) Choix du meilleur voisin de l'ensemble généré.
6) Calcul tabou de la solution suivante $S_{(i+1)}$.
7) Création et la mise à jour d'une table de hachage qui comportera les dernières meilleures solutions obtenues.
8) Évaluation de la solution calculée $S_{(i+1)}$ obtenue à chaque itération.
9) Validation de la solution calculée $S_{(i+1)}$.
10) Vérification du critère d'arrêt de l'algorithme.

En ce qui suit, nous décrivons les principales étapes de l'algorithme de chiffrement tabou proposé et que nous l'avons appelé *TabuCrypt*.

V.3.1. Création de la solution initiale

La solution initiale exploitée par *TabuCrypt* s'obtient en codant les données originales suivant le même mode de codage adopté par *OEEA* ou *AntCrypt*. C'est d'ailleurs le même codage utilisé pour modéliser toutes les autres solutions.

V.3.2. Choix des éléments à déplacer

Les solutions sont représentées soit, par un vecteur englobant 1393 éléments pour le cas de manipulation de données texte soit, par un autre regroupant 768 éléments pour le cas de manipulation de données images. *TabuCrypt* cherche à réarranger aléatoirement le vecteur correspondant à la solution initiale codant la donnée originale en permutant les éléments entre eux. Toutefois, la taille du vecteur est assez grande (1393 éléments ou même 768 éléments), c'est pourquoi la manipulation un par un de ses éléments sur l'ensemble des itérations de l'algorithme nécessitera un grand temps de calcul. Ainsi, sauf un sous ensemble du vecteur globale sera manipuler pour déplacement. Cela permet, d'un coté, d'optimiser le temps de calcul et, d'un autre coté, de converger petit à petit vers la solution finale en évitant, ainsi, le problème de convergence prématurée.

V.3.3. Génération de voisinage

Le voisinage est le responsable sur le fait de pousser l'algorithme à l'amélioration de la qualité des résultats. Dans notre cas, il est généré de la manière suivante :

Pour chacun des éléments du sous-ensemble construit dans l'étape précédente, nous générons aléatoirement un ensemble de voisins candidats au déplacement avec l'élément en question. Lors de l'élection d'un voisin, les conditions suivantes doivent être vérifiées :

- ✓ Un voisin ne doit pas être élu plus qu'une fois, c'est-à-dire, tous les voisins de l'ensemble de voisins sont différents les uns des autres.
- ✓ Les voisins élus n'appartiennent pas dans la liste tabou.

Une fois l'ensemble des voisins construit, nous choisissons celui qui diffère le plus de l'élément à déplacer. Formellement, soit E_V l'ensemble de n voisins V, et E_i l'élément candidat au déplacement. La sélection du voisin v avec lequel l'élément E_i sera permuté se fait comme suit :

$$v = V_j \,/\, j = \overline{1,n} : Max\left(\left|E_i - V_j\right|\right) \qquad \text{(V.1)}$$

V.3.4. Mise à jour de la liste tabou

Dans notre cas et à une itération donnée, la liste tabou regroupe les voisins qui ont participé à des déplacements au cours des itérations précédentes avec l'un des éléments du sous-ensemble construit dans l'étape 2 décrite dans la section V.3.2. Ainsi, la mise à jour de cette liste (liste tabou) se fait à chaque fois qu'un élément sera permuté avec un voisin choisi suivant l'étape 3 décrite dans la section V.3.3 en ajoutant ce dernier à la liste des éléments tabous. A la fin de chaque itération, le contenu de la liste tabou sera effacé.

La politique utilisée pour rendre tabous certains éléments évite la modification des nombres d'occurrences originaux, puisque le processus de calcul de la solution ne doit pas modifier les éléments du vecteur initial mais plutôt il cherche à changer leur dispersion.

V.3.5. Calcul de solutions

Au cours d'une itération donnée, le calcul de la solution proposée à cette itération se base sur la solution calculée à l'itération précédente. En effet, des éléments du vecteur présentant la solution à l'itération précédente échangeront leurs positions avec celles des voisins désignés pour chacun d'eux. Cette nouvelle dispersion des éléments forme le nouveau vecteur solution. Il est à noter que les éléments non élus pour un déplacement garderont leurs positions sans modification.

V.3.6. Mise à jour de la table de hachage et évaluation des solutions

La table de hachage sert à mémoriser les différents points visités de l'espace de recherche en gardant les différentes solutions calculées sur l'ensemble des itérations. Le but sera d'interdire la recherche à revenir vers des points déjà visités à travers les précédentes itérations. Ainsi, une solution générée à la fin d'une certaine itération sera comparée au contenu de la table de hachage avant qu'elle soit soumise à une validation finale en vu de l'accepter comme intéressante solution à partir de laquelle la recherche peut continuer sur l'ensemble des itérations restantes. Si cette solution représente un nouveau point de l'espace non encore visité, elle sera ajoutée à la table de hachage pour quelle fera l'objet des futurs tests. Cette politique sert à échapper aux minima locaux et à favoriser l'exploration de nouveaux points de l'espace de solutions possibles.

La validation définitive d'une solution S_i calculée à une itération i donnée, consiste tout d'abord, à l'évaluer. Dans notre cas, l'évaluation se fait suivant la fonction d'évaluation illustrée par la formule suivante, notant que S_0 soit la solution initiale (voir étape 1).

$$F(S_i) = \sum_{j=1}^{l} \left| S_{i_j} - S_{0_j} \right| \qquad \text{(V.2)}$$

Où l : représente le nombre de valeurs possibles que peut prendre un élément d'une donnée (égale à 1393 pour le cas de données texte et 768 pour le cas de données images).

Une fois l'évaluation faite, la qualité de la solution en question sera comparée à celle de la solution calculée à l'itération précédente. Si elle est meilleure que la précédente, la recherche à travers la prochaine itération va se continuer à partir du point représenté par cette solution, sinon elle sera rejetée et la recherche continuera à partir de la solution calculée à l'itération précédente.

Ces points peuvent être résumés comme suit : à chaque itération la solution calculée est ajoutée à la table de hachage puis elle sera examinée selon les trois critères suivants :

1) Si la solution S_i obtenue n'est pas une nouvelle solution alors elle sera rejetée et les étapes de 2 à 6 sont à refaire.

2) Si la solution S_i obtenue est une nouvelle solution mais plus mauvaise que celle de l'itération *i-1* (S_{i-1}), de même elle est rejetée, mais sauvegardée dans la table de hachage, et les étapes de 2 à 6 sont à refaire.

3) Si la solution S_i obtenue est une nouvelle solution mais, cette fois ci, est meilleure que celle de l'itération (i-1), alors elle sera retenue pour continuer la recherche à partir de la prochaine itération et, ainsi, elle servira à calculer la solution suivante S_{i+1}. En même temps, elle sera mémorisée dans la table de hachage.

Remarque :
Dans notre algorithme, on considère comme *mouvement global* le passage d'une solution courante S_i vers une solution suivante S_{i+1}, en respectant les conditions citées précédemment. Toutefois, les *mouvements élémentaires* sont les déplacements des éléments et leurs voisins vers leurs positions mutuelles. Ainsi, un mouvement global émerge comme résultat des mouvements élémentaires.

V.3.7. Critère d'arrêt

Pour toute recherche itérative un critère d'arrêt est obligatoire pour éviter le problème de boucles infinies. Ce dernier qui doit être fixé à l'avance permet de déterminer le nombre de fois que la tâche à exécuter sera répétée.

Pour notre algorithme, nous avons choisi de fixer, expérimentalement, le nombre d'itérations. Ce choix influe directement sur longueur du temps de calcul et de la qualité de la solution construite.

V.3.8. Mécanismes avancés en recherche tabou

V.3.8.1. Intensification / Exploitation

L'intensification consiste à approfondir la recherche dans certaines régions de l'espace, identifiées comme susceptibles de contenir un optimum global.
Pour notre cas, les mouvements globaux en plus des mouvements élémentaires serviront à mieux intensifier la recherche. En effet, les mouvements élémentaires traduisent les déplacements des éléments d'un vecteur solution vers les positions de leurs voisins et vise versa. Ces voisins sont les éléments optimaux (les meilleurs voisins) de l'ensemble des voisins candidats au déplacement (voir étape 3). Donc, nous cherchons à exploiter les qualités des éléments. De même, les mouvements globaux cherchent à exploiter les qualités des solutions (les vecteurs d'éléments) en n'acceptant de poursuivre la recherche qu'à partir des solutions améliorantes lors du passage d'une itération à l'itération qui suit.

V.3.8.2. Diversification / Exploration

La diversification vise à utiliser des mouvements encore jamais réalisés afin d'explorer de nouvelles régions de l'espace de recherche et des régions éloignées du voisinage actuel. Dans notre cas, cette caractéristique est accomplie par la sélection aléatoire d'un nouveau voisinage non tabou pour chaque élément.

V.3.8.3. Critère d'aspiration

Le critère d'aspiration autorise un mouvement tabou sous certaines conditions. Il consiste à tester si la solution produite de statut tabou présente un coût inférieur ou de qualité meilleure que ceux de la meilleure solution trouvée jusqu'à présent. Si c'est le cas, le statut tabou de la solution est levé.

Dans notre cas, la notion de tabou est exploitée lors de la construction d'une certaine solution en interdisant les mouvements élémentaires vers les éléments voisins avec lesquels les éléments du sous-ensemble construit comme l'indique l'étape 2 ont échangé de positions. Ainsi, le mécanisme d'aspiration n'a pas de place dans notre méthode du fait que toute libération d'un élément de la liste tabou entraîne une modification des éléments des vecteurs et, par conséquent la modification des caractéristiques servira à calculer ultérieurement la donnée déchiffrée, tandis que notre but est, plutôt, modifier la répartition des éléments des vecteurs.

V.3.9. Déchiffrement

L'opération inverse au chiffrement est le déchiffrement qui permet de regénérer la donnée originale précédemment chiffrée par notre algorithme de chiffrement tabou *TabuCrypt*. Pour ce faire, ce processus exploite une information secrète calculée lors du chiffrement à coté de l'image chiffrée. Il s'agit d'une clé de session secrète. Elle représente les permutations des positions des l éléments (1393 éléments ou 768 éléments) du vecteur codant une certaine donnée à travers les différentes itérations. Une fois elle parviendra sans modification au destinataire approprié, elle servira à calculer la donnée originale.

V.3.10. Réglage des paramètres et résultats

Le schéma général de l'algorithme finalement implémentant les étapes de l'approche de cryptage tabou proposé est le suivant :

Algorithme V.2 *TabuCrypt*

Étape 1 : Générer une solution initiale de gain total G_0.
Étape 2 : Initialiser
- le nombre d'itération iter (à 1),
- la table de Hachage H (ajouter la solution initiale à H),
- la liste tabou T (vide).

Répéter les étapes de 3 à 11 jusqu'à iter = iterMax.
Étape 3 : Initialiser le compteur d'éléments (composants d'une solution)
Répéter les étapes de 4 à 6 jusqu'à
Étape 4 : Sélectionner aléatoirement un ensemble de voisins des éléments non tabous, puis choisir le meilleur de l'ensemble.
Étape 5 : Déclarer comme tabou le voisin choisi (voisin ajouté à T).
Étape 6 : Déplacer l'élément courant et le voisin choisi vers leurs positions mutuelles.
Étape 7 : Evaluer la nouvelle solution calculée pour mesurer son gain G_{iter}.
Étape 8 : Si : la solution calculée est déjà contenue dans H *alors :* Solution ignorée, *Sinon :* Ajouter la solution à H
Étape 9 : Si $G_{iter} > G_{iter-1}$ *alors* continuer le calcul à partir de $Solution_i$ *Sinon* Continuer le calcul à partir de la $solution_{iter-1}$
Étape 10 : Vider T.
Étape 11 : iter + +.

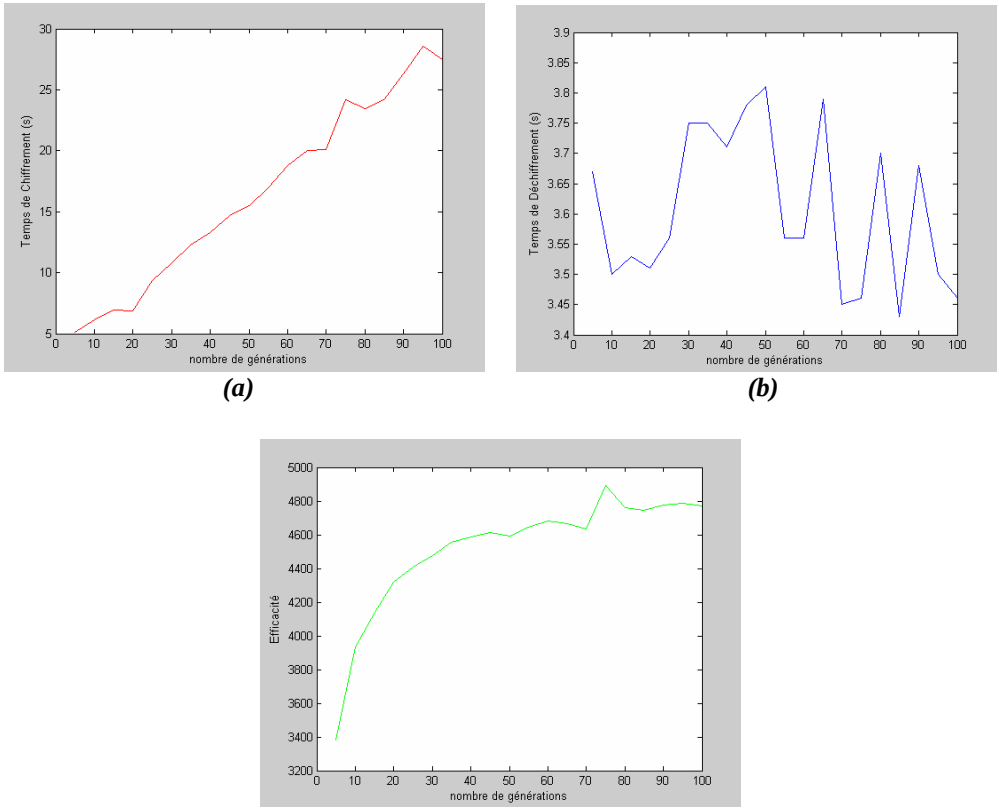
Nous précisons, dans cette section, les valeurs des paramètres de *TabuCrypt* ainsi que les résultats de son application sur les données modèles précédemment présentées (Texte1, Texte2, Image Lena et image Logo).

V.3.10.1. Réglage des paramètres

Les principaux paramètres de *TabuCrypt* concernent le nombre de générations ou d'itérations de l'algorithme et le nombre de voisins seront présentés.

Après *IterMax* itérations, fixée expérimentalement, la procédure de recherche tabou s'arrête et fournit la meilleure solution trouvée. La figure V.1 récapitule la moyenne des différentes valeurs de test et leurs impacts sur le temps de convergence et la qualité de la solution trouvée représentant une version chiffrée de l'image de test Lena.

D'après les résultats résumés dans la figure, ci-dessous, nous constatons que la meilleure efficacité a été obtenue pour un nombre de générations égale à 75 après un temps de chiffrement de 24.20 s et un temps de déchiffrement de 3.46 s. De même, les nombres de générations 80, 85, 90, 95 et 100 ont donné de bonnes valeurs d'efficacité (4760, 4744, 4777, 4789 et 4771, respectivement), mais leurs temps de chiffrement sont assez longs (23.46 s, 24.18 s, 26.29 s, 28.54 s et 27.48 s, respectivement).



(a) ***(b)*** ***(c)***

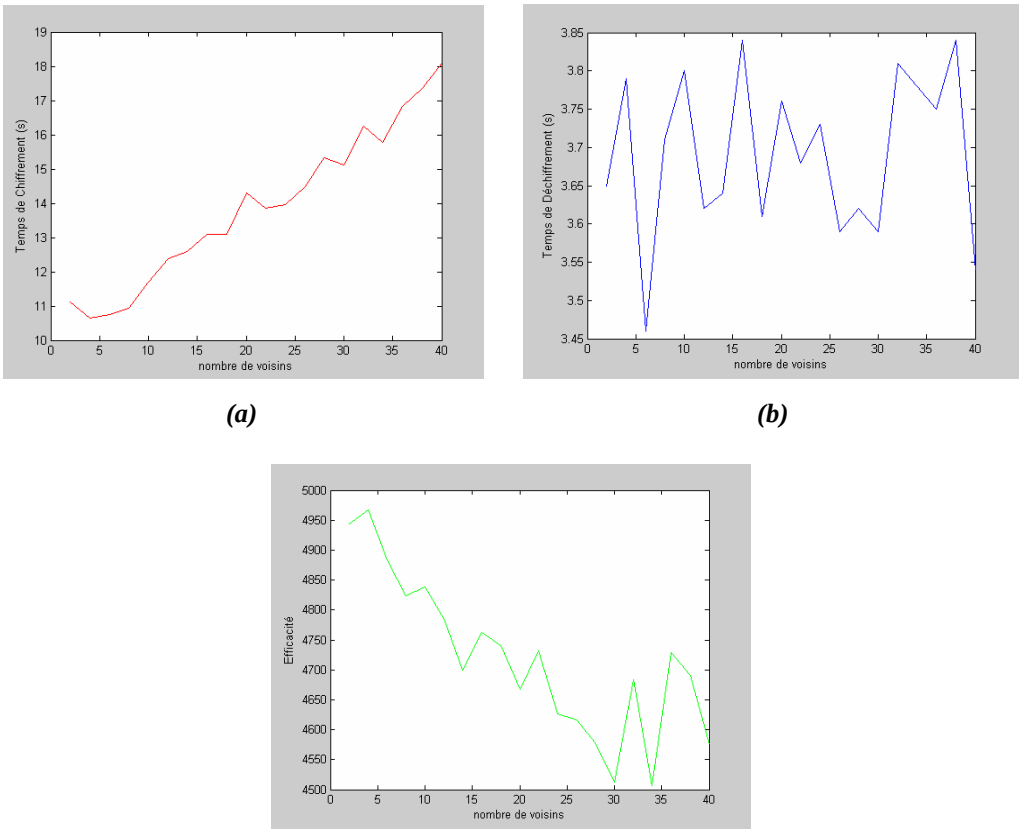
Figure V.1. *Résultats obtenus suivant les différentes valeurs de test de nombre d'itérations (générations) :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Toutefois, les résultats obtenus, pour le test avec 45 générations, sont acceptables que se soit sur le plan efficacité (4612) ou temps de chiffrement (14.70 s). D'autres valeurs de nombre de générations (55, 60, 65 et 70) ont donné des résultats d'efficacité de même ordre que celle du nombre de générations 45, mais après des temps de chiffrement relativement plus longs (29.93, 28.76, 20.04 et 20.12, respectivement). Ainsi, la valeur de *IterMax* a été fixée à 45 générations.

L'autre paramètre à régler est le nombre de voisin. Pour ce faire, nous avons utilisé le même mécanisme que celui utilisé pour régler le paramètre relatif au nombre d'itérations. Les résultats des différents tests sont résumés à travers la figure V.2 en donnant l'influence des différents nombres de voisins testés sur le temps de convergence et l'efficacité de la solution trouvée.

D'après le contenu des figures V.1 et V.2, la meilleure efficacité (4967) a été obtenue avec un nombre de voisins qui vaut 4 après un temps de chiffrement assez raisonnable et représentant le plus petit temps de calcul sur l'ensemble des tests effectués (10.64s).

Ainsi, nous avons opté à régler le paramètre relatif au nombre de voisins en lui attribuant la valeur 4.



(a) ***(b)***

(c)

Figure V.2. *Résultats obtenus suivant les différentes valeurs de test de nombre de voisins :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Le tableau suivant (tableau V.1) résumé les valeurs de paramètres adoptés par *TabuCrypt.*

	Valeurs des paramètres de*TabuCrypt*
Nombre de voisins	4
Nombre de générations	45

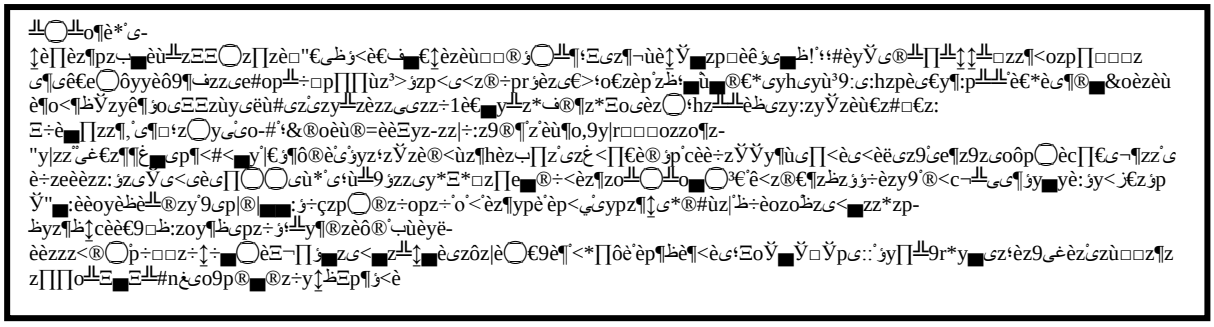
Tableau V.1. *Valeurs adoptées pour les paramètres de TabuCrypt.*

V.3.10.2. Résultats

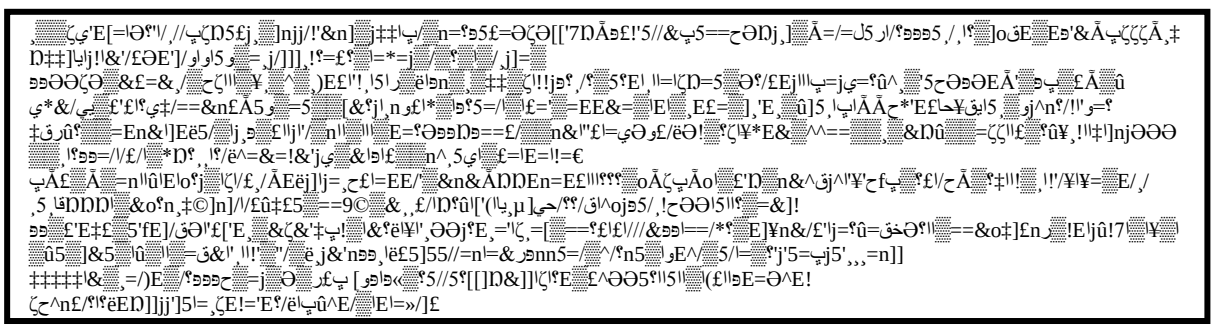
A ce niveau nous présentons les résultats de chiffrement par *TabuCrypt* des données tests. Deux versions chiffrées de chacune des données tests sont présentées accompagnées des clés générées dans chaque cas. Le tableau V.2 résume les temps de chiffrement et de déchiffrement ainsi que l'efficacité de chacun des exemples.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure V.3. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets)

728	898	353	668	901	875	906	437	425	40	739	524
797	890	877	786	357	654	889	258	566	884	883	899
399	721	896	133	905	483	410	895	735	117	900	876
878	303	479	907	706	879	100	813	91	453	152	634
874	186	597	902	480	341	99	42	736	438	456	893
794	344	881	267	887	474	617	191	885	56	892	886
897	377	601	904	34	98	672	880	903	673	496	760
373	882	894	888	347	463	750	891	520	248	465	571
624	116	573	744	866	646	926	68	1228	131	311	532
299	518	139	1134	1120	734	714	290	22	97	541	1047
1234	961	1110	819	52	153	255	1006	560	960	1077	942
1109	154	836	30	412	1119	1202	1037	1003	383	1238	335
392	1035	1241	119	448	409	1115	266	1036	857	971	194
713	912	422	378	500	782	274	1048	1102	651	1129	1091
810	696	1040	994	321	362	46	563	1148	149	339	858
503	328	110	770	1053	773	272	598	77	1087	1187	1121
551	418	225	801	1185	1082	955	557	737	506	464	1070
1240	168	187	843	796	82	867	450	200	764	190	982
1076	967	873	260	695	224	1018	207	1049	277	3	652
792	1014	1213	936	466	621	829	2	400	1085	38	23
1159	570	509	816	973	164	205	1201	112	1078	870	333
364	1212	350	250	102	536	172	629	705	178	381	722
283	1100	823	426	84	122	134	1128	591	361	798	24

1224	1111	179	1143	526	1219	1166	176	834	10	1221	1093
1062	1232	167	587	647	342	490	1041	1114	785	525	1020
473	1233	856	562	423	579	527	427	1194	215	653	242
1237	767	795	1153	862	952	998	644	1081	461	182	232
29	539	1025	1223	434	411	320	932	1204	499	11	546
1112	558	958	1160	486	467	1039	436	1245	1151	421	920
126	692	174	802	581	1066	852	355	1023	180	125	821
1104	129	265	1094	158	1101	824	871	555	1156	106	681
1145	645	803	1208	114	1068	1080	1222	269	996	1074	263
1064	1075	1184	213	699	123	312	1056	1130	175	468	754
380	1063	616	234	1200	768	865	1154	108	1069	631	698
360	1207	397	356	711	540	484	1180	1215	941	1220	596
604	1244	774	916	1050	749	420	169	1029	1242	424	376
1107	1178	820	352	372	118	443	818	701	1136	690	950
1196	93	724	543	1155	476	1034	963	196	978	1086	472
1230	781	610	273	1125	1137	531	120	554	264	969	763
165	407	390	1209	583	49	648	183	331	104	985	130
1126	1152	977	43	917	508	614	992	428	1067	582	983
987	454	497	230	530	285	374	1097	758	308	354	495
664	128	825	441	990	288	951	1131	1008	1004	632	121
752	249	105	682	1195	460	1162	981	746	297	228	700
1116	1095	296	83	1164	1198	528	502	27	762	769	1139
655	863	514	1028	276	811	1177	663	432	12	1002	716
826	96	59	458	1225	1181	136	416	830	1072	937	209
94	986	784	940	146	658	204	976	945	918	185	1012
643	611	1169	1142	1079	1173	793	457	1191	393	212	635
239	984	261	231	1010	1186	1203	576	1124	1140	404	738
991	33	1163	358	586	1098	324	968	1083	642	972	76
1013	1182	1092	751	16	88	817	957	569	135	1117	535
622	670	1009	291	832	329	28	979	72	236	1144	462
298	79	626	772	221	238	487	559	1236	556	946	633
702	69	235	1005	109	140	319	452	1054	155	1019	304
485	756	512	74	398	216	1210	925	1171	911	1243	842
997	478	1239	293	921	429	160	933	1088	431	145	1229
661	295	322	35	1055	1043	1016	201	1150	637	620	482
369	790	253	640	719	491	1061	1042	237	113	914	1122
64	348	171	727	233	254	522	206	1174	81	115	511
1193	928	1218	346	931	405	1	613	1214	394	709	592
359	307	553	708	850	163	999	166	39	1011	370	949
710	1113	1157	935	1022	860	147	78	948	198	1211	177
330	19	771	211	684	993	677	980	1206	927	723	804
1065	970	1044	1099	729	523	442	908	1217	1165	159	9
828	995	150	244	1231	15	325	137	1189	1205	844	691
922	964	564	89	281	243	671	218	657	1000	1172	939
1059	1032	683	649	783	989	757	606	385	965	470	954
389	195	959	806	1060	615	676	630	219	1170	1052	294
257	864	869	371	715	301	840	318	1118	1158	268	1027
627	435	1103	47	1015	1175	975	1192	313	1084	1132	529
67	787	953	1167	1135	924	919	1073	1031	1058	853	271
384	471	812	947	18	1161	765	567	934	217	66	759
545	475	8	1147	505	1001	1045	599	1141	1235	1176	584
284	1216	929	704	1089	943	501	717	445	966	693	605
259	1168	733	743	747	1138	550	336	343	53	814	1133
144	913	439	1051	872	1127	909	1024	459	915	17	504
944	1096	1046	73	726	1105	1026	956	1188	718	549	1146
48	910	494	189	222	1017	827	334	612	585	745	678
208	580	974	938	1123	396	1190	1030	367	55	337	366
962	246	21	4	777	1179	1149	1227	220	930	1226	1021
800	659	1183	731	202	1197	403	1090	988	533	379	1057
1106	1071	515	282	451	1038	574	1033	1199	226	1108	923
1007	280	111	544	406	61	748	1274	402	753	181	1374
143	306	868	1264	6	193	489	618	1351	667	1346	1375
1267	32	732	687	286	688	507	363	1308	859	95	575
855	1376	345	1358	65	1361	1257	565	44	1371	214	703
447	36	561	1329	799	203	625	720	278	1287	845	210
444	1280	1327	1303	414	1268	1273	846	477	446	1253	578
251	1377	822	229	1328	1295	847	419	1320	837	1378	831
1373	1362	538	1379	662	1254	1334	778	1331	58	45	602
87	287	854	839	1292	780	1322	309	619	401	279	588
849	1350	332	791	351	156	1247	1355	639	157	132	1380
1282	1270	1381	641	375	730	1300	498	388	594	519	1382
689	1313	1258	542	595	162	1383	493	275	252	5	1367
1348	368	776	227	1319	675	413	1338	20	1318	188	338
516	548	568	590	1294	415	1343	609	417	1356	54	1315
223	1323	1316	1357	685	1246	1384	433	41	815	1360	1296
755	1345	316	808	650	310	449	1284	623	1347	1302	101

1364	481	1293	142	170	517	680	1314	1324	1281	1385	317
1321	572	833	1261	141	62	697	305	1330	51	173	1312
669	148	1256	656	92	391	161	382	1251	600	665	805
686	1342	589	1369	534	349	1249	838	1332	1336	521	1272
387	1262	1340	1301	70	552	192	1386	1311	603	1277	766
440	789	638	292	37	1279	1291	1372	725	1387	247	851
1299	86	1388	674	848	1310	1359	57	138	1286	124	1285
1266	1248	809	1252	340	289	1389	775	488	71	694	607
31	1349	628	1263	1390	1255	365	1341	300	1275	262	1309
1250	761	25	50	85	408	256	577	835	841	742	1370
707	712	666	1283	1354	455	1317	386	430	1269	492	1339
14	660	1288	314	1368	1306	151	1353	807	90	1265	788
608	510	1391	1335	327	1304	741	7	1365	197	245	861
1297	1259	80	679	1333	75	1326	1352	779	26	199	270
547	537	636	1289	740	323	326	395	1298	60	1344	302
1392	1337	1271	593	127	63	184	13	513	103	1307	469
1290	1393	1260	1366	1305	1325	1278	1276	107	241	315	240
1363											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets)

7	36	33	79	39	24	81	16	5	73	3	1
20	88	30	44	56	41	11	62	65	47	98	26
814	806	645	389	313	988	737	838	346	946	243	890
300	123	778	333	207	229	663	571	127	816	821	355
1220	1344	1025	1270	1147	1345	1020	1218	1338	1037	1085	1041
1016	1226	1114	1073	1346	1190	1195	1253	1227	90	40	70
15	95	6	53	87	34	91	25	82	76	45	13
93	14	55	28	37	17	72	66	85	61	22	64
21	51	8	77	35	78	86	38	10	42	18	99
31	60	2	27	54	92	94	80	19	71	69	29
67	46	84	89	12	9	96	4	57	58	63	49
59	75	50	23	83	48	526	275	178	752	965	810
453	272	436	898	463	245	321	586	113	506	208	419
708	183	640	451	376	195	372	414	942	895	238	302
443	631	474	712	434	917	277	392	187	467	607	557
845	647	688	188	52	97	630	680	865	449	583	164
262	105	634	317	639	119	928	464	714	802	797	932
823	822	953	684	529	694	270	991	482	154	956	978
440	135	197	670	280	351	987	248	881	779	933	327
606	214	662	695	292	456	909	608	158	285	491	877
311	771	395	891	938	803	563	612	561	963	715	501
447	153	498	134	792	220	215	811	874	427	283	418
538	68	32	43	74	687	345	661	924	284	486	359
839	495	115	893	732	103	307	442	305	675	600	136
157	603	106	417	122	918	441	785	930	982	181	466
255	758	765	794	817	271	854	597	494	900	219	198
484	315	194	218	923	382	667	309	901	408	120	558
204	772	336	761	470	266	519	996	240	582	795	969
168	142	541	559	540	853	564	578	936	916	788	882
274	861	527	365	651	367	707	665	162	435	469	929
589	979	939	290	753	755	776	584	656	269	391	416
860	700	786	514	344	566	492	319	993	139	384	790
256	475	252	437	244	145	276	544	374	598	373	828
258	652	870	685	784	318	801	927	353	109	180	324
689	858	910	108	940	508	525	137	749	697	975	296
489	650	503	329	101	570	326	850	716	581	246	627
403	782	375	556	985	949	126	907	422	179	282	221
161	944	981	604	517	349	212	783	138	906	835	873
594	545	880	655	462	967	539	457	203	528	717	952
341	316	366	813	591	836	299	769	926	951	547	904
848	844	840	800	744	576	819	995	412	413	147	588
812	424	356	404	554	869	619	186	390	150	692	825
740	852	343	601	831	431	312	989	535	481	377	702
764	286	855	682	976	261	515	768	750	254	696	883
654	169	304	618	889	727	710	805	362	799	360	621
543	530	288	428	379	118	587	531	423	833	426	210
342	241	905	787	572	439	560	954	748	653	402	773
394	980	448	649	223	488	760	415	368	959	611	913
287	493	569	729	253	767	132	322	536	479	553	476
648	410	815	964	562	458	323	217	385	830	144	234
173	617	643	143	660	628	461	551	473	193	337	257
644	698	673	293	104	872	757	334	843	826	510	452
289	107	573	213	251	459	516	512	110	998	438	242
974	971	314	471	669	550	620	465	148	580	593	777

192	990	155	230	614	504	638	983	915	775	842	149
236	363	759	999	140	745	851	885	736	766	941	294
871	867	994	152	914	407	396	555	690	886	585	896
369	613	731	966	249	774	460	170	505	331	970	721
846	807	892	295	577	298	804	961	191	796	948	159
590	791	674	117	742	720	738	490	405	664	599	184
432	166	509	762	672	986	615	542	306	165	130	622
455	728	683	445	820	658	957	605	177	279	176	335
859	129	364	706	202	879	809	502	348	383	657	477
281	947	832	671	857	868	546	595	112	268	233	724
741	228	713	878	908	167	370	887	925	100	325	704
200	211	499	681	626	574	677	763	433	701	624	446
637	354	111	960	841	977	125	483	444	478	972	711
131	808	430	133	387	380	163	468	911	310	897	864
160	513	227	141	945	518	352	718	984	524	487	425
116	781	411	278	636	876	128	320	532	330	629	725
834	454	250	114	291	668	520	534	849	770	185	358
623	973	709	409	303	146	579	259	955	429	175	747
397	950	659	679	239	856	642	610	609	568	301	958
496	297	533	121	937	480	201	730	552	818	500	225
400	182	521	328	894	922	935	522	686	888	189	931
199	511	754	406	124	232	102	332	920	361	398	693
224	386	399	592	350	739	567	703	265	226	919	263
899	222	472	746	156	635	378	507	206	171	273	827
401	866	565	548	338	789	421	992	209	151	676	641
485	884	699	260	393	523	902	388	726	997	968	190
381	751	633	231	264	756	824	357	420	625	450	743
196	216	602	371	237	962	174	235	735	875	837	172
863	829	666	247	719	912	793	339	847	943	646	734
934	347	798	340	205	575	733	267	537	903	678	308
616	862	596	632	549	691	705	723	497	921	780	722
1229	1311	1126	1074	1105	1179	1155	1223	1023	1161	1175	1129
1184	1134	1305	1319	1088	1082	1347	1079	1281	1152	1144	1149
1288	1176	1186	1289	1254	1283	1348	1017	1159	1189	1349	1217
1131	1039	1350	1272	1064	1192	1188	1228	1115	1095	1264	1310
1058	1201	1198	1022	1014	1043	1108	1071	1351	1352	1027	1165
1151	1059	1028	1353	1141	1200	1143	1354	1247	1100	1029	1112
1005	1355	1257	1083	1194	1046	1356	1035	1024	1307	1101	1091
1278	1287	1092	1139	1120	1250	1007	1246	1225	1116	1051	1357
1244	1086	1209	1096	1090	1284	1358	1106	1127	1318	1309	1290
1193	1216	1076	1052	1259	1359	1117	1360	1361	1239	1047	1034
1003	1057	1277	1111	1093	1332	1301	1009	1183	1328	1178	1061
1099	1265	1215	1240	1362	1213	1269	1298	1049	1312	1363	1171
1263	1252	1364	1197	1077	1243	1365	1366	1145	1214	1142	1018
1315	1123	1146	1107	1060	1367	1113	1006	1160	1337	1249	1326
1011	1279	1368	1208	1196	1275	1056	1170	1110	1234	1335	1261
1294	1069	1031	1203	1063	1267	1258	1300	1202	1157	1241	1030
1150	1280	1206	1369	1181	1180	1021	1370	1158	1121	1097	1205
1128	1012	1293	1089	1341	1237	1187	1317	1102	1010	1068	1304
1371	1291	1372	1148	1153	1065	1256	1303	1306	1124	1295	1053
1013	1162	1268	1044	1373	1078	1032	1166	1177	1137	1026	1255
1374	1375	1133	1236	1191	1172	1048	1245	1282	1119	1296	1062
1130	1308	1376	1207	1299	1036	1377	1333	1136	1199	1378	1379
1292	1164	1045	1248	1343	1380	1314	1235	1321	1066	1232	1182
1271	1336	1320	1260	1251	1381	1382	1168	1286	1383	1384	1316
1322	1385	1042	1262	1173	1212	1038	1386	1040	1274	1154	1327
1273	1185	1387	1313	1210	1388	1019	1132	1033	1001	1156	1389
1122	1070	1098	1211	1055	1329	1323	1285	1072	1054	1325	1004
1302	1118	1231	1219	1204	1087	1222	1221	1125	1080	1342	1224
1015	1340	1390	1050	1330	1233	1094	1081	1067	1104	1174	1297
1391	1331	1075	1392	1339	1238	1230	1135	1324	1276	1163	1334
1103	1084	1002	1138	1167	1109	1242	1140	1008	1266	1169	1000
1393											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز; لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال

(a)

£ễ∏؟ŒŒ₪¨à{Œ¤£Œ∏´ъà‹ъàŒъ ﻭî¨àŒễé∏¨£ъễ££î¨îغﻭàîŒ₪Œ¨ﻭъ{à∏é₪¨àŒض¨éغŒ‹ààéîъŒà∏¨غễŒ9˚ễé∏ŒŒqъ˚ﻭë¨∏∏∏c¨î{ﻭî8à‹9¨
Œ∏ъ{îъ£ễà˚∏îъàŒ∏î£∏ъ¨´Œàî^ëîîîŒààîéë¨Œễ¨ъ´éà∏àààŒ£8¨£££¨¨∏´£èễ£éъëŒŒŒ£ъъ¨{éъ؟£ëà¨ŒŒ£ŒàŒ¨ŒŒ9£´£∏9£ﻭŒë؟éễéà
₪∏غà8ъễﻭъŒъ£ŒŒŒééàễë£¨Œ∏8؟Œ9}ъ¨˚Œﻭ Œ¨∏∏∏‹؟Œ8££àŒъъŒŒ£ễ‹îﻭﻭﺧ ễ؟∏ëé{£‹ъŒ9é8₪£∏غ¨9∏ﻭŒ؟}8ъî¨ëîéŒ£∏‹∏Œغﻭ£
£ъî˚∏ŒъŒàъﻭ¨îﻭŒ}∏ààà9¨î¨Œ£ễ8¨{àŒễq£ъ8£ﻭ₪∏ﺧغễîъ¨غ‹ъsغ¨´îễé∏ễ££éèŒغ8ﻭŒъ£Œëà9}¨éﻭễﻭ¨£à¨éŒŒŒ∏∏∏é˚ễîъŒîễà‹∏ễ∏ễé
ààŒ∏‹î‹ъŒ8∏ễ{é∏˚¨éŒŒễàé؟‹ъŒъéễغŒễ∏££ŒŒàъﺧé£¨¨àễ8}ﻭéŒ¨î9∏Œﻭ Œë¨îъ£‹ŒŒëé؟Œî£àéễëъъ£ễàễî£Œъ£éŒъëc¨ﺧŒ‹ëﻭîéễîé
؟ễ؟îъễ˚î¨ŒﻭŒъŒễ8£éﻭé¨غ8؟ъ£ﻭễéﺧ∏ъàễŒ∏8¨ъễŒ£é∏é£Œ88Œễ9‹^ﻭî¨îŒ¨{˚£££Œ´₪£₪Œ¨à{ﻭễễŒ₪ë؟ŒîàŒ₪ъîغŒ9ễ∏ъ¨ъъ£ъ£à£Œ
é£ŒëŒễъ؟ъ£ëî‹îŒ£ъ‹Œ´£∏ﻭŒغ¨Œ£∏éễ£ъ9£îŒc∏ъàî¨8ễ8ﻭŒà£¨{î¨Œ₪∏¨£₪∏8éễъễà{Œъ£‹à9ễ؟ъà₪Œ¨ъ£˚₪àŒъîъë؟ŒŒ9£îîŒ8∏£
₪éễ¨Œ£¨ъ¨{ëc∏ﻭễ∏∏∏Œﻭ}è{è8îễ£¨£ễﻭàà£ễŒъ£Œ¨Œà؟àŒŒŒŒŒà˚؟ъ₪ъъŒéîà£c{´Œễc´¨8´{éîﻭ¨éﺧغъъễ£∏

(b)

¤}ŠŠŠگ؛شŠ31ﻩ»[ŠŠ؛3133û7صú)Šúú~Š17wﻩ)11µŠ3گú3گ3گ¤ﻩŠŠ3ﺀ¤»1Š[1ú-
/Šگص31w1ﺀگ3wg7wŠگ؛گش[¤1¤11صگ7¤گ71m7w‹ŠwŠگŠ1ﻩŠص[«3صw¤ﺀ3گ‹w3¤¤)1¤صŠw/"f¤31)Š17¤1گگŠ3گw)»ú¤3ﺀ¤گگwŠ3Š3
Š111صگú‹ŠصŠŠ1¤3‹‹ú§¤گگwگ»‹-[ﻩگگwŠwúﻩŠŠگ‹úŠ¤؛گ7گ[Šw7Šﻩ--گw1ﻩثú1‹-
1ﺀﻩwŠwŠŠŠw3¤¤Šصm1¤Šگú¤Šwگ"گگ؛صú)w7Šگ»3گ;]Šw¤ﺀ1صŠŠ¤Šصú1Š7ŠگŠww3گŠ7گ‹گگ1Šگ3¤گگŠﺀ3گﺀû1صص¤گگﺀ/Š
1)71گ7؛31گ-wگwŠg1شû1گ3¤ﻩúگŠw[11wŠ‹گگ¤ú-
¤1‹ŠŠ11wŠ[11Š3گê¤ش[7w1ŠصگŠ7Š¤Šûگ"[ث»گŠ¤¤Šúw[177؛ŠŠگŠŠúm1w[7Š1Š7úثew1؛~w3¤wگ؟ŠŠگ¤"گگ¤Š1úw13ﻩﺀ1شگw1
ﺀگگw3w13¤»w)صŠ77گ7؛7گ3u[3ûŠ3Šﻩ31گŠ1ŠŠ7؛Šf3m71گگگww7]گ7گŠ"(Šص¤w7f»Šû7Šw33-¤ŠŠ-
گ¤§7ŠŠ]úŠŠ[ŠŠwگú7/)»13¤گ3w"ﺀŠw¤)گأŠ1ﺀيŠŠŠŠگŠ¤3¤گ؛Šصگﻩﺀ؛گ/31گثŠ¤ú‹3[û¤ﺀŠ3گ311گwwwŠﺀصگw7گw)ú1-
g«7§[Š7)ŠŠ/11ﺀ311ﺀ¤Šﺀw]Š37Šw711؛w¤گŠگ«e~3Š3صwú»ش¤û‹Šگ311Š;wg33wŠ¤7ﻩشúúﻩwگ§w771w¤گŠ1w[1»؛w33گŠ؛mŠﺀŠŠ
ww3w¤wگg(ﺀ؛ŠŠشگ3"§w7(گŠگŠûw[گ)úگŠê7úŠشŠw»1گگگ7؛گأ§¤w}گص‹Š؛¤Š1ض¤fگگm)1گêگگگûúگú7¤گ1؛Šص»1wگŠ1-
"11"úm1‹3‹Š؟گwŠ¤ﺀ3w[3ﻩŠﻩŠ1mw3‹ﺀ37‹1ص[Šw1w3گ]~~3Š¤131‹1؛êúŠw1µŠگŠ1؛7گ‹Š3شw3گ3¤3Š7گ¤Š[úmﺀŠeú1گ[ص¤گﻩ3؛e
e1ص71گگشŠúŠشگ7‹w1گ3ú77ww73Š¤7-ﺀmگگûúŠwŠû[ثúگ§wگ3ﻩú7w7¤؛گg1;¤3ﺀﻩ-[ص¤[13Š/گ1‹ŠŠ1ث)؛؛گ]شگwŠ3w7Šû
Š)Š¤wﺀش[گü¤؛fص3گگگ}¤گŠ)[¤¤wú1‹Š3ŠŠm31"]3؛گ"113گ3]3‹§گ71صŠ¤§ŠگŠگg؛ŠfŠ31؟ث3w"w؛Šwگ»گ7"گŠ[ú1؛73~ûŠﺀگw
1؛-‹؛‹¤؟گگ//¤ﺀﺀصŠŠŠ1[گŠŠŠŠ-Š11گ¤ث7úگŠگص§گŠث؛ﺀû[/mŠ7[‹Šگ"w1w¤-گ

(c)

Figure V.4. (a) *Donnée originale Texte2****,*** ***(b)*** *Première version chiffrée****,*** ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{\text{Clé}}$ = 1,8704 K octets)

8	2	4	7	6	9	1	5	811	910	899	896
878	885	622	892	767	914	357	901	895	297	771	211
70	851	189	890	897	694	880	913	662	78	908	900
906	210	170	889	879	904	833	407	208	300	498	306
259	891	871	858	469	698	543	87	903	368	52	884
266	835	642	886	92	909	905	882	883	405	527	371
915	658	12	893	113	172	457	898	888	633	887	242
877	881	414	376	911	556	912	907	902	163	648	281
894	876	949	701	174	1155	643	192	649	213	466	1103
514	1076	1022	660	624	3	632	430	623	944	925	770
785	796	67	1194	230	707	799	1177	1218	245	508	1007
179	484	1209	1223	549	1118	1037	1167	740	73	1160	248
193	1088	287	1200	304	131	175	544	511	428	1173	176
1176	931	983	815	951	370	32	955	826	53	945	754
168	863	1009	1219	692	532	459	273	1018	356	730	255
636	604	539	11	984	79	930	817	383	581	829	1148
1133	764	961	1068	1127	1140	1225	630	956	403	1143	1235
822	601	755	1217	535	1073	960	1137	203	325	545	1036
745	1112	971	1097	1168	1232	223	437	805	1062	1204	321
272	374	1080	260	748	86	1045	434	429	1157	675	994
145	1142	587	585	1229	1069	1044	267	386	471	1163	147
1102	1182	747	593	798	1054	1019	82	981	978	348	1066
928	669	1192	1065	1038	311	1196	943	1141	597	610	63
741	1085	1075	234	1063	962	850	682	1156	401	843	670
81	1058	84	1000	151	525	60	1110	225	501	969	552
341	446	486	985	1093	580	372	1187	998	517	65	778

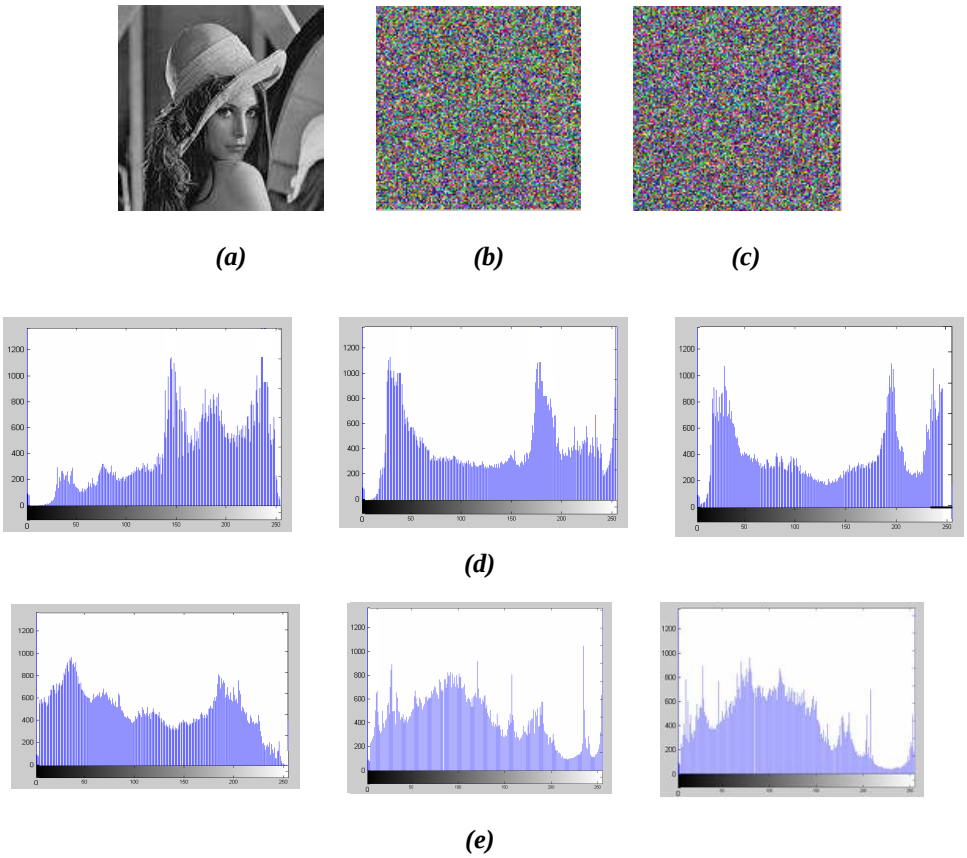
606	137	97	938	180	952	1230	90	35	1067	1084	1083
1014	463	237	814	650	319	935	379	315	309	46	676
1139	1060	1238	240	1152	645	689	846	221	1003	251	116
635	1215	182	28	467	460	171	495	377	674	80	1228
1027	392	452	54	502	957	1064	1079	228	733	932	359
609	997	647	834	1149	1004	751	166	923	830	1166	976
980	1198	1116	975	578	631	75	253	235	1095	1224	860
965	449	1109	43	979	150	763	202	271	252	394	59
516	1205	239	838	874	247	353	685	704	169	83	177
1012	389	114	39	1074	659	924	982	104	565	783	839
705	167	1222	987	656	292	57	93	194	1056	1165	282
861	500	526	564	34	411	922	1189	566	445	506	99
1119	793	257	637	398	1087	1122	1185	105	582	51	494
490	1024	199	207	696	782	596	521	412	138	50	1233
1030	993	1191	258	653	1023	1105	312	757	492	712	419
173	1055	406	948	21	713	119	378	402	929	1186	574
19	654	286	1098	327	246	852	1017	1039	183	592	936
1082	954	332	714	781	328	666	85	518	620	787	1033
20	920	347	750	546	756	1034	1181	1031	62	1035	497
22	522	519	346	1051	591	803	373	1239	1015	927	37
382	966	586	218	140	1124	410	1183	74	595	1180	942
841	1170	872	1125	742	1243	845	316	1104	106	554	159
802	967	1099	710	344	158	196	939	1195	1091	146	614
569	1002	562	1226	641	279	567	416	1178	100	1132	236
823	49	809	296	360	254	575	1013	1048	313	441	801
206	992	1193	157	48	723	1234	280	184	768	615	415
691	824	1213	762	743	753	187	233	404	262	479	1016
422	365	481	1171	661	837	323	963	686	772	605	1227
599	572	777	1072	443	717	1244	1135	1131	735	448	1047
109	657	941	1078	1071	531	865	794	1032	25	107	1100
238	621	120	334	71	795	1077	1001	1159	626	1161	451
342	23	699	1040	919	1145	510	336	715	274	739	468
1231	1172	1237	831	1150	1144	423	14	832	1188	31	1162
68	918	385	338	766	867	1050	528	216	305	577	524
972	709	1081	222	639	399	72	1212	953	1216	124	112
432	1203	1201	126	570	1146	1211	101	426	125	1136	178
209	706	1005	1174	1086	921	926	33	917	358	1011	161
977	450	42	1057	688	728	1111	847	219	456	322	970
197	999	1199	720	366	1175	201	627	1130	320	536	290
1236	995	229	868	478	1214	330	589	1008	625	355	690
968	310	973	337	488	1006	672	77	148	746	808	603
608	1070	343	667	553	354	752	989	946	520	568	628
1061	725	693	1089	110	1117	1151	326	629	1108	1092	294
855	759	16	792	1208	651	391	776	58	1164	256	800
263	916	1120	1179	424	579	547	1115	369	160	974	538
15	89	541	934	947	18	480	1049	1029	465	612	191
958	1028	986	1134	774	548	1138	1210	1197	937	1221	806
1096	542	613	1123	433	678	804	726	76	933	1090	571
1158	1043	959	1094	1041	10	1052	1106	1020	1169	724	1241
862	990	988	269	461	644	711	1153	421	602	677	684
1107	594	1206	1242	164	853	1021	475	277	1113	214	1025
1053	440	491	573	1207	1059	683	134	139	1046	318	864
285	1121	560	1026	1154	397	996	190	509	1190	350	702
1128	417	1042	438	950	673	1010	1114	515	991	665	810
91	869	964	493	188	395	108	844	152	769	335	827
1101	1129	88	435	64	472	1147	807	551	56	1126	719
1202	791	857	136	779	217	1240	444	1184	499	409	1220
668	224	940	299	1343	697	1344	616	442	95	1336	1334
1286	204	646	836	220	489	301	700	1313	288	820	561
773	1297	1345	156	607	473	476	1259	721	133	98	840
1346	351	464	559	1347	384	1348	1304	375	1349	563	1268
436	873	652	439	142	731	1264	314	111	681	393	611
784	1302	558	1270	1350	205	1351	165	13	303	361	1269
1352	734	618	381	732	363	1353	1354	664	557	380	1294
513	94	530	727	1291	244	1355	1338	367	425	454	758
333	1356	1300	1357	1248	339	1342	1341	588	1282	250	317
760	122	1308	512	1358	69	1299	129	102	1251	1359	534
331	788	1298	396	1360	268	264	103	427	482	1325	1307
523	1361	329	1324	1362	1363	1290	1364	859	455	155	663
1340	1333	819	261	1365	1366	1329	1328	786	293	284	1367
154	1331	270	1265	849	132	27	135	729	130	749	505
1368	231	638	1369	390	1370	345	540	1371	1278	1293	276
1372	1250	308	737	17	1373	241	298	362	1262	198	127
447	153	474	1272	1252	825	349	121	1315	1303	215	47
1337	1305	249	1260	186	775	671	1320	387	1374	291	870
289	1375	400	41	227	1289	144	30	55	1339	744	1316

1296	128	1319	504	1376	1245	1377	529	1378	848	431	485
1255	418	1254	708	352	1267	212	232	640	1246	1314	458
1379	722	1327	1380	634	181	118	797	123	590	1280	679
1381	1326	1258	195	1271	507	38	1382	66	1279	453	1256
1309	1383	738	483	340	655	736	533	1312	1384	420	364
842	149	780	1263	1332	1275	1274	283	816	1284	1330	1266
866	1385	117	598	818	703	200	716	408	1285	854	576
295	1386	1292	115	695	1249	1318	40	185	302	1306	1283
1288	45	487	1323	875	143	600	477	583	1281	1253	550
1387	278	619	828	324	1257	813	812	24	790	503	496
36	1388	1389	243	96	1301	1321	718	555	789	1247	388
26	1261	821	1276	162	761	1322	687	275	765	1390	1287
617	1310	141	265	44	1277	226	1391	1392	1311	29	1393
1317	856	584	1295	680	537	462	307	413	1273	61	1335
470											

Clé de chiffrement de la deuxième version chiffrée de Texte2 ($Taille_{Clé}$ = 1,8704 K octets)

681	257	54	272	127	416	212	193	126	445	404	584
108	735	917	617	694	270	386	241	702	562	898	175
381	459	904	737	596	689	130	228	326	836	460	289
37	958	638	703	356	817	741	466	971	858	1	909
688	524	53	13	448	624	634	199	369	807	116	686
102	121	204	490	397	187	670	650	522	163	256	950
868	720	153	510	660	580	68	427	698	546	523	900
621	306	312	331	721	488	314	243	350	313	452	325
844	986	82	499	494	48	12	991	526	453	479	910
978	56	409	607	172	240	620	627	391	779	15	641
174	931	71	90	55	265	963	947	981	60	84	250
95	39	977	513	748	220	235	516	507	916	401	190
360	213	362	725	447	333	105	491	857	78	881	383
89	934	705	999	123	426	231	293	323	91	943	557
803	846	324	945	791	718	826	903	396	258	839	602
713	83	859	38	639	411	400	201	359	545	487	92
138	766	879	799	346	558	928	864	49	961	365	519
149	146	371	18	281	158	430	684	277	885	754	58
374	435	269	318	358	850	335	953	567	443	970	899
169	719	291	789	809	613	432	595	275	882	247	347
62	282	851	685	97	775	151	814	601	384	439	771
980	343	552	598	106	853	541	192	612	993	687	154
271	461	496	125	708	776	948	744	939	375	232	952
286	927	675	671	483	575	373	455	155	305	421	761
731	279	869	288	642	933	788	367	122	676	46	997
35	31	830	902	873	480	736	402	751	477	336	654
450	704	696	295	236	534	244	944	988	629	230	861
341	100	433	906	786	390	750	905	307	42	351	659
714	301	395	34	205	549	77	438	215	285	454	662
871	883	949	884	165	573	935	398	728	112	888	5
985	304	412	440	505	514	219	292	920	131	965	72
349	509	833	287	157	707	308	47	472	810	492	144
339	489	734	139	233	223	474	261	533	209	987	656
912	994	475	911	968	98	772	506	553	214	610	469
132	202	159	568	925	983	938	237	224	669	777	693
538	964	891	746	431	464	604	227	722	756	773	554
50	527	142	252	529	334	171	712	465	437	515	255
874	57	758	540	161	760	733	503	792	27	486	532
822	385	462	234	407	299	583	210	446	88	544	757
436	319	700	109	478	614	393	732	246	599	556	316
682	753	767	70	586	423	796	361	570	793	177	913
73	8	414	30	372	81	709	429	456	79	64	936
508	878	578	63	780	329	442	135	110	457	597	226
870	512	876	458	812	866	266	942	765	768	302	26
330	69	76	537	724	907	3	946	608	500	561	872
4	424	23	535	865	181	120	493	539	315	160	93
589	588	611	701	200	216	661	531	419	115	677	273
399	982	380	188	890	382	24	377	749	655	194	922
673	87	653	823	267	740	85	221	44	582	310	189
813	956	834	695	995	366	410	889	600	353	140	425
590	783	838	908	672	185	819	806	918	501	892	816
955	229	930	128	778	536	542	413	560	484	559	504
975	99	941	186	867	420	984	476	518	66	606	622
405	976	141	937	592	797	378	631	485	211	33	566
666	521	723	28	635	297	957	441	7	683	251	992
626	511	101	591	21	665	716	636	996	551	548	274
585	363	979	332	96	117	609	249	129	923	644	710

637 517 738 51 43 357 969 563 222 203 322 547
470 860 344 798 355 863 248 628 921 845 841 184
847 502 65 565 134 886 848 951 940 471 594 406
774 818 663 45 468 14 645 972 303 959 770 337
855 260 124 752 824 623 619 574 451 781 962 9
825 368 498 640 196 428 877 473 166 197 463 354
415 764 118 815 587 739 276 183 67 321 94 298
564 577 136 785 897 801 444 36 808 364 895 657
820 759 179 311 495 658 147 742 376 691 16 579
167 896 370 434 625 804 467 571 805 745 387 379
664 6 2 690 652 795 967 300 422 726 827 837
572 191 831 61 150 901 225 706 924 327 729 593
320 253 17 152 543 667 954 715 254 893 697 137
263 180 11 143 74 990 19 278 875 763 119 345
605 264 408 22 576 699 802 497 309 854 929 86
680 678 842 168 482 113 394 649 679 727 296 782
835 787 615 618 338 389 25 743 692 52 550 843
114 919 207 828 80 40 284 178 730 10 238 966
242 569 794 832 103 755 392 182 75 481 849 647
821 259 974 104 294 880 418 973 603 674 998 530
616 59 32 262 348 651 195 20 133 340 894 632
156 388 403 643 829 176 581 960 29 245 198 217
762 711 717 41 239 862 747 668 148 914 555 206
449 926 280 352 417 111 932 852 342 784 633 173
887 989 162 107 218 170 328 646 208 790 915 528
290 811 769 630 648 317 840 520 283 145 164 268
525 856 800 1319 1015 1163 1032 1115 1334 1035 1004 1255
1286 1101 1309 1186 1325 1012 1335 1304 1336 1136 1337 1107
1172 1113 1249 1190 1227 1058 1010 1251 1111 1023 1338 1127
1302 1114 1033 1100 1228 1128 1112 1031 1020 1299 1109 1311
1339 1226 1081 1202 1076 1054 1295 1340 1069 1341 1198 1216
1342 1183 1343 1002 1148 1070 1176 1328 1303 1003 1121 1029
1277 1229 1089 1230 1256 1091 1175 1060 1042 1119 1182 1188
1074 1050 1132 1224 1333 1170 1078 1294 1168 1344 1149 1320
1239 1162 1203 1223 1052 1310 1269 1233 1210 1083 1265 1307
1275 1298 1045 1326 1345 1346 1272 1048 1317 1347 1261 1030
1263 1049 1285 1165 1348 1276 1349 1350 1027 1201 1235 1262
1189 1351 1313 1005 1063 1017 1195 1000 1150 1352 1016 1353
1062 1305 1130 1024 1354 1221 1073 1355 1179 1126 1356 1205
1161 1244 1240 1001 1129 1022 1197 1284 1087 1080 1006 1291
1137 1147 1152 1055 1181 1268 1252 1250 1327 1053 1160 1044
1134 1357 1072 1037 1151 1040 1225 1093 1008 1358 1056 1192
1246 1359 1271 1360 1047 1315 1090 1361 1013 1156 1362 1204
1157 1041 1082 1096 1184 1105 1123 1329 1363 1281 1097 1118
1241 1215 1364 1146 1288 1258 1365 1131 1366 1034 1092 1297
1300 1098 1367 1280 1368 1248 1018 1142 1245 1120 1177 1211
1267 1369 1173 1370 1371 1057 1372 1122 1214 1159 1237 1138
1021 1038 1166 1217 1064 1180 1208 1283 1106 1065 1185 1373
1278 1314 1374 1200 1322 1071 1220 1219 1316 1145 1169 1222
1375 1293 1067 1059 1104 1019 1260 1242 1257 1133 1376 1125
1043 1377 1167 1378 1379 1308 1140 1079 1085 1077 1232 1103
1380 1290 1381 1154 1331 1124 1266 1199 1253 1116 1270 1209
1158 1036 1102 1046 1382 1306 1095 1007 1088 1231 1108 1193
1238 1206 1039 1236 1187 1274 1383 1282 1094 1171 1384 1075
1264 1084 1254 1196 1259 1164 1155 1025 1385 1178 1386 1068
1086 1387 1318 1243 1207 1026 1321 1324 1323 1028 1213 1061
1218 1191 1212 1330 1388 1066 1332 1234 1117 1051 1153 1301
1273 1099 1389 1390 1144 1139 1143 1009 1247 1287 1014 1135
1391 1279 1174 1141 1392 1312 1110 1292 1296 1289 1011 1194
1393



(a) ***(b)*** ***(c)***

(d)

(e)

Figure V.5. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

369	423	529	171	397	412	474	15	689	64	112	390
267	21	118	304	6	28	574	465	356	338	717	638
339	110	243	458	508	450	361	608	635	115	593	749
151	742	575	716	428	109	486	584	294	722	521	141
87	695	373	225	452	602	679	13	208	85	468	634
709	298	70	297	514	601	439	359	320	251	222	292
319	765	456	614	142	645	625	288	383	217	125	476
396	718	274	236	729	483	567	114	487	492	303	542
426	628	675	713	296	505	370	730	409	395	764	641
669	460	337	264	230	204	623	358	293	418	150	520
134	328	469	762	440	533	651	551	535	286	438	462
103	355	347	63	158	16	561	205	410	643	703	432
519	693	247	620	346	40	736	366	129	652	164	327
371	506	363	154	241	447	434	624	585	163	748	376
380	144	407	59	143	667	252	10	536	548	766	360
335	455	311	497	200	466	249	751	35	119	165	341
36	741	398	448	656	18	65	233	734	17	531	512
595	660	104	586	81	435	299	101	92	309	454	436
710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579
352	739	698	306	253	577	248	735	515	342	285	61
219	138	568	705	658	427	511	532	210	357	307	153
633	711	98	433	619	691	60	20	137	733	485	324
659	419	719	684	269	392	754	68	259	284	467	295
48	300	224	541	600	715	630	613	172	66	425	558
604	38	692	34	557	263	258	384	51	317	375	111
325	146	701	472	388	607	289	94	644	377	745	580
500	598	496	240	443	594	364	554	74	637	187	712
655	747	576	145	47	501	417	490	622	162	491	626
578	699	750	14	128	493	282	226	459	351	589	209
582	545	740	596	147	275	121	362	470	53	193	291
333	290	179	523	527	42	30	402	738	525	113	374
664	646	507	609	685	756	189	700	148	260	654	464

191	350	378	176	257	538	599	603	11	702	372	571
237	12	96	170	192	653	88	416	540	25	180	522
642	166	72	246	194	707	680	33	639	159	215	75
728	227	516	681	344	90	99	559	127	513	265	7
23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555
69	708	345	494	271	238	726	393	534	132	385	587
686	647	368	203	3	481	482	332	199	167	687	560
539	495	743	44	583	91	610	690	336	31	261	403
379	255	725	498	323	386	198	572	220	484	73	322
677	606	340	272	343	58	503	381	26	421	517	761
79	676	334	348	760	49	663	235	597	566	313	627
39	414	618	518	479	55	752	186	694	563	649	175
223	133	549	420	277	510	727	305	391	228	632	404
196	206	84	52	184	755	212	581	763	429	107	697
724	590	77	737	445	71	682	502	509	120	254	413
758	640	149	616	43	195	242	239	211	611	453	757
218	86	19	310	387	2	105	117	424	244	504	116
449	591	8	229	605	399	629	182	57	461	314	588
177	234	190	102	106	714	122	617	612	216	673	312
431	326	621	80	382	721	93	45	156	665	553	767
475	174	329	394	415	547	188	331	543	365	308	46
67	670	330	157	354	302	389	480	367	232	768	27
207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631
41	471	131	201	245	54	202	444	556	544	720	270
32	688	759	442	279	478	136	197	123	273	671	437
262	457	672	268	349	95	50	488	706	546	183	221
178	160	528	696	62	9	214	615	573	489	430	97
744	22	753	524	446	213	746	731	181	231	283	661
570	401	276	78	400	4	562	526	683	301	89	477

Clé de chiffrement de la deuxième version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

156	453	55	623	217	430	383	574	180	389	257	147
405	685	168	355	721	754	738	227	187	474	143	551
127	578	506	528	477	713	723	319	702	417	384	67
512	575	469	301	86	237	703	204	366	2	268	247
340	452	642	21	12	343	128	214	390	220	262	76
495	281	286	66	397	169	445	170	219	24	48	531
261	338	478	282	635	44	191	209	735	323	410	514
255	234	43	54	516	291	97	344	720	695	129	332
461	117	522	270	89	537	298	161	515	346	758	737
253	35	75	739	240	719	131	27	548	98	455	195
78	683	710	178	595	387	400	403	633	353	350	250
687	37	462	235	119	53	200	385	192	530	179	419
392	488	61	457	45	125	1	576	58	312	552	509
546	706	276	241	617	73	585	681	751	753	517	677
399	91	473	557	701	26	302	637	632	32	565	755
135	539	47	266	110	505	17	631	647	727	303	577
427	411	493	23	743	458	259	122	285	114	489	544
182	72	582	194	264	120	439	279	669	718	292	207
101	407	446	158	475	536	284	630	215	724	25	256
172	622	650	734	87	638	450	521	416	665	762	519
700	251	491	335	287	608	742	34	426	104	370	290
7	315	77	480	330	486	183	619	730	19	331	141
599	555	229	202	359	731	757	273	504	308	167	155
230	542	654	304	165	423	360	733	705	662	648	763
28	433	92	664	694	716	175	760	656	190	171	415
263	627	470	51	100	589	4	707	311	148	88	696
16	466	498	590	111	224	363	468	239	408	162	621
36	333	373	626	561	113	341	70	717	511	379	3
371	109	508	378	463	673	693	221	367	30	388	8
212	750	748	667	374	33	358	661	289	116	447	674
566	173	364	747	314	401	500	185	328	140	634	198
676	449	527	490	260	106	588	79	672	68	759	502
591	142	443	94	420	581	698	108	471	572	296	188
436	38	395	437	545	614	163	525	605	704	121	160
587	318	294	764	307	615	107	761	337	151	422	604
326	69	412	223	300	213	472	249	678	690	580	658
765	11	57	573	99	124	184	523	402	501	226	479
649	361	137	193	618	181	174	39	248	299	579	484
62	670	345	351	636	382	347	454	41	118	644	675
74	620	745	218	252	610	265	438	280	564	60	362
404	503	271	712	418	569	550	406	236	13	600	603

96	82	547	126	568	369	199	534	602	612	485	596
365	31	441	203	682	49	216	729	85	456	722	154
641	258	254	524	668	483	616	435	598	317	532	549
768	613	46	465	628	680	460	225	297	186	246	645
321	133	459	520	196	584	145	413	59	692	231	375
688	102	444	210	293	322	643	714	274	242	339	201
749	65	325	767	691	736	726	424	766	601	144	376
349	159	429	130	583	233	18	529	709	646	625	208
269	380	377	64	594	487	451	244	639	305	176	711
20	112	756	653	71	684	464	157	29	563	189	606
553	651	699	232	50	352	138	728	497	467	152	310
103	10	507	327	715	586	245	434	15	14	746	640
205	81	56	541	526	309	177	393	134	663	562	741
725	368	597	391	394	348	272	288	238	660	40	153
136	533	295	396	571	744	84	146	342	425	164	629
123	689	567	52	320	708	166	740	316	275	559	611
481	686	267	657	93	556	197	752	386	5	336	494
139	554	428	90	83	732	313	592	671	243	448	431
518	132	95	496	277	607	535	80	306	22	540	381
570	372	6	334	652	356	211	659	149	543	499	421
228	278	492	513	357	329	679	432	482	510	115	655
9	624	283	63	150	666	354	324	476	697	409	609
440	414	593	105	442	560	558	206	222	42	538	398

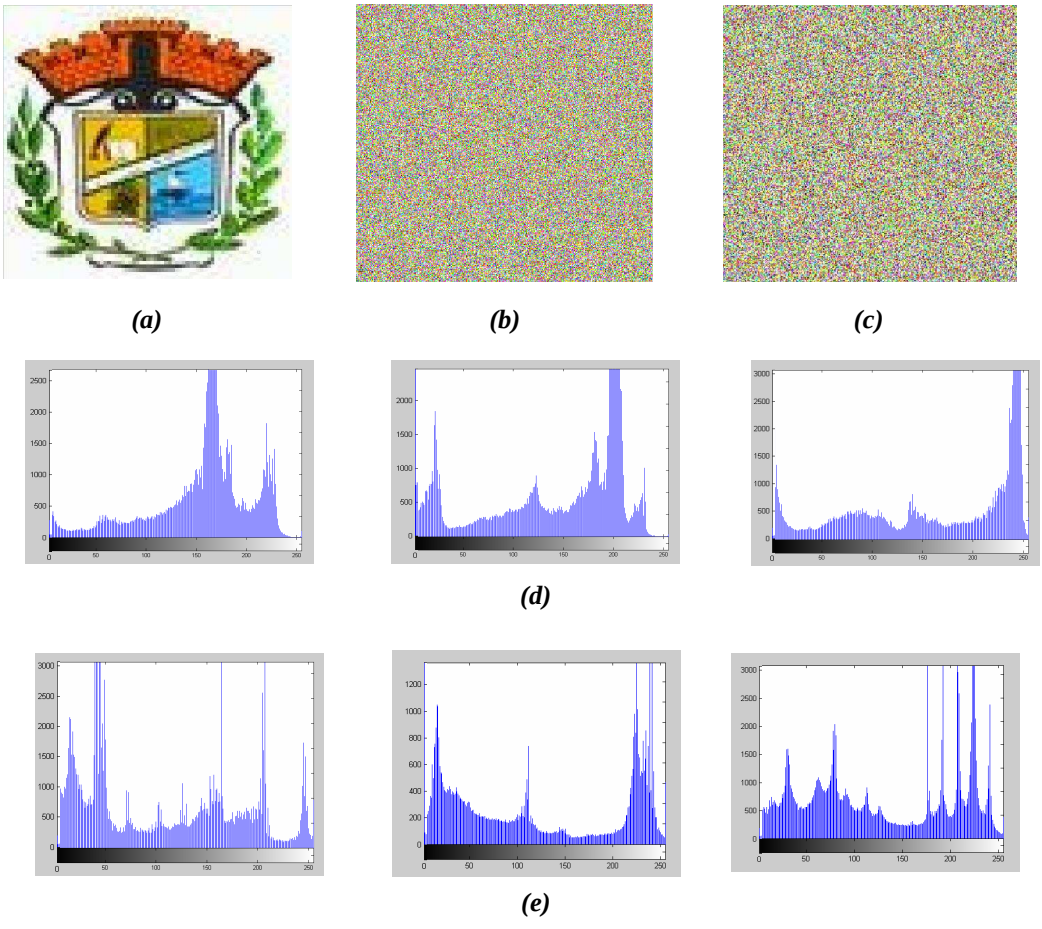


Figure V.6. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

65	9	12	62	15	21	94	7	28	11	81	3
61	36	78	70	91	25	56	97	85	4	30	24
83	32	57	40	54	31	63	8	95	26	75	55
19	96	20	38	49	92	47	44	1	98	52	89
35	67	6	72	69	53	74	60	14	43	84	23
22	82	88	73	90	76	39	13	59	33	58	68
41	50	86	46	5	51	27	99	79	10	71	66
87	48	16	2	45	64	42	80	17	34	77	29
37	93	128	625	414	134	459	456	642	446	447	205
635	230	755	387	503	179	646	371	252	744	662	574
575	475	768	734	494	638	389	441	312	571	445	214
367	308	498	643	659	705	175	676	237	530	674	370
527	626	415	330	333	472	600	238	339	137	730	257
262	144	660	139	217	323	739	329	578	420	618	735
463	199	729	207	340	336	409	551	295	296	477	222
468	276	151	258	725	178	764	245	132	366	525	305
196	717	363	484	658	136	342	397	408	198	598	443
589	455	256	239	287	216	181	290	338	111	334	751
345	392	560	697	355	585	418	18	610	162	534	154
467	241	250	694	464	461	614	500	743	170	244	741
404	259	211	645	318	650	593	442	187	282	639	288
648	267	655	220	573	436	552	344	164	225	727	165
226	667	693	416	116	265	536	532	686	522	471	587
316	260	486	514	488	268	601	628	767	365	382	314
528	195	493	430	108	227	212	145	194	753	716	675
547	152	517	374	182	539	356	677	724	760	377	434
546	119	702	204	590	481	394	114	509	395	301	419
473	490	656	728	168	583	538	426	745	518	149	141
131	279	644	140	580	701	325	348	341	285	203	298
651	263	723	235	462	691	622	722	129	437	354	163
335	398	647	122	469	603	553	737	269	511	101	240
110	361	400	402	317	186	733	277	766	142	264	172
399	613	703	588	231	315	106	669	531	380	233	180
319	504	275	113	425	153	372	748	424	281	157	234
249	581	466	200	289	533	411	630	579	223	221	376
679	569	596	188	421	758	519	146	427	243	123	331
671	508	678	206	304	364	273	440	561	324	297	713
670	274	118	576	715	611	752	620	379	489	369	750
343	278	103	479	621	485	224	526	453	229	604	352
765	653	413	714	499	169	595	762	746	465	410	403
537	100	609	327	117	683	617	570	740	492	373	417
310	505	155	543	246	390	689	102	747	710	594	616
497	381	406	756	563	548	487	632	757	720	512	636
749	474	631	294	433	476	347	661	197	718	393	709
577	174	712	665	592	328	599	435	311	375	719	385
280	480	299	261	292	423	242	159	283	696	542	761
641	349	582	458	565	171	432	431	731	520	732	597
541	605	666	209	368	202	612	337	680	156	272	652
619	633	657	135	125	150	148	506	255	495	104	313
535	540	401	521	412	320	554	634	572	684	654	332
523	627	449	690	183	460	251	682	602	664	742	685
470	309	161	350	515	266	452	640	501	444	567	606
502	429	322	483	711	454	428	726	357	544	378	177
184	291	407	763	624	591	201	218	510	391	556	124
478	293	566	557	623	629	189	721	321	219	270	232
208	302	708	608	586	362	213	388	672	707	248	568
307	271	405	130	115	663	185	253	143	681	698	545
176	422	754	695	450	396	668	109	439	210	607	105
692	673	759	637	112	516	699	121	584	286	736	358
107	147	326	353	562	228	127	513	215	687	555	491
700	649	529	190	738	191	507	138	615	704	383	524
496	193	359	133	384	564	306	550	438	173	236	386
549	167	346	120	448	457	126	300	558	192	254	451
559	482	706	303	284	160	247	688	166	351	158	360

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

498	116	41	239	486	387	497	89	154	262	303	25
129	485	272	506	489	496	446	502	195	127	234	350
240	282	491	338	501	304	256	484	29	209	382	212
238	512	511	494	335	166	495	79	437	381	299	499
480	508	505	481	358	73	482	483	492	392	427	359
507	269	510	236	487	493	243	372	167	231	182	37

94	106	490	87	333	500	71	64	16	504	488	160
503	438	97	227	1	317	336	331	324	76	150	478
509	90	203	716	401	386	194	442	477	328	529	4
762	474	132	3	709	407	735	376	86	596	2	48
651	228	704	200	345	380	201	626	18	5	23	24
385	660	531	732	145	248	562	677	373	663	725	648
736	712	453	84	365	52	569	128	170	183	526	754
43	294	746	146	224	83	112	745	434	448	764	582
400	541	436	635	339	597	390	632	717	26	738	551
383	721	237	93	156	144	463	623	375	627	727	130
152	173	728	731	281	722	220	68	707	656	153	204
193	680	252	54	95	550	429	465	759	70	542	34
55	528	149	414	763	590	232	729	218	646	639	600
458	49	472	57	678	630	664	374	593	533	750	585
190	344	384	117	631	676	539	205	208	300	577	657
697	196	696	394	141	689	131	559	445	513	99	69
669	706	693	578	32	142	756	247	169	768	100	742
217	290	59	264	393	755	330	701	703	332	743	726
266	444	221	549	199	163	659	370	270	250	621	702
268	614	740	552	21	96	435	766	245	622	326	28
278	417	316	752	15	343	733	634	430	618	311	181
340	189	325	532	557	624	765	636	747	9	470	571
140	595	341	538	178	710	308	402	296	573	63	271
125	610	553	449	215	699	20	411	519	263	692	357
690	162	464	439	471	126	666	105	242	720	147	681
604	420	12	523	714	615	739	11	35	155	184	136
192	426	118	318	546	705	368	580	65	253	283	186
682	460	53	670	719	399	724	440	749	364	450	222
45	202	561	121	389	19	109	164	47	261	223	246
77	558	408	82	476	179	423	535	56	616	8	139
38	760	80	406	521	405	527	298	441	249	454	396
723	447	708	748	91	348	684	120	180	715	314	210
617	655	751	658	683	312	391	168	607	61	598	33
404	424	673	570	601	418	295	642	346	613	694	643
159	661	111	257	50	327	287	674	589	289	39	62
219	285	443	143	30	640	274	462	757	362	124	310
397	753	60	259	214	254	216	81	92	412	548	574
555	293	276	534	695	667	671	605	473	302	665	176
433	594	255	265	591	587	516	456	267	206	518	455
369	319	306	98	431	603	292	395	291	378	602	103
191	638	107	688	40	174	581	356	85	102	469	679
711	347	46	213	409	119	138	566	547	371	342	377
211	628	641	165	554	609	425	78	421	413	108	575
525	612	568	10	280	114	349	520	536	468	517	652
451	649	88	197	198	416	226	110	134	157	185	172
592	556	337	415	66	524	734	75	241	279	113	698
744	654	315	422	351	27	515	320	687	611	686	700
297	572	653	650	361	543	629	761	14	475	619	175
586	51	579	288	355	540	466	637	244	730	606	564
353	36	691	235	230	563	379	398	44	522	567	576
7	403	718	388	363	685	565	758	599	329	133	273
137	459	229	122	625	584	17	275	322	251	187	367
410	334	645	560	158	123	633	583	305	101	151	741
428	537	457	452	313	354	207	737	277	366	675	233
258	309	148	620	479	647	58	161	352	767	323	22
42	432	467	713	588	225	668	177	321	419	104	6
360	188	644	301	135	74	72	461	672	31	307	286
171	67	260	284	530	545	608	544	115	13	662	514

Le tableau suivant (tableau V.2) représente une récapitulation des résultats de chiffrement des données tests présentées ci-dessus :

			Taille donnée (éléments)	Taille clé (bits)	Efficacité	Temps chiffrement (s)	Temps déchiffrement (s)
Données texte	Texte1	Version-Chiff1	1084	15323	15784.0	28.35	2,06
		Version-Chiff2			15030.0	28.02	2,13
	Texte2	Version-Chiff1	1151	15323	19498.0	29.16	2,67
		Version-Chiff2			18712.0	28.97.	2,65
Données images	Lena	Version-Chiff1	131 X 131	7680	139362.0	14.62	3.7
		Version-Chiff2			134074.0	14.21	4.01
	Logo	Version-Chiff1	420 X 395	7680	246426.0	15.06	3.11
		Version-Chiff2			225688.0	15.28	3.81

Tableau V.2. *Résultats obtenus par TabuCrypt.*

V.4. Discussion et évaluation des résultats

La première chose à remarquer d'après les résultats présentés dans la section précédente, est que le temps de chiffrement est proportionnellement dépendant de la taille de l'image à chiffrer. En effet, le temps de chiffrement de l'image Lena (34.826 secondes en moyenne) est plus petit par rapport au temps de chiffrement de l'image Logo (35.59 secondes en moyenne), du fait que la première est plus petite que la deuxième. La même remarque est valable pour le cas des deux données texte Texte1 et Texte2. Cette différence en temps revient au fait que le temps de chiffrement englobe le temps de lecture et de codage de la donnée à chiffrer en plus du temps de calcul, proprement dit, de la solution (temps de la recherche tabou). Le premier (temps de lecture et de codage) est strictement dépendant de la taille de la donnée à chiffrer, tandis que ce n'est pas le cas pour le deuxième (temps du calcul tabou). Donc, c'est surtout le temps de lecture et codage qui fait la différence en temps de chiffrement global d'une donnée par rapport à une autre.

De même, du coté du temps de déchiffrement c'est la phase de décodage de la donnée chiffrée qui fait la différence en temps de calcul.

Maintenant et en comparaison avec nos algorithmes développés et présentés précédemment à travers les deux précédents chapitres, le temps de calcul de ce dernier algorithme se trouve meilleur que le temps de calcul de certains de ces algorithmes et plus mauvais que le temps de calcul de certains autres algorithmes. La figure V.7 positionne le temps de calcul de l'algorithme *TabuCrypt* parmi nos quatre algorithmes PosESecL1, PosESecL2, OEEA et *AntCrypt*.

D'après le résumé de temps de calcul de nos cinq algorithmes développés par utilisation des trois métaheuristiques d'algorithmes évolutionnaires (PosESecL1, PosESecL2 et OEEA), de colonies de fourmis (*AntCrypt*) et de recherche taboue (*TabuCrypt*), nous constatons que *TabuCrypt* possède un temps de calcul meilleur que celui de PosESecL1 et de PosESecL2 mais plus grand que celui de OEEA et celui de *AntCrypt*.

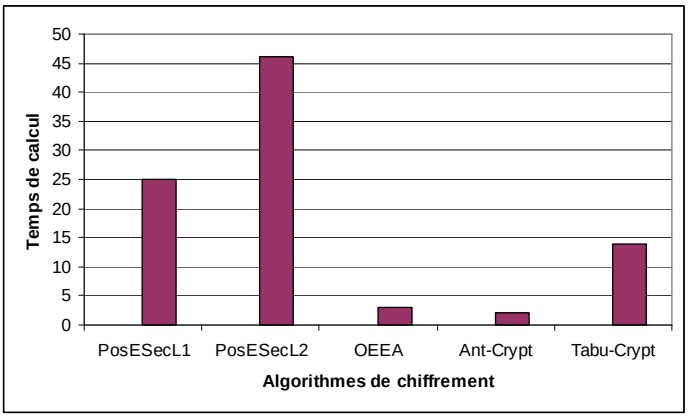


Figure V.7. *Comparaison des temps de calcul.*

Et comme *TabuCrypt* est un algorithme bâti par exploitation d'une métaheuristique, et nous l'avons expliqué précédemment, l'adaptabilité et l'efficacité d'application de telles méthodes dans un tel domaine à cause de leur large utilisation de l'aléatoire a été la source de notre motivation d'application de métaheuristique. L'algorithme développé sera, ainsi, doté d'un grand pouvoir confusionnel pour compliquer le plus possible la tâche de cryptanalyse.

En effet, l'aspect non déterministe innové à travers nos approches proposées, et par conséquent celle, ici, présentée, représente l'un des points forts de l'algorithme *TabuCrypt* tout comme nos autres algorithmes leur assurant, ainsi, une résistibilité contre les attaques différentielles. Ceci est assuré par une élection aléatoire, sur l'ensemble des générations, des voisins des éléments codant une solution intermédiaire pour calculer une autre solution intermédiaire ou une solution finale qui soit satisfaisante.
De même, l'attaque statistique sera presque impossible à appliquer grâce à ce dernier point en plus du fait que les données originales et leurs version chiffrées sur une instance, seront presque toutes différentes comme le montre le tableau suivant (tableau V.3) résumant les valeurs d'application des mesures de similarité NPCR, MAE et MSE.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.8927	0.97	0.68
Lena-version chiffrée 2	0.9104	1.05	0.71

Tableau V.3. *Niveaux de confusion de Tabu-Crypt.*

De son tour, l'attaque exhaustive sera mise à l'écart grâce à la taille de clé qui est largement sécurisée et qui est de taille égale à 15323 bits dans le cas de manipulation de données texte et qui égale à 7680 bits dans le cas de manipulation de données images.

Les attaques fréquentielle sont aussi pénalisées du fait, d'un coté, que ces dernières étudient les fréquences d'apparition des caractères dans un message (précisément un texte) écrit suivant une langue naturelle, alors que notre algorithme *TabuCrypt* peut traiter toutes sortes de message constitué de caractères du code Unicode. D'un autre coté, le message chiffré peut être constitué de caractères complètement ou partiellement différents de ceux constituant le message en clair.

V.5. Conclusion

La recherche Tabou est une méthode d'optimisation qui utilise la notion de voisinage d'une solution. Cette méthode permet, à partir d'une solution initiale, de visiter un ensemble de voisins. Ces voisins seront évalués et permettent l'évolution, à travers des règles spécifiques, vers une solution finale.

Ainsi et dans ce chapitre, nous avons présenté notre nouvel algorithme de chiffrement tabou, *TabuCrypt*, qui opère suivant le mode de chiffrement à base d'occurrence expliqué et validé dans le troisième chapitre.

Nous avons présenté en détails les différentes étapes du processus développé et nous l'avons testé sur différentes données test. Ainsi, notre objectif qui se résume en l'exploitation d'un voisinage en recherche locale de type taboue pour la manipulation de données texte et images, est atteint dans certains cotés vu les qualités de la méthode tabou (simplicité du principe, performance…) en donnant lieu à des résultats de bon niveau de confusion. Toutefois, le temps de convergence du processus de résolution est relativement lent par rapport à certains de nos précédents algorithmes proposés : OEEA et *AntCrypt*. Cela revient à l'aspect parallèle encapsulé à travers le principe soit des AEs ou des algorithmes de colonies de fourmis.

Conclusion et perspectives

Au cours de cette thèse nous avons proposé une nouvelle approche de cryptage de données texte et images par exploitation de métaheuristiques parmi lesquelles, nous nous sommes intéressés à celle inspirée du monde biologique et désignée par « approches biomimétiques ». Il s'agit des algorithmes évolutionnaires inspirés des principes de l'évolution naturelle des espèces. Par la suite, nous avons abordé une approche inspirée de modèles d'organisation naturels observés dans les sociétés animales, en particulier, dans une colonie de fourmis. Ces deux méthodes exploitées appartiennent à la catégorie des métaheuristiques à population. De même et pour l'autre catégorie qui est celle des métaheuristiques à trajectoire, nous sommes aussi arrivés à proposer un autre algorithme de cryptage utilisant le principe d'une recherche tabou.

Ainsi et après avoir étudié un panel assez diversifié des techniques de cryptage et avoir analysé les problèmes de sécurisation du transfert, nous avons élaboré nos propositions de stratégies de résolution d'un tel problème. Autrement dit et afin de détailler nos apports, notre thèse a été organisée autour de deux parties dont l'objectif de la première est de situer le lecteur dans le contexte de notre travail pour lui permettre de suivre les démarches réalisées dans cette étude. Dans la deuxième, nous avons détaillé la conception, la réalisation et l'évaluation de l'approche de résolution proposée.

Notre première contribution a porté sur l'élaboration et la conception de nouveaux algorithmes de chiffrement efficaces et robustes qui traitent le problème de taille de clés leur permettant, ainsi, d'être loin des attaques exhaustives.

Pour ce faire, nous avons commencé par une étude approfondie du problème tout en essayant de le ramener en un problème d'optimisation par une formulation adéquate aux métaheuristiques à appliquer : métaheuristique des algorithmes évolutionnaires, métaheuristique des algorithmes de colonies de fourmis et celle de recherche tabou. Pour la première, les obstacles à franchir étaient surtout la définition d'un codage adéquat de solutions. Dans ce sens, deux modes de codage différents ont été proposés : codage à base de positions et un deuxième à base d'occurrences donnant naissance, ainsi, à trois algorithmes de chiffrement, PosESecL1, PosESecL2 et OEEA de qualités différentes dont celui opérant suivant le mode de chiffrement par occurrences (OEEA) était le plus efficace et le plus robuste.

L'application de la deuxième métaheuristique a démontré l'efficacité du codage à base d'occurrences. À ce stade aussi, il a fallu définir avec précaution certains paramètres tels que la fonction d'évaluation, le mécanisme de manipulation de phéromone (incrémentation et évaporation) ainsi que celui de génération de la clé de chiffrement qui doit doter l'algorithme développé, *AntCrypt*, d'une résistance contre les attaques exhaustives vu que cet algorithme est de type symétrique.

De même, l'exploitation de la troisième métaheuristique a, aussi, abouti au développement d'un cinquième algorithme de chiffrement, *TabuCrypt*. Cela a nécessité la projection des composants d'une telle métaheuristique sur le problème de cryptage, à savoir la

liste tabou, stratégie de choix de voisins, table de hachage et les mécanismes avancés en recherche tabou (Intensification, diversification et aspiration). Les résultats étaient satisfaisants que ce soit sur le plan cryptographique que cryptanalytique, toutefois, il nécessite plus de temps de calcul que certains de nos autres algorithmes (*OEEA* et *AntCrypt*).

Notre deuxième contribution était l'innovation d'algorithmes de chiffrement non déterministes en se bénéficiant de l'aspect aléatoire de la sélection naturelle ou du déplacement des fourmis. En effet, l'aléatoire représente l'ennemi des cryptanalystes. En appliquant toutes les attaques redoutables, il est donc quasiment impossible de parvenir à l'information initiale ou au moins d'établir une relation entre les données chiffrées puisqu'on obtient toujours des versions chiffrées différentes à chaque application de l'algorithme même lors d'un chiffrement sur plusieurs instances d'une même donnée originale.

En fait, à travers notre étude nous sommes arrivés à présenter les métaheuristiques comme des méthodes de résolution approchées se voulant simples et adaptables à tout type de problèmes. Leur capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la solution trouvée. Du point de vue de la recherche opérationnelle, cet inconvénient n'est pas toujours un problème, tout spécialement quand une seule approximation de la solution optimale est recherchée. En nous situant dans ce cas, nous aurions pu mettre en évidence l'avantage de ces nouvelles approches pour résoudre le problème de cryptage de données texte ou images.

Toutefois et malgré que nos algorithmes développés soient de bonne qualité cryptographique, plusieurs autres idées et applications en relation avec ce présent travail pour le compléter, restent à concrétiser. En premier lieu, il est urgent d'aborder le problème de métaréglage [Dréo, 2004], non seulement pour régler automatiquement les paramètres des métaheuristiques, mais aussi pour maîtriser leurs fonctionnements.

De même et malgré que les algorithmes évolutionnaires ou même les algorithmes de colonies de fourmis permettent d'obtenir des solutions pas toujours optimales, mais possiblement satisfaisantes à de nombreux problèmes complexes, leur puissance d'optimisation est liée à la puissance sous-jacente des machines les exécutant.

Comme ce sont des algorithmes inspirés de la nature, ils sont fortement parallélisables et il se trouve que le récent développement des cartes graphiques, dédiées au calcul de rendu 3D dans un premier temps, puis généralisées au milieu des années 2000 avec les *General Purpose Graphic Processing Units (GPGPU)*, permet d'utiliser le grand nombre de cœurs de calcul qu'elles comportent pour autre chose que du rendu graphique.

Dans [Mait, 2011], l'auteur détaille et étudie comment les algorithmes évolutionnaires peuvent bénéficier de ces nouvelles architectures, avec leurs particularités. Plusieurs algorithmes ont été présentés, qui permettent l'utilisation efficace de ces processeurs pour l'évolution artificielle, que ce soit pour les algorithmes génétiques et stratégies d'évolution, mais aussi pour la programmation génétique. Différentes stratégies de parallélisation ont été utilisées, dont l'une nécessitant le développement d'un opérateur parallèle de réduction de la population. L'adaptation des paramètres de ces algorithmes pour permettre le portage de ces derniers sur GPGPU est détaillée, ainsi que certains points impactant directement les performances de ces portages.

Ainsi, de tels schémas peuvent être envisagés pour une parallélisation de nos algorithmes développés surtout PosESecL2 possédant un bon niveau confusionnel mais souffrant d'une lenteur de calcul vu la grande taille de chromosomes manipulés codant des données à chiffrer de grandes tailles.

D’un autre coté et vu que nos algorithmes de cryptage développés sont de type symétrique, une méthode de communication sécurisée de clé de chiffrement secrète peut être envisagée lors de la manipulation de données images. Il s’agit de marquer l'image cryptée dans des régions spécifiques, ou des régions déterminées pour l'utilisateur. Nous pouvons également utiliser des méthodes de marquage (watermarking) sans perte pour insérer la clef cryptée dans l'image cryptée. Il serait souhaitable de développer des évaluations sur la robustesse concernant les attaques. Ainsi, nous souhaitons faire des études concernant la cryptanalyse de nos méthodes pour classifier son niveau de sécurité face aux attaquants.

Par ailleurs, nos perspectives de recherche à long terme s’orientent vers l’étude et le développement d’autres méthodologies de cryptage de données. Nous nous intéresserons, en particulier à :

- La métaheuristique des automates cellulaires inventée par Ulam [Ulam, 1950] et Von Neumann [VonN, 1966] et basée sur le concept de la vie artificielle. Ce concept s’appuie sur des règles extrêmement simples permettant de construire des structures très complexes et esthétiques, d’où sa ressemblance avec le principe d’auto-organisation en biologie animale, notamment avec les mécanismes de la colonie de fourmis. Ces règles peuvent stipuler, par exemple, qu'une naissance nécessite un certain rassemblement de population, que les cellules ne peuvent survivre à un trop grand isolement et qu'une trop forte concentration les étouffe. Donc, nous comptons adapter ces règles au problème de cryptage où de génération en génération, l’application de ces règles permettra l’émergence d’une solution au problème.

- L’approche des « hyper-heuristiques » introduite dans [Cowl, 2000], [Cowl, 2002], [Burk, 2003], [Burk, 2005] et qui consiste à travailler dans un espace de recherche composé de métaheuristiques afin d’optimiser le choix de la métaheuristique à utiliser pour un problème donné. Il s’agit, donc, de mettre en confrontation un ensemble de métaheuristiques différentes pour en sélectionner la plus optimale.

Ministère de l'Enseignement Supérieur et de la recherche Scientifique

BADJI MOKHTAR-ANNABA UNIVERSITY
NIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار – عنابة

Faculté des sciences de l'ingénieur
Département d'informatique

THESE

Présentée en vue d'obtention du diplôme de ***DOCTORAT en Informatique***

Sécurisation évolutionnaire du transfert d'images

Option
Informatique

Par
SOUICI Ismahane

Directeur de Thèse
SERIDI Hamid Pr. Univ. 08 mai 1945 Guelma

Devant le jury

Président :	**KHADIR Tarek**	Pr. Univ. Badji Mokhtar Annaba
Examinateurs :		
	FARAH Nadir	Pr. Univ. Badji Mokhtar Annaba
	MEROUANI Hayet	MC. Univ. Badji Mokhtar Annaba
	KHOLADI Khiredine	Pr. Univ. Mentouri Constantine
	CHIKHI Salim	Pr. Univ. Mentouri Constantine

Année universitaire : 2012/2013

Remerciements

En premier lieu, je tiens à exprimer ma profonde gratitude à Monsieur SERIDI Hamid, Professeur à l'université 08 mai 1945 de Guelma, pour avoir dirigé ma recherche depuis le Magister et pour la confiance et l'intérêt qu'il m'a témoignés tout au long de l'élaboration de cette thèse. Je lui suis très reconnaissante pour le suivi régulier et formateur reçu durant ces quatre années de Doctorat, et bien avant, durant les trois années de Magister. Je le remercie pour la démarche scientifique rigoureuse et l'esprit d'auto-critique qu'il a su m'inculquer. J'espère avoir été à la hauteur de ces espérances.

J'adresse mes remerciements les plus sincères à Monsieur KHADIR Tarek, Professeur à l'université Badji Mokhtar d'Annaba, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Mes vifs remerciements vont aux membres du jury qui ont accepté de prendre de leur temps pour examiner mon travail : Monsieur FARAH Nadir, Professeur à l'université Badji Mokhtar d'Annaba, Madame MEROUANI Hayet, Maître de Conférences à l'université Badji Mokhtar d'Annaba, Monsieur KHOLLADI Khiredine, Professeur à l'université Mentouri de Constantine et Monsieur CHIKHI Salim, Professeur à l'Université Mentouri de Constantine.

Au cours de ces quatre années de recherche, j'ai eu à effectuer un stage de courte de durée en 2010, au laboratoire CReSTIC de l'université de Reims Champagne Ardenne, France, sous la direction de Monsieur AKDAG Herman, Professeur à l'université de Reims Champagne Ardenne, France. Qu'il trouve ici, l'expression de mes remerciements les plus sincères, pour son chaleureux accueil. Je tiens à remercier, aussi, Monsieur Cyril DE RUNZ, Maitre de Conférences à l'université de Reims Champagne Ardenne, France pour sa disponibilité, ainsi que pour ses discussions fructueuses que j'ai eues avec lui.

Merci à celles et ceux qui auront à lire tout ou une partie de ce manuscrit et qui y trouveront un intérêt quelconque.

Enfin, un grand MERCI à ma famille et, en particulier, mes parents pour le soutien et les encouragements qu'ils ont su m'apporter pour arriver au terme de cette thèse.
J'espère les honorer avec ce travail.

الملخص

في يومنا هذا أصبحت شبكات المعلوماتية معقدة و التنصتات غير القانونية عليها قائمة. لهذا أصبح من الضروري حماية المعلومات الحساسة المنقولة عبر هذه الشبكات, أين يعد علم التشفير واحد من الحلول الجد فعالة, الذي و على الرغم من كل التطورات التي سجلت بهذا المجال ماتزال بعض النقائص مسجلة خاصة ما يتعلق بحجم المفتاح المستعمل و مقاومة الهجمات المتطورة.

في هذا السياق ينصب موضوع بحثنا للدكتوراه حيث نسعى إلى تطوير خوارزمات تشفير باستعمال métaheuristiques التي تعد من الطرق التي توفر حلولا تقريبية و تفتح مجالات تصميمية لطرق حل مهمة لمشاكل التحسين.
هذه الأخيرة تنقسم إلى قسمين : metaheuristiques التي تستعمل مجموعة حلول métaheuristiques التي تستعمل حلا واحدا. وبما أن métaheuristiques توظف طابع العشوائية في معظم مراحل عملها و الذي يعد عنصرا مهم إستعماله في مجال التشفير لتعقيد مهمة المهاجمين, لهذا وقع إختيارنا على هذه الطرق لحل مشكل تشفير المعلومات النصية و الصور.

إذن وضمن الصنف الأول الذي يشمل métaheuristiques التي تستعمل مجموعة حلول قمنا بتطوير ثلاث خوارزمات تعتمد الطريقة التطورية المستوحاة من نظرية "دارون" و هذا حسب نوعين مختلفين من التشفير (التشفير المعتمد على الوضعيات و التشفير المعتمد على الظهور), بالإضافة إلى خوارزم رابع و الذي يستعمل métaheuristique مستوحاة من comportement الحقيقي للنمل.
بالنسبة للصنف الثاني والذي يضم métaheuristique التي تستعمل حلا واحدا, توصلنا إلى تكييف طريقة البحث الطابوه إستعمالها في حل المشكل لمدروس و هكذا قمنا بتطويرخوارزم تشفير طابوه خامس.

أخيرا قمنا بتجريب ومقارنة كل الخوارزمات المقترحة فيما بينها وبالنسبة لعدد من مراجع التشفير حيث أظهروا خصائص جيدة سواءا فيما يتعلق بدرجة الغموض أو مقاومة الهجمات الأكثر تطورا. قطة أخرى جد مهمة توصلنا إليها عبر عملنا هذا هي استحداث خوارزمات non déterministes , نقطة تساهم بدرجة كبيرة في زيادة قوة مثل هذه الخوارزمات.

Résumé

Aujourd'hui, les réseaux informatiques sont complexes et les écoutes illégales possibles. Il se pose donc un réel problème quant à la sécurité lors de la transmission de données. Pour des raisons éthiques, le transfert des données délicates ne peut se faire avec un tel risque et doit donc se protéger. La protection la plus adaptée pour ce type de communication réside dans la cryptographie. Cependant et malgré toutes ses évolutions et ses mises en œuvre, elle est toujours entravée par quelques défauts citant en particulier la taille des clés et la résistance contre les attaques avancées.

Ainsi, notre travail de thèse porte sur le développement de nouveaux algorithmes cryptographiques par exploitation des métaheuristiques constituant une partie importante des méthodes approchées et ouvrant des voies très intéressante en matière de conception de méthodes heuristiques pour l'optimisation. Ces dernières se scindent en deux catégories : les métaheuristiques à population de solutions et les métaheuristiques à une seule solution ou encore dites à trajectoire. Et vu que les métaheuristiques exploitent un aspect pseudo-parallèle durant différentes étapes de leurs schémas opératoires, chose qui est très intéressante à exploiter dans le domaine cryptographique pour compliquer de façon considérable la tâche des cryptanalystes, donc notre choix s'est porté sur l'utilisation de telles méthodes pour résoudre le problème de cryptage de données texte et images.

En effet, et pour la première catégorie de métaheuristiques à population, nous avons développé trois algorithmes de chiffrement exploitant les principes évolutionnaires inspirés de la théorie darwinienne, mais opérants suivant deux modes de chiffrement différents (chiffrement à base de positions et un chiffrement à base d'occurrences) ; et un quatrième algorithme utilisant une métaheuristique inspirée du comportement réel des fourmis (ACO).

Pour la deuxième catégorie qui est celle de métaheuristiques à trajectoire, nous avant arrivé à adapter la méthode de recherche taboue et l'exploiter pour résoudre notre problème étudié et ainsi développer un cinquième algorithme de cryptage tabou.

Les algorithmes proposés ont été évalués et comparés entre eux et avec les principaux algorithmes standards de cryptage où ils ont montré de bonnes caractéristiques relatives soit à leur degré confusionnel ou à leur résistibilité aux attaques les plus avancées. L'autre point crucial résultant de nos travaux de thèse était l'innovation d'algorithmes de chiffrement non déterministes, chose qui augmente considérablement leur robustesse.

Mots clés : Cryptage, attaques avancées, métaheuristiques, algorithmes évolutionnaires, recherche tabou, algorithme de colonies de fourmis.

Abstract

Today, computer networks are complex and the illegal wiretapping is possible. This raises a real problem of security when transmitting data. For ethical reasons, the transfer of delicate data can not be done with such a risk and must protect themselves. Protection best suited for this type of communication lies in cryptography. However, despite all its developments and implementations, it is still hampered by some shortcomings citing in particular the key size and resistance against advanced attacks.

Thus, the object of our thesis is the development of new cryptographic algorithms by exploitation of metaheuristics constituting a significant portion of the approximate methods and opening very interesting channels in the design of heuristic methods for optimization. The latter fall into two categories: metaheuristics of solutions population and metaheuristics of one solution or also tell trajectory metaheuristics. And since metaheuristics exploit a pseudo-parallel aspect during different stages of their operating patterns, something that is very interesting to use in the cryptographic domain to significantly complicate the task of the cryptanalysts, so our choice fell on the use of such methods to solve the problem of the encryption of text and images data.

Indeed, for the first class of metaheuristics of population, we developed three encryption algorithms exploiting the evolutionary principles inspired by the Darwinian theory but operating in two different encryption modes (positions based encryption and occurrences based encryption) and a fourth algorithm using a metaheuristic inspired by the ants real behavior (ACO).

For the second category which is trajectory metaheuristics, we arrived to adapt the method of tabu search and use it to solve the problem at hand. Thus, a fifth taboo encryption algorithm was developed.

The proposed algorithms were evaluated and compared among themselves and with principal encryption standard algorithms where they showed good characteristics for either their confusional degree or their resistibility against the most advanced attacks. The other crucial point arising from our thesis work was the innovation of non-deterministic encryption algorithms, something that greatly increases their robustness.

Table des matières

Chapitre III : CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

Chapitre IV : CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

Chapitre V : CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Dès que les hommes ont appris à communiquer, ils ont trouvé des moyens d'assurer la confidentialité d'une partie de leurs communications : l'origine de la cryptographie remonte sans doute aux origines de l'homme. En effet, le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire de les rendre inintelligibles sans une action spécifique.

Du bâton nommé « scytale » au Vie siècle avant JC, en passant par le carré de Polybe ou encore le code de César, on assista au développement plus ou moins ingénieux de techniques de chiffrement expérimentales dont la sécurité reposait essentiellement dans la confiance que leur accordaient leurs utilisateurs. Après la Première Guerre mondiale a lieu une première révolution technologique.

Mais ce n'est qu'à l'avènement de l'informatique et d'Internet que la cryptographie prend tout son sens. Les efforts conjoints d'IBM et de la NSA conduisent à l'élaboration du DES (Data Encryption Standard), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XXe siècle. À l'ère d'Internet, le nombre d'applications civiles de chiffrement (banques, télécommunications, cartes bleues…) explose. Le besoin d'apporter une sécurité accrue dans les transactions électroniques fait naître les notions de signature et authentification électroniques. La première technique de chiffrement à clef publique sûre (intimement liée à ces notions) apparaît : le RSA.

Donc, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information. C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à ramener et à modéliser le problème de cryptage comme un problème d'optimisation.

En effet, l'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solutions algorithmiques efficaces valables pour toutes les données.

Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles que les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens.

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques*. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, les chromosomes et les mécanismes d'intensification et de diversification pour la métaheuristique des algorithmes évolutionnaires), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Les métaheuristiques sont représentées essentiellement par les *méthodes de voisinage* comme le recuit simulé et la recherche tabou, et les *algorithmes évolutifs* comme les algorithmes génétiques et les stratégies d'évolution. Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi et hormis cette introduction et la conclusion générale qui reprennent les travaux de l'ensemble des chapitres et quelques perspectives majeures pour la poursuite de ce travail, le manuscrit est divisé en deux grandes parties.

Un état de l'art aussi complet que possible relatif soit au problème étudié qui est celui de cryptage, soit aux approches de résolutions exploitées qui sont les métaheuristiques, fera l'objet des deux premiers chapitres formant la première partie. En effet, le premier chapitre traite la cryptographie depuis sa première apparition jusqu'à nos jours en présentant un bref aperçu historique permettant de comprendre la distinction entre les trois disciplines (cryptologie, cryptographie et cryptanalyse), les fonctionnalités de base de la cryptographie (confidentialité, authentification, intégrité et la non-répudiation) et leurs différentes catégories (la cryptographie symétrique, asymétrique, hybride et quantique) tout en citant les fameux algorithmes appartenant aux principales catégories.
De son tour, le deuxième chapitre présente une introduction aux métaheuristiques tout en citant les concepts de base ainsi qu'une classification des métaheuristiques illustrée d'exemples d'algorithmes de chacune des classes.

Dans la deuxième partie de notre recherche, nous proposons un nouvel axe de recherche que représente l'application des métaheuristiques pour résoudre le problème de cryptage de données textes et images. Ainsi, les deux catégories de métaheuristiques ont été exploitées : métaheuristiques à population et les métaheuristiques à trajectoire.
Dans le cadre de la première, nous avons choisi d'exploiter deux métaheuristiques qui sont les algorithmes évolutionnaires (AEs) et les algorithmes de colonies de fourmis. Le principal avantage de ces méthodes heuristiques vient de leur capacité à traiter le problème de cryptage en ne possédant qu'un minimum d'informations sur celui-ci. Chose qui est primordiale dans ce problème pour augmenter la confusion des algorithmes développés.

Ainsi, le troisième chapitre résume l'application d'AEs où les différentes étapes partant du codage sont explicitées. En effet, pour un problème à résoudre, il est nécessaire d'adapter la représentation des individus aux objectifs recherchés. Cette adaptation permet de faire converger l'AE plus ou moins rapidement et de tenir compte naturellement des contraintes de la modélisation du problème. Dans le cas présent, l'espace de recherche que nous cherchons à optimiser est l'ensemble représenté par les solutions possibles du problème de cryptage et par une fonction d'évaluation de chaque solution.

Le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, les algorithmes ont été groupés en deux catégories distinctes suivant les modes de chiffrement utilisé implémentant différents codages : chiffrement à base de position ou chiffrement à base d'occurrences.

D'une manière globale, nous allons définir un individu de la population comme une donnée chiffrée possible. Cet individu doit pouvoir être évalué numériquement par la fonction d'évaluation proposée. Cette fonction de fitness calcule la qualité d'une donnée chiffrée en fonction du besoin de confusion. Nous montrerons aussi l'application de notre méthode sur quelques exemples d'images de différentes tailles. Une interprétation et une discussion des résultats obtenus seront abordées pour pouvoir, par la suite, comparer les nouveaux algorithmes proposés de cryptage évolutionnaire où l'algorithme opérant suivant le mode de chiffrement par occurrences a montré une efficacité en termes de temps de calcul, de pouvoir de confusion, de résistibilité aux attaques et de longueur de clé meilleure que celles des algorithmes opérants suivant le mode de chiffrement par positions.

Cette conclusion a été démontrée par l'application d'une deuxième méthode de résolution exploitant les algorithmes de colonies de fourmis illustrée sur un quatrième chapitre où les résultats obtenus été très satisfaisants.

Au niveau du cinquième chapitre, un autre algorithme de cryptage a été proposé s'inscrivant cette fois-ci, sous la deuxième catégorie de métaheuristiques dans le but d'explorer l'espace de recherche par exploitation de la notion de voisinage. Il s'agit d'un algorithme utilisant le principe d'une recherche tabou pour pouvoir choisir la configuration représentant la meilleure solution au problème de cryptage de données texte ou images.

Dans chacun des chapitres décrivant nos algorithmes de cryptage proposés, nous présentons des exemples et résultats expérimentaux sur des données tests. Nous avons aussi effectué des bilans en faisant ressortir les avantages et faiblesses des méthodes développées.

CHAPITRE I

CRYPTOGRAPHIE

I.1. Introduction

Depuis les temps historiques les plus reculés, l'homme a perçu le besoin de cacher, de dissimuler ou faire mystère des informations personnelles ou confidentielles, et cela bien avant l'ère informatique afin de les rendre inintelligibles à des lecteurs indésirables. C'est pourquoi, les codes ont existé.

Les origines de la cryptographie semblent remonter à plus de 4000 ans en Egypte. Plusieurs indications archéologiques tendent à montrer que les « écritures secrètes » sont en fait anciennes que l'invention de l'écriture elle-même. Polybius développa un système de codage des lettres de l'alphabet consistant à remplacer chaque lettre de l'alphabet par deux nombres, donnant la ligne et la colonne où se trouve cette lettre dans une matrice. Jules César utilisait une simple méthode de substitution de lettres pour communiquer secrètement avec ses généraux : c'est un chiffre par décalage,…etc.

C'est cependant au cours de la seconde guerre mondiale que la cryptographie s'inscrit véritablement comme élément central des stratégies militaires. Le cas le plus connu est certainement l'histoire entourant le décodage du code Enigma par les Polonais et les Britanniques. Une conjonction d'espionnage classique et d'efforts de mathématiciens polonais permet de déduire la clé utilisée et ainsi de décoder les messages encodés avec Enigma. On trouve apparemment bien moins de détails sur les efforts cryptographiques durant la guerre froide, probablement parce que ces informations sont encore « Top Secret ».

On en est maintenant à l'époque moderne où le champ d'application de la cryptologie s'est élargi et a trouvé un regain d'actualité avec toutes les applications nouvelles suscitées par l'utilisation de l'Internet. La révolution d'Internet et l'utilisation de plus en plus d'informations massives sous forme numérique facilitent les communications et rendent de ce fait plus fragiles les informations que l'on détient, c'est pourquoi il devient nécessaire de protéger le contenu de certains messages des inévitables curieux. En effet, les réseaux ouverts créent des brèches de sécurité et il est plus aisé à un adversaire d'accéder aux informations. Dans une communication à distance, des questions cruciales se posent et leurs réponses s'imposent ; comment être sur :

- que l'on parle à la bonne personne (authenticité),
- que nos propos ne sont pas altérés (intégrité),
- que la conversation n'est pas espionnée (confidentialité),
- de l'identité de l'émetteur (non répudiation) ?

Indéniablement, avec l'essor fulgurant des nouvelles technologies, le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements.

Alors que la cryptographie consiste à sécuriser les données, tandis que, la cryptanalyse est l'étude des informations cryptées afin d'en découvrir le secret. Grâce à la cryptanalyse, les militaires ont pu mener leurs guerres en découvrant les correspondances de leurs ennemis et en contrôlant les réseaux de communications mais d'autre part si elle est utilisée par une personne mal intentionnée il peut générer des dégâts onéreux aux entreprises ou aux sociétés.

I.2. Les Fondements de la Cryptographie

I.2.1. Terminologie

Comme toute science, la cryptographie possède son propre langage. Dans ce qui suit les mots clés de domaine cryptographique.

I.2.1.1. Cryptographie

Le terme cryptographie vient en effet des deux mots grecs : *Kruptus* qu'on peut traduire comme secret et *Graphein* pour écriture. Ainsi la cryptographie est l'art de dissimuler une information écrite en clair (plain text) en *cryptogramme* (cipher text) pour qu'elle soit incompréhensible que par son destinataire légitime par le biais d'une clé appelé « clé de chiffrement » (processus de *chiffrement*). Pour rendre l'information à nouveau intelligible par le biais d'une clé appelé « clé de déchiffrement » le processus inverse est appliqué (processus de *déchiffrement*).

On distingue généralement deux types de clefs :

- **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

Le schéma suivant illustre le processus cryptographique :

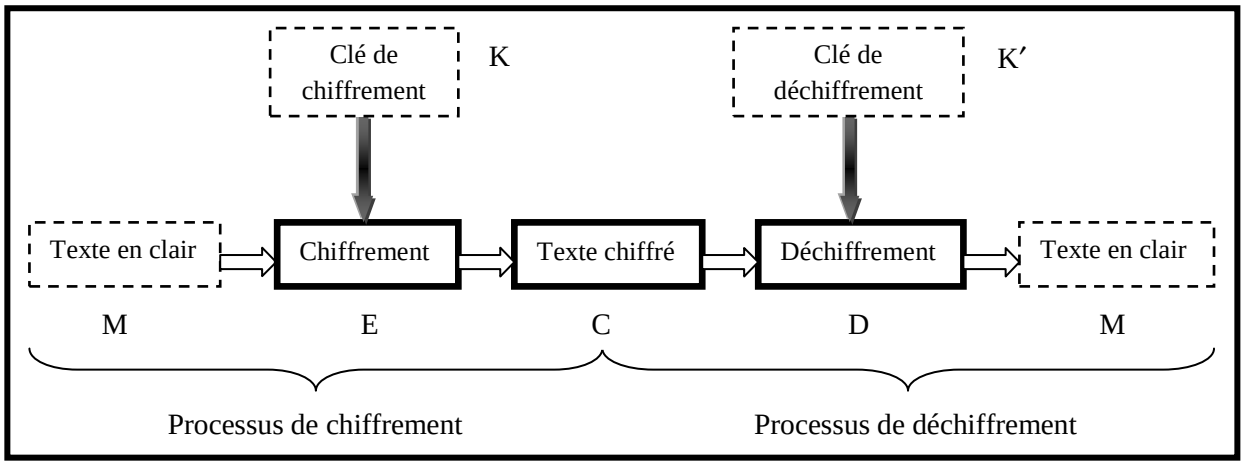


Figure I.1. *Processus cryptographique.*

I.2.1.2. Cryptanalyse

a. Définition

C'est l'art d'étude des crypto systèmes en cherchant leurs failles et leurs vulnérabilités afin de retrouver des messages clairs correspondant à des messages chiffrés sans avoir à connaître les clés utilisées dans le chiffrement. Lorsque tous les éléments de la méthode utilisée pour coder des messages sont repérés, on dit qu'on a **cassé** ou **brisé** le système cryptographique utilisé. Plus un système est difficile à briser, plus il est sûr.
La personne qui pratique la cryptanalyse est appelée : cryptanalyste. Il tente à décrypter le message chiffré pour découvrir son secret. On a distingué entre le verbe « décrypter » et « déchiffrer » puisque ce dernier est réservé pour le déchiffrement par le destinataire légitime.

b. Types de cryptanalyse

On distingue plusieurs types d'attaques :

- ***Attaque sur texte chiffré seul (ciphertext-only)*** : l'attaquant a seulement la possibilité d'intercepter un ou plusieurs messages chiffrés. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- ***Attaque à texte clair connu (known-plaintext attack*)** : se base sur la connaissance d'une partie du texte en clair pour déduire le reste du message. La tâche est de retrouver la clef utilisée pour chiffrer ce message.
- ***Attaque à texte clair choisi (chosen-plaintext attack)*** : se base sur la possibilité de choisir un texte clair et d'obtenir son chiffrement et en ayant la possibilité de générer les versions chiffrées de messages clair avec un algorithme considérer comme une **boîte noire** tel que les algorithmes à clé publique puisque l'algorithme est public.
- ***Attaque à texte chiffré choisi (chosen-ciphertext attack)*** : le cryptanalyste possède des messages chiffrés et essaye de les déchiffrer de son choix. Sa tâche est de retrouver la clef.

c. Familles d'attaques cryptanalytiques

Il existe plusieurs familles d'attaques cryptanalytiques, les plus connues sont les suivantes :

- **L'analyse fréquentielle** : examine les répétitions des lettres du message chiffré afin de trouver la clé. Cette technique est découverte par Al-Kindi au IXe siècle [www1] contre les chiffrements mono-alphabétiques. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est basée sur le fait que, dans chaque langue, certaines lettres ou combinaisons de lettres apparaissent avec une certaine fréquence.
- **L'attaque par dictionnaire** : le mot testé est pris dans une liste prédéfinis contenant les mots de passe les plus courants et aussi des variantes de ceux-ci. Ces listes sont généralement dans toutes les langues les plus utilisées, elles contiennent des mots existants ou des mots diminutifs (par exemple « powa » pour « power » ou « K7 » pour « cassette »). Elle est souvent couplée à l'attaque par force brute.
- **L'attaque par force brute** : s'appuie sur le cassage d'un mot de passe en testant tous les mots de passe possibles. C'est le seul moyen de récupérer la clé dans les algorithmes les plus modernes et encore inviolés comme AES.
- **La cryptanalyse linéaire** : c'est une attaque à texte clair inventée par le japonais Mitsuru Matsui [www1]. L'idée est de trouver des approximations linéaires entre les bits de sortie, les bits d'entrée et les bits de la clé. Si certaines de ces approximations apparaissent avec une probabilité suffisante, on a alors démontré que la correspondance

entre entrée et sortie n'est pas purement aléatoire. Le nombre de clés à envisager pour déchiffrer le message est restreint.

- **La cryptanalyse différentielle** : découverte par deux cryptologies israéliennes : Bihan et Shamir [www1]. Elle consiste à comparer les sorties de l'algorithme quand on lui met en entrée deux messages ayant une différence fixe. On étudie comme variant les sorties si les deux messages ne diffèrent que par un seul bit. Si en déplaçant ce bit à l'intérieur des messages, certains bits des sorties restent inchangés, on a alors trouvé une faille dans l'algorithme et celui-ci est attaquable.

I.2.1.3. Cryptologie

La cryptologie du grec kruptos « secret, caché » et logos « discours » qui embrasse à la fois la cryptographie et la cryptanalyse. Elle se partage entre la cryptographie, qui inclut la conception des mécanismes destinés à assurer les fonctions suivantes : confidentialité, intégrité, authentification et traçabilité (non répudiation), et la cryptanalyse dont le but est de déjouer les protections ainsi mises en place.

I.2.1.4. Fonction de hachage

Une fonction de hachage est une fonction mathématique qui à partir d'un message (d'une donnée) génère une autre chaîne, l'empreinte (généralement plus courte). Cette empreinte est très sensible au texte initial (une petite modification du texte provoque une grande modification d'empreinte).

I.2.1.5. Signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message. Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur.

I.2.1.6. Les certificats

Pour assurer l'intégrité des clés publiques, ces dernières sont publiées avec un certificat. Un certificat (ou certificat de clés publiques) est une structure de données qui est uniquement signée par une autorité certifiée.

CA, Certification Autorité, est une autorité en qui les utilisateurs peuvent faire confiance. Il contient une série de valeurs, comme le nom du certificat et son utilisation, des informations identifiants le propriétaire et la clé publique, la clé publique elle-même, la date d'expiration et le nom de l'organisme de certificats. Le CA utilise sa clé privée pour signer le certificat et assure ainsi une sécurité supplémentaire. Si le récepteur connaît la clé publique du CA, il peut vérifier que le certificat provient vraiment de l'autorité concernée et, ainsi, de s'assurer que le certificat contient des informations viables et une clé publique valide.

I.2.2. Fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur fiabilité et confidentialité au travers d'un canal peu sûr (téléphone, réseau informatique ou autre). Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions en plus de la fiabilité et la confidentialité. Il s'agit de garantir l'intégrité et l'authenticité des données échangées. Les postulats de sécurité associés à la cryptographie sont :

a. Intégrité

Vérifier l'intégrité des données consiste à déterminer si les données, ressources, traitements ou services n'ont pas été altérées durant la communication de manière fortuite ou intentionnelle.

b. Confidentialité

La confidentialité est le maintien du secret des informations. Elle consiste à rendre l'information discrète ou inintelligible à d'autres personnes que les seuls acteurs de la transaction. La confidentialité peut être vue comme la «protection des données contre une divulgation non autorisée» [Gher, 2004].

c. Authentification

L'authentification consiste à assurer l'identité d'un utilisateur c.à.d. de garantir à chacun de correspondant que son partenaire est bien celui qu'il croit être.
On distingue deux types d'authentification :

- ***Le contrôle d'accès :*** C'est l'opération permettant d'être certain de l'identité d'une personne utilisateur pour permettre l'accès à des ressources uniquement pour la personne autorisée (par exemple l'utilisation d'un mot de passe pour un disque dur).
- ***Authentification de l'origine des données :*** Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services, l'authentification et l'intégrité.

d. La non répudiation (traçabilité)

La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction ; l'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information.

I.2.3. Problèmes de la cryptographie

Quelque soit le cryptosystème utilisé pour le chiffrement, ceci reste toujours cassable un jour ou l'autre. La sécurité d'un cryptosystème repose en fait sur la complexité des algorithmes définis et sur les puissances de calcul disponibles pour une attaque. Ainsi la solution adoptée actuellement est de faire « *retarder le travail des cryptanalystes* », c.à.d. faire en sorte que la durée nécessaire pour déchiffrer un code soit supérieure à la durée de validité des données.

I.3. Algorithmes cryptographiques

Au cours d'un échange visant à communiquer de façon secrète deux protagonistes, appelé ici l'émetteur et le récepteur à travers un canal peu sûr, l'information à transmettre sera donc chiffrée par un procédé de chiffrement en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de en utilisant une clé de déchiffrement sans laquelle son procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique : *algorithme cryptographique* (ou *chiffre***)**. La sécurité des données chiffrées est entièrement dépendante de deux facteurs : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un *cryptosystème***.**

I.3.1. Le principe de Kerckhoffs

Pour briser un cryptosystème, un opposant cherche à obtenir deux éléments d'information :

1. Quel est le type de système de codage utilisé ? et,
2. Quelle est la clé d'encodage utilisée ?

Bien entendu, son travail est simplifié (mais certainement pas terminé) s'il connait le type de système utilisé. Avec le temps cette information finit par circuler. Cette hypothèse de travail est appelée le principe de Kerckhoffs. Ce principe consiste à affirmer que la sécurité d'un système de chiffrement ne devrait pas être fondée sur le secret de la procédure utilisée, mais essentiellement sur le secret de la clé.

Auguste Kerckhoffs écrit en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire », où il disait [Kerc, 1883] :

« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants:

1) le système doit être matériellement, sinon mathématiquement, indéchiffrable.
2) il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.
3) la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
4) il faut qu'il soit applicable à la correspondance télégraphique.
5) il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
6) enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer ».

Les points 2 et 3 sont les axiomes fondamentaux de la cryptographie suivant lesquels l'attaquant possède tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé spécifique pour le chiffrement. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ce secret sera découvert ou volé.

I.3.2. Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplait (P, C, K, E, D) satisfaisant les points suivants :

1) P est un ensemble fini de blocs de textes clairs possibles.
2) C est un ensemble fini de blocs de textes chiffrés possibles.
3) K est un ensemble fini de clefs possibles.
4) Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k (e_k (x)) = x$ pour tout texte clair $x \in P$.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$ qui doit être transmise préalablement loin des yeux d'Oscar (à travers un canal de communication sûr). Supposant qu'Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne : $x = x_1 x_2 \ldots x_n$ avec : $n \in Z$, $n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera : $y = y_1 y_2 \ldots y_n$.

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1 x_2 \ldots x_n$. Le procédé de communication est illustré sur la Figure I.2.

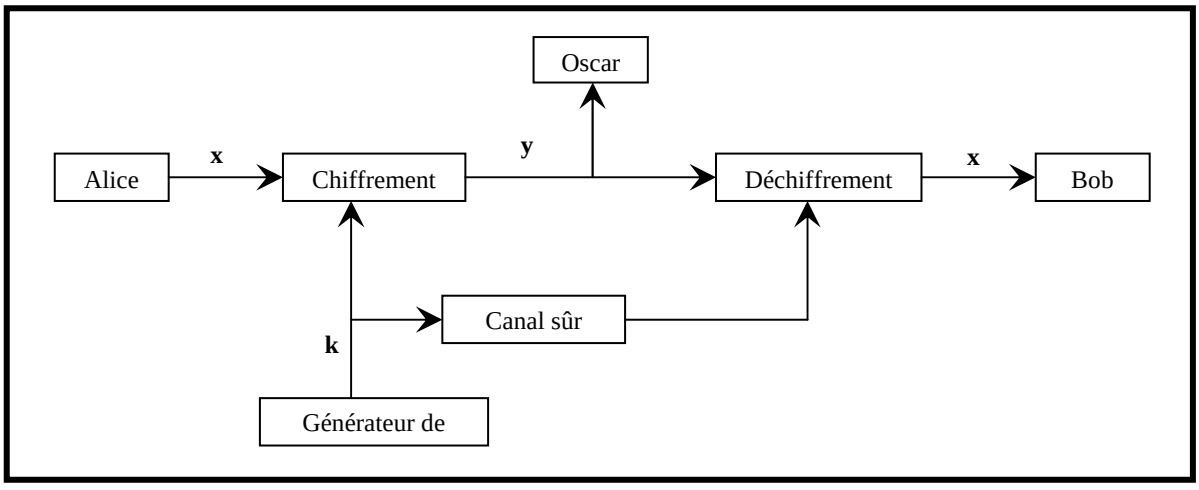


Figure I.2. *Le procédé de communication.*

I.3.3. Classes de cryptographie

Le schéma suivant illustre les différentes classes de la cryptographie :

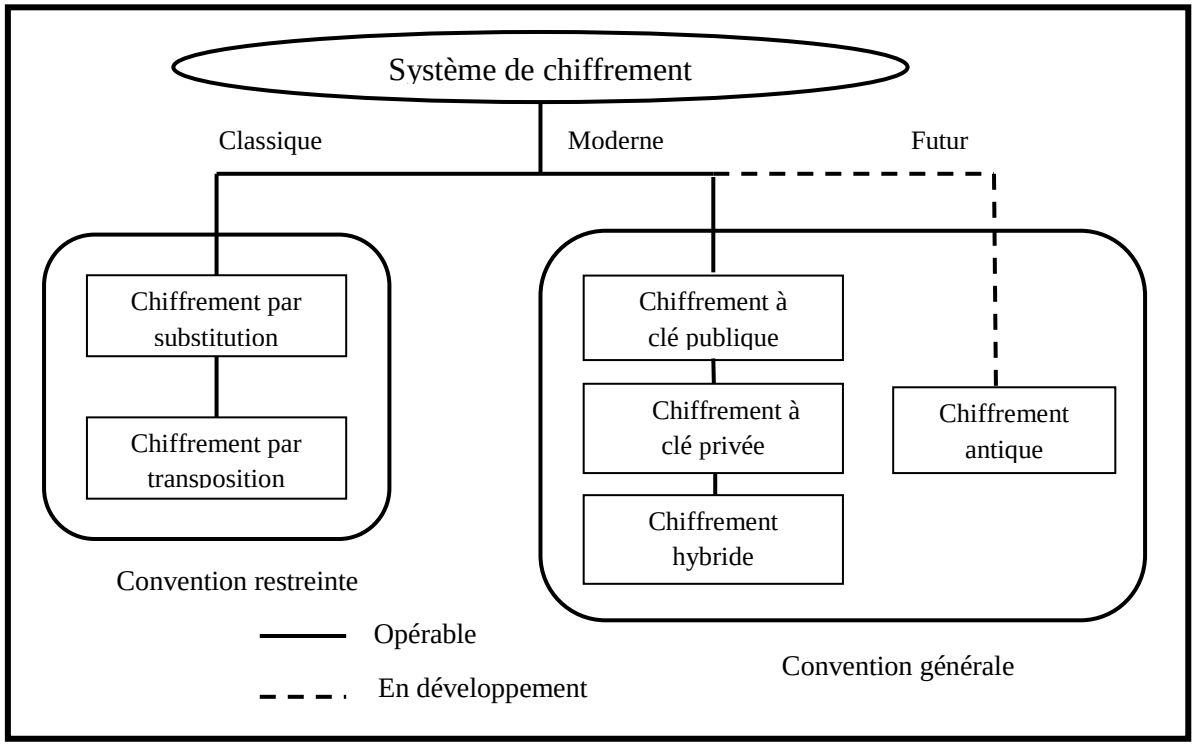


Figure I.3. *Les classes de la cryptographie*

I.3.3.1. La cryptographie classique

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble et ils sont trop simples pour offrir la moindre sécurité. Pour cacher la substance d'un texte, ils utilisent la substitution de caractères par d'autres ou les transposer dans des ordres différents. De ce fait, la confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide. On appelle généralement cette classe de méthodes : le chiffrement **à usage restreint**.

a. Cryptographie par substitution

La substitution signifie que chaque lettre (ou groupe de lettres) est substituée par une (ou groupe) lettre(s), chiffre(s) ou symbole(s). Le déchiffrement consiste à effectuer la substitution inverse. Selon la façon de substituer, on quatre catégories :

- **Substitution simple (mono-alphabétique)**

Le codage par substitution mono-alphabétique (ou encore les alphabets désordonnés) est le plus simple à imaginer. Chaque lettre dans le message clair est remplacée dans le message chiffré par une autre lettre différente unique pour toutes les occurrences de celle-ci.

Dans la littérature, plusieurs algorithmes ont été proposés, entre autres, nous citons : le chiffre de César, le chiffre Atbash, le carré de Polybe, etc.

- **Substitution poly-alphabétique**

Au lieu de remplacer une lettre par une même autre lettre dans tout le message comme dans la substitution simple, elle est remplacée périodiquement par différentes lettres. L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles.

- **Substitutions homophoniques**

Au lieu d'associer un seul caractère crypté à un caractère en clair, on dispose d'un ensemble de possibilités de substitution de caractères dans lequel on choisit aléatoirement. Par exemple : C=>S, K ; G=>G, J ; Q=>K ; S=>S, Z ; PH=>F ; …etc.

- **Substitution par polygrammes**

Au lieu de substituer des caractères, on substitue par exemple des digrammes : groupe de deux caractères. Pour se faire, deux moyen sont utilisés : soit par table (Chiffre de Playfair) ou par transformation mathématique (Chiffre de Hill).

b. Cryptographie par transposition

Elle consiste à permuter les lettres du message à chiffrer entre elles, afin de le rendre inintelligible. Plusieurs variations de transposition sont utilisées, parmi eux on trouve :

- **Transposition simple (à base matricielle)**

Elle consiste à écrire le texte en clair dans une matrice de ***n*** colonnes (une lettre dans chaque case), et ensuite de construire le texte chiffré en prenant les lettres à partir de cette matrice colonne par colonne. La clé dans ce cas est le nombre ***n***.

- **Transposition avec substitution simple**

L'idée dans ce cas est de combiner la transposition avec une substitution simple. Il s'agit ainsi de chiffrer le message clair par une méthode de substitution simple, et en suite d'en appliquer une transposition. Une autre astuce est souvent utilisée qui consiste à appliquer une fonction de permutation sur l'ordre d'arrangement des colonnes. On cite à titre d'exemple : le chiffre de DELASTELLE.

c. Conclusion

Les nouvelles techniques de communications (moyens de transports rapides, journaux, télégraphe, télégraphie sans fil) donne une nouvelle impulsion à la cryptologie. L'interception devient simple et le décryptement des informations devient vital. La cryptologie entre dans son ère moderne et les algorithmes classiques ne sont plus efficaces.

I.3.3.2. La cryptographie moderne

La cryptographie entre dans son ère moderne avec l'utilisation intensive des ordinateurs à partir des années septante. Vue la nécessité croissante de sécuriser les données dans tous les domaines (économique, industriel, informatique,...etc.) …. La cryptographie est appelée à devenir une technique de plus en plus fondamentale pour la protection des informations possédées et/ou échangées par des individus ou des organisations. Dans la cryptographie moderne, on utilise aussi des problèmes mathématiques que l'on ne sait pas (encore) résoudre, par exemple factoriser des grands nombres (chiffre RSA).

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ la *cryptographie à clef secrète,* ou encore appelée *symétrique,*
- ✓ la *cryptographie à clef publique,* dite également *asymétrique*.

a. La cryptographie symétrique

- **Principe**

Le cryptage à clé privée ou symétrique (ou encore dit conventionnel) utilise la même clé pour chiffrer et déchiffrer un message. Autrement dit, les clefs de chiffrement et de déchiffrement sont identiques ou on peut facilement calculer l'une à partir de l'autre. La connaissance de cette clef est cruciale pour la confidentialité des informations échangées. L'emploi d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Ces systèmes sont vulnérables parce qu'il est généralement possible de découvrir la clé à partir des messages codés. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1}. En excepte le système à masque jetable qui est hors porté de cette attaque.

Les algorithmes symétriques sont de deux types :

- ✓ Les algorithmes de *chiffrement en continu*, qui agissent sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé *chiffrement en flux* (Stream cipher).
- ✓ Les algorithmes de *chiffrement par blocs* (Bloc cipher), qui opèrent sur le texte en clair par groupes de bits appelés blocs.

Plusieurs algorithmes de chiffrement symétriques ont été définis. Nous présentons, ci-dessous, les plus connues de ces méthodes.

- **Quelques algorithmes de chiffrement symétrique**

A ce niveau, on va présenter, plus ou moins en détails, les plus fameux des algorithmes de chiffrement s'inscrivant sous ce mode.

1) Masque jetable (one-time pad)

Ce chiffre appelé aussi chiffre de Vernam ou encore le chiffrement parfait; est un algorithme de cryptographie inventé par Gilbert Vernam en 1917. Ce chiffrement est le seul qui soit théoriquement impossible à casser puisque la sécurité de ce système repose sur la génération complètement aléatoire de la clé [www2]. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront équiprobables. Malgré cela, il présente d'importantes difficultés de mise en œuvre pratique, il ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

Il consiste à combiner le message en clair avec une clé présentant les caractéristiques suivantes :

- choisir une clé K_M aussi longue que le texte à chiffrer.
- utiliser une clé constituée de caractères choisis aléatoirement.
- ne jamais utiliser 2 fois la même clé (d'où le nom de masque jetable).
- pour chiffrer un message faire le ou exclusif du message et de la clé : $C=M \oplus K_M$.
- pour déchiffrer un message l'opération est la même : $M= C \oplus K_M = M \oplus K_M \oplus K_M$.

2) DES (Data Encryption Standard)

Jusqu'aux années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence de besoins civils, le NBS (National Bureau of Standards) lança le 15 mai 1973 un appel d'offres dans le Federal Register (l'équivalent du Journal Officiel américain) [Stin, 1996] pour la création d'un système cryptographique qui doit :

- Reposer sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement.
- Etre facile à implémenter, logiciellment et matériellement et être très rapide aussi.
- Avoir un haut niveau de sécurité, lié uniquement à la clé et non à sa confidentialité.

Ce cryptosystème permet de chiffrer des messages de 64 bits avec une clef de 56 bits. De ce fait, c'est un système de chiffrement par blocs. Pour chiffrer un texte, il faut d'abord le découper en blocs de 64 bits puis appliquer le chiffrement sur chacun des blocs. Ainsi, les données en entrée de cet algorithme (texte clair et les données en sortie (texte chiffré) seront des bocs de 64 bits.
Sa clé est une chaîne binaire de 64 bits, mais en fait seuls 56 bits servent réellement à définir la clé. Les bits 8, 16, 24, 32, 40, 48, 56, 64 sont des bits de parité. Le $8^{ème}$ bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Ceci permet d'éviter les erreurs de transmission. Il y a donc pour DES 2^{56} clés possibles, soit environ 72 milliards possibilités. Malgré ça, c'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, alors qu'elle a été largement suffisante au moment de sa conception.

Nous donnons ci-dessous une idée simple et intuitive du fonctionnement du DES qui est un algorithme relativement simple puisqu'il combine des permutations et des substitutions. Il se déroule en quatre étapes [www3] :

1. **Préparation-Diversification de la clé :** le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K 16 sous-clésK_1,…, K_{16} à 48 bits.
   La figure ci-dessous montre comment obtenir à partir d'une clé de 64 bits 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES.

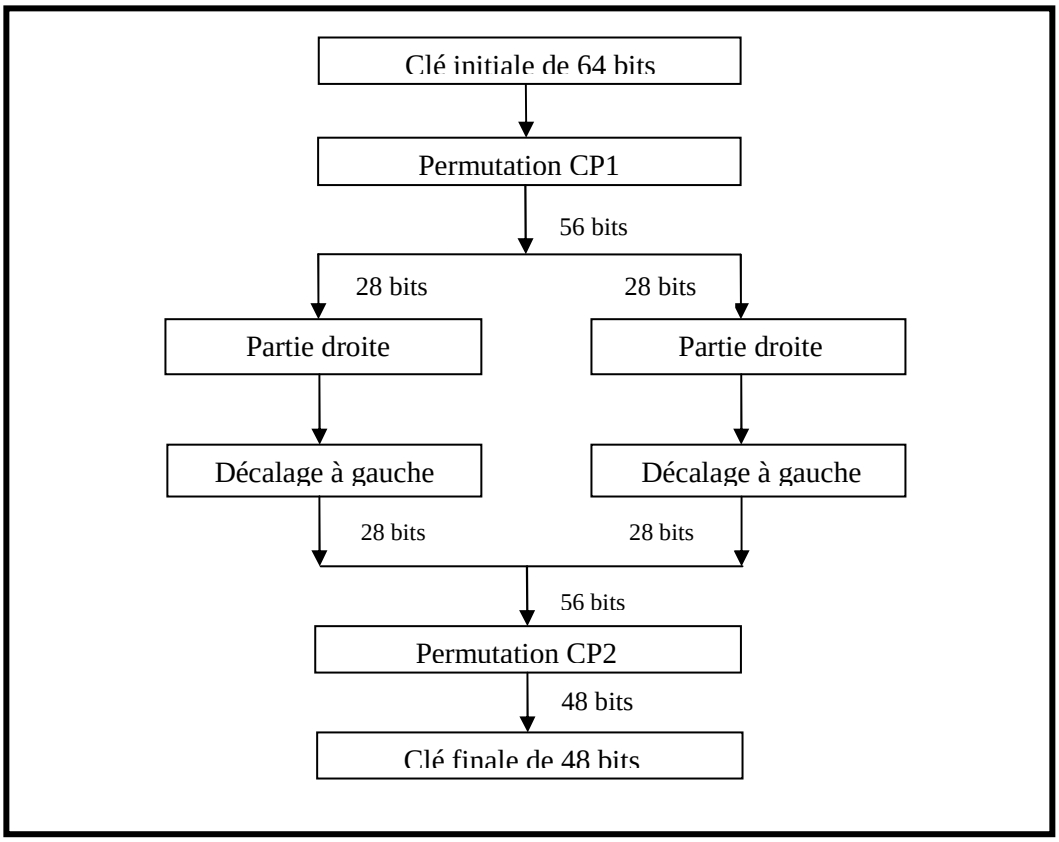


Figure I.4. *Génération des clés.*

La première étape consiste à faire une permutation noté CP1 dont la matrice est présentée ci-dessous, puis éliminer les bits de parité afin d'obtenir une clé d'une longueur de 56 bits. Elle est composée de deux matrice G_i et D_i chacune de 28 bits. Ces deux blocs subissent ensuite une rotation à gauche. Et enfin, ces derniers sont regroupé en un bloc de 56 bits ensuite une permutation CP2 a eu place pour fournir en sortie une clé de 48 bits. Des itérations de l'algorithme permettent de donner les 16 clés K_1, …, K_{16}.

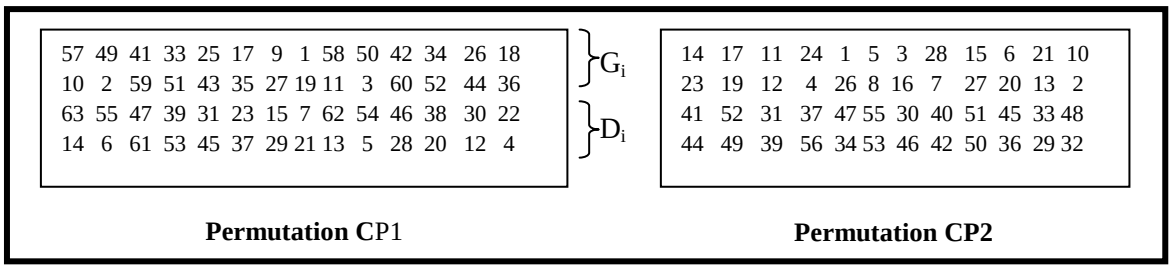


Figure I.5. *Permutation CP1 et CP2.*

2. **Permutation initiale :** pour chaque bloc de 64 bits x du texte, on calcule une permutation finie y=PI(x). y est représenté sous la forme : y = G_0 D_0 , G_{16} étant les 32 bits à gauche de y, D_0 les 32 bits à droite (voir Figure I.6). Quant à leurs significations, par exemple, la permutation initiale (IP) signifie que le $58^{ème}$ bit de la chaine à chiffrer, *x*, est le premier bit de IP(*x*),…

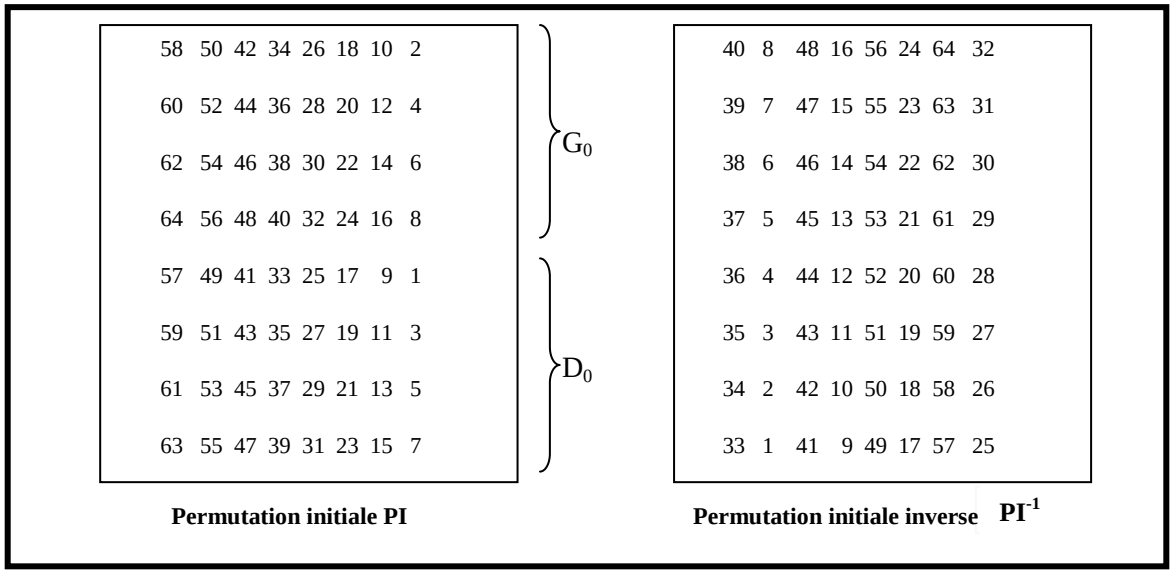


Figure I.6. *La permutation initiale et son inverse.*

3. **Itérations (Rondes) :** On applique 16 rondes d'une même fonction. A partir de $G_{i-1}D_{i-1}$ (pour i de 1 à 16), on calcule G_iD_i en posant :
   - $G_i = D_{i-1}$
   - $D_i = G_{i-1} \oplus f(D_{i-1},K_i)$

D'après la formule qui correspond au calcul de D_i, on constate que la fonction *f* utilise deux arguments ayant des tailles différentes : D_{i-1} de 32 bits et k_i de 48 bits. Ainsi, D_{i-1} sera étendu en 48 bits grâce à une matrice appelée table d'expansion (notée E), dont les 48 bits sont mélangés et 16 d'entre eux sont dupliqués (voire Figure I.7).

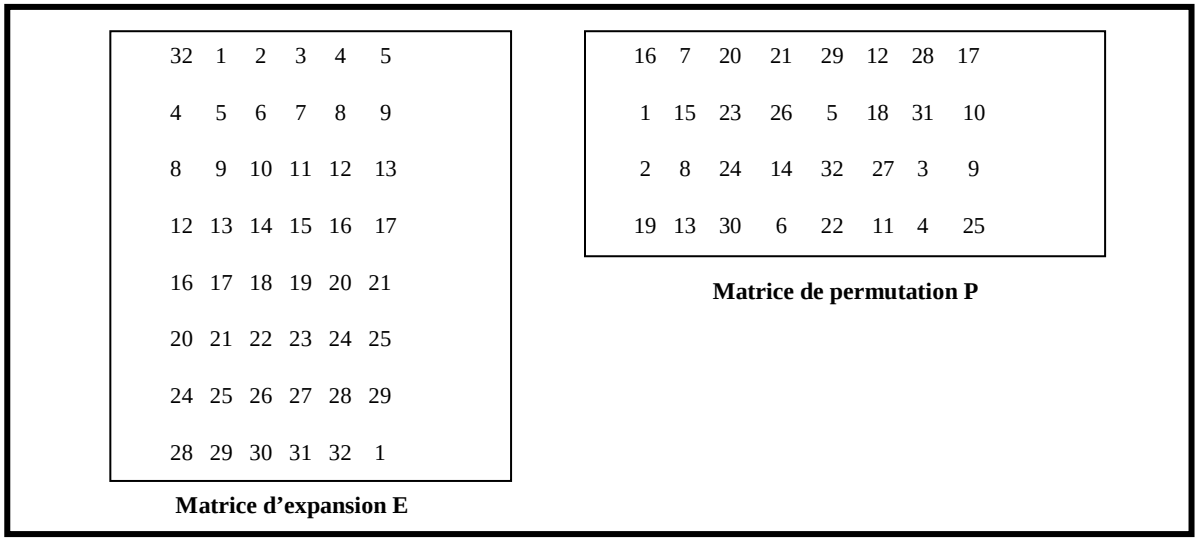


Figure I.7. *Matrice d'expansion E et de permutation P.*

L'opération qui suit consiste à calculer un « ou exclusif » entre le résultat obtenu jusqu'à maintenant, codé sur 48 bits, et G_{i-1} codé sur 32 bits. Donc, le résultat final de *f* doit être codé sur 32 bits au lieu de 48 bits. Pour cela, la chaine de $48 = 8 \times 6$ bits sera transformée en une chaîne de $32 = 8 \times 4$ bits en utilisant des dispositifs appelés *boites-S*. Elles sont au

nombre de huit, où chacune calcule un bloc de 4 bits à partir d'un bloc de 6 bits. Enfin, on applique une permutation à ce 32 bits pour obtenir la valeur finale de *f* (voir la Figure I.7).
L'ensemble de ces résultats en sortie de P est soumis à un « ou exclusif » avec le G_0 de départ (comme il est indiqué dans la Figure I.9) pour donner D_1, tandis que le D_0 initial donne G_1, et ainsi de suite pour les 16 itérations comme il est mentionné auparavant.
La succession d'opérations constituant la fonction ***f*** est représentée à travers la Figure I.8**.**

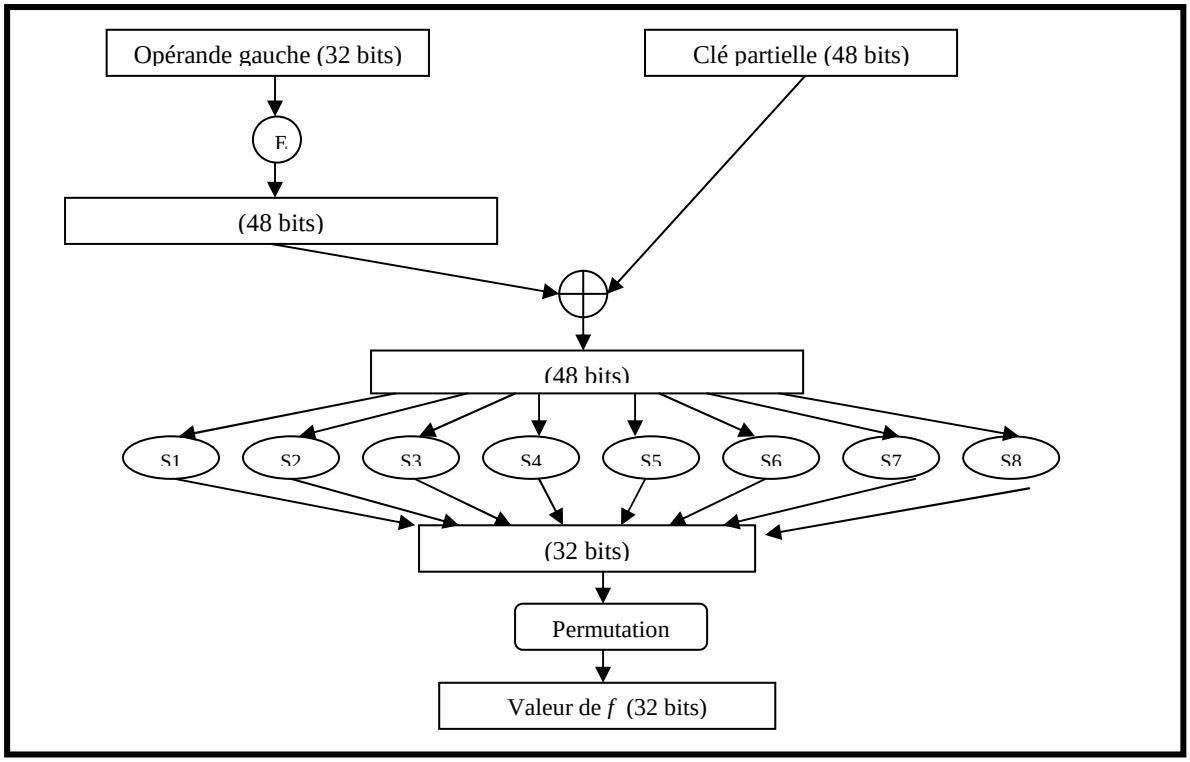


Figure I.8. *Schéma de la fonction f.*

4. **Permutation initiale inverse :** A la fin des itérations, les deux blocs G_{16} et D_{16} sont récoltés, puis soumis à la permutation initiale inverse : $Z = PI^{-1}(G_{16}\ D_{16})$. Le résultat en sortie est un texte codé de 64 bits.
   Cette description peut être résumée par le schéma général illustré par Figure I.9, où on a seulement représenté quelques-unes des 16 étapes.

Remarque : le déchiffrement suit le même algorithme avec la même clef *K*. Seules les sous-clés sont appliquées dans le sens inverse en commençant par la clé k_{16} jusqu'à la clé k_1.

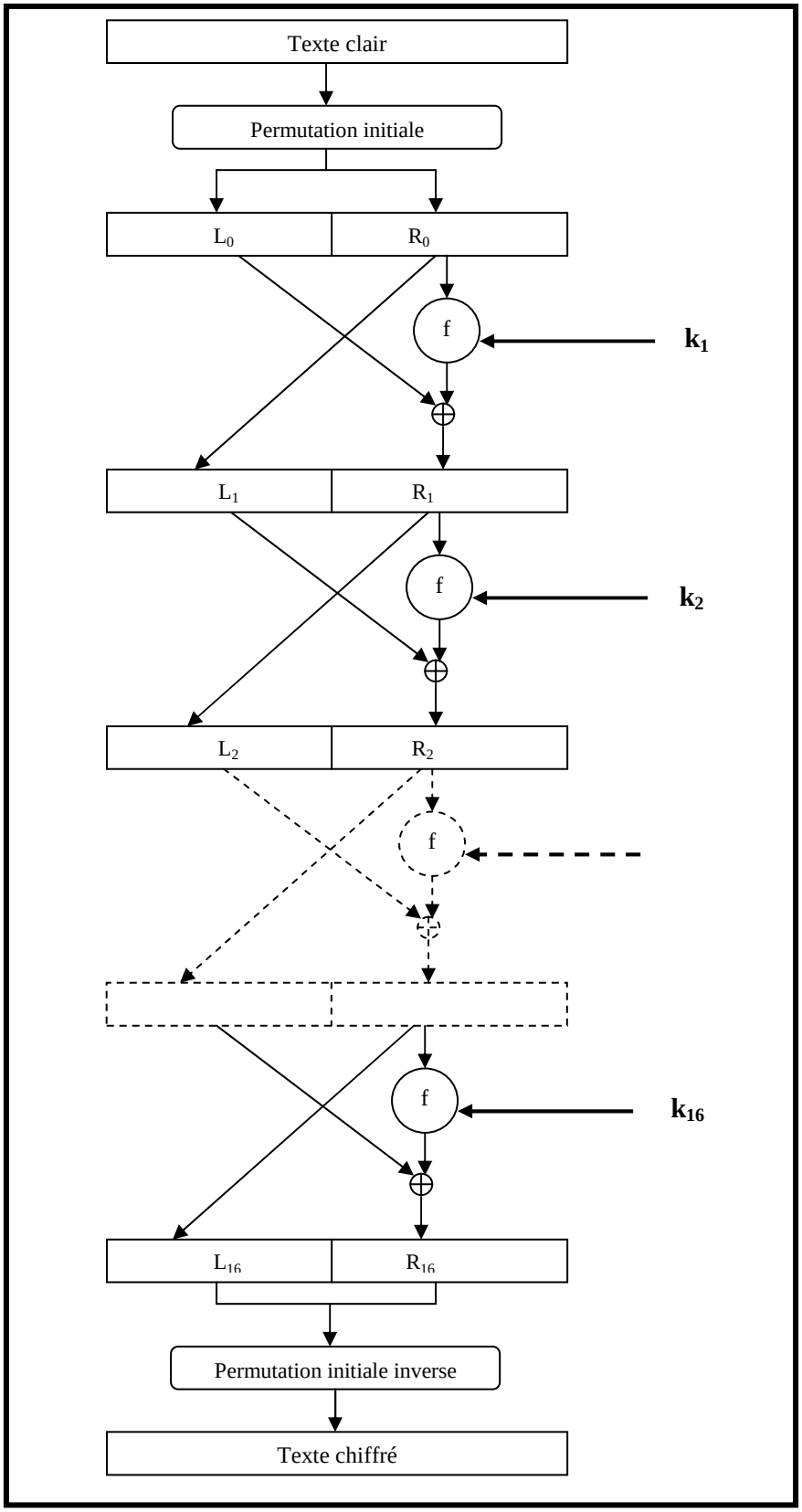


Figure I.9. *Schéma général de DES.*

Le DES a fait l'objet de très nombreuses attaques. On peut citer quelques une :

- **Cryptanalyse différentielle :** c'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires. Ainsi, un DES à 8 ou à 10 tours peut facilement être cassé, mais le DES complet à 16 tours est resté hors de portée de cette attaque.

- **Cryptanalyse linéaire :** c'est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Elle n'est utilisable que pour un DES restreint à quelques tours, mais le DES réel n'est pas menacé par cette attaque.

- **Recherche exhaustive :** les laboratoires RSA ont lancé en Janvier 1997 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES pour démontrer que la taille des clés DES a devenu insuffisante. Ils ont réussi le 17 Juin 1997. Bien qu'une telle recherche demande des moyens considérables, elle a montré que le DES n'offre plus aujourd'hui une grande sécurité.

3) Triple DES (3DES)

Pour palier l'insuffisance cryptographique observée du cryptosystème DES, dû à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. Triple DES est apparait comme une solution pour remédier les faiblesses de DES. Comme son nom l'indique, le principe du triple DES est d'effectuer 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Ce principe peut être formulé comme suit :

$$\textbf{Triple-DES}_{k1,\,k2} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}^{-1}_{k2} \;\textbf{O}\; \textbf{DES}_{k1}$$

On peut constater que le DES sera retrouvé comme cas particulier de la formule ci-dessus, lorsque $k_1 = k_2$. Le déchiffrement de son tour est formulé par :

$$\textbf{Triple-DES}^{-1}_{k1,\,k2} = \textbf{DES}^{-1}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}^{-1}_{k1}$$

Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc composée de 112 bits. Une autre variante à trois clés DES différentes peut être conçue. D'une façon plus formelle, son principe peut être donné par la formule suivante :

$$\textbf{Triple-DES}_{k1,\,k2,\,k3} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}_{k3}$$

Malgré cela, cette variante reste aussi fragile à une attaque de coût en 2^{112} s'appuyant sur l'un des deux messages intermédiaires.

4) AES (**Advanced Encryption Standard**)

Avec le temps et les progrès de l'informatique, les 2^{56} clés possibles du DES se voient incapables de fournir une barrière infranchissable dû à la faiblesse des clés de 56 bits. Désormais avec des moyens modestes, percer les messages chiffrés par DES en un temps raisonnable, ne pose aucun problème. De ce fait, l'AES a vu le jour. Il est issu d'un appel à candidatures international lancé en janvier 1997 et ayant reçu 15 propositions. Au bout de cette évaluation, ce fut le candidat Rijndael du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen [www4] qui a été choisi par le NIST en Octobre 2000 pour être l'algorithme AES et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. II a été déclaré vainqueur de la deuxième ronde dans laquelle

s'opposaient les 5 candidats finalistes (MARS, RC6, Rijndael, Serpent, TwoFish). L'AES a une taille longue de sa clé allant jusqu'à 256 bits qu'il le permet de résister toujours a la cryptanalyse. L'AES est devenu le nouveau standard du chiffrement symétrique comme le signifie cet acronyme. L'AES est libre d'utilisation comme l'est l'algorithme DES.

Techniquement, le chiffrement AES travaille avec des blocs de 128 bits seulement et la longueur des clés utilisées peut être de 128, 192 ou de 256 bits. Chaque bloc subit une séquence de transformations que nous résumons à travers les points suivants :

1. Addition de la clé secrète et du bloc en question avec un « ou exclusif ».
2. ByteSub : les 128 bits sont répartis en 16 blocs de 8 bits (16 octets), qui sont ensuite placés dans une matrice de 4×4. Chaque octet est transformé par une fonction non linéaire S (S-box) conçu pour résister à la cryptanalyse linéaire et différentielle.
3. ShiftRow: les lignes de cette matrice sont soumises à une rotation vers la droite où l'incrément pour la rotation varie selon le numéro de la ligne. La $2^{ème}$ ligne est décalée d'une colonne, la $3^{ème}$ ligne de 2 colonnes, et la $4^{ème}$ ligne de 3 colonnes.
4. MixColumn: chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne. Cela revient à multiplier la matrice 4×4 par une autre matrice 4×4.
5. AddRoundKey: une clé dite *de tour* est générée à partir de la clé secrète par un sous-algorithme dit *de cadencement*. Cette clé de tour est ajoutée par un « ou exclusif » au dernier bloc obtenu.

Ces différentes opérations, définissant un *tour*, sont répétées plusieurs fois. Cependant dans le dernier tour il n'y a pas d'opération MixColumn. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours. La figure suivante résume le principe de fonctionnement de cet algorithme de chiffrement.

Le déchiffrement consiste en appliquer les opérations inverses dans chacune des étapes (*InvSubBytes*, *InvShiftRows*, *InvMixColumns*). *AddRoundKey* (à cause du XOR) est son propre inverse. On réitère ce processus le nombre de tour-1. Pour le dernier tour on exclu l'opération *InvMixColumns* ; et comme dans le chiffrement pas d'opération *InvMixColumns* dans le dernier tour.

De nombreux travaux de cryptanalyse ont été publiés mais pour l'instant il n'a pas été cassé et la recherche exhaustive demeure la seule solution. En particulier, on cite :

- **Attaques sur des versions simplifiées :** *Niels Ferguson* et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours. En effet, cette dernière attaque repose sur le principe des clés apparentées. Dans une telle attaque, la clé demeure secrète mais l'attaquant peut spécifier des transformations sur la clé et chiffrer des textes à sa guise. Il peut donc légèrement modifier la clé et regarder comment la sortie de l'AES se comporte.

- **Attaques sur la version complète :** plusieurs chercheurs ont mis en évidence des possibilités d'attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL. Le XSL fait appel à une analyse heuristique dont la réussite n'est pas systématique. Elles sont impraticables car le XSL demande au moins 2^{87} opérations voire 2^{100} dans certains cas. Le principe est d'établir les équations (quadratiques / booléennes) qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer et l'AES est toujours considéré comme sûr.

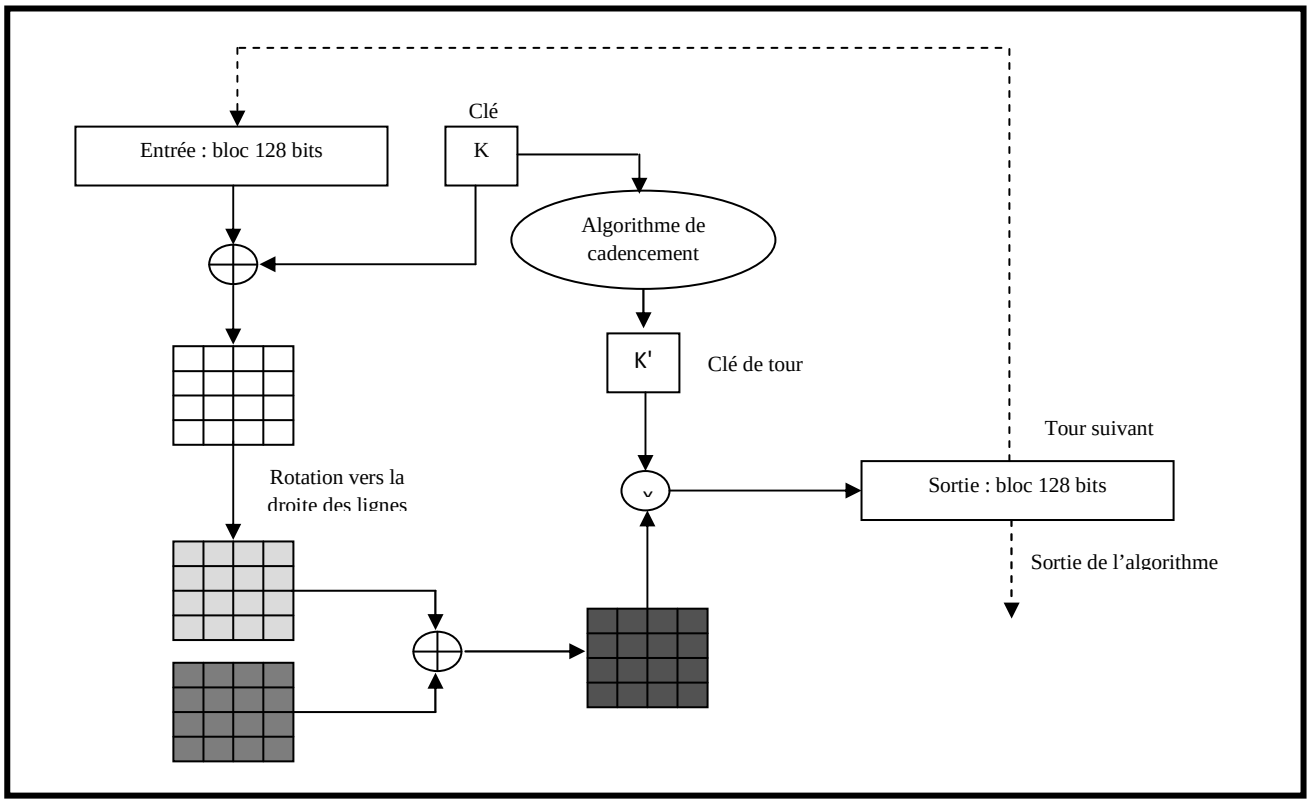


Figure I.10. *Schéma général de l'AES.*

5) IDEA (International Data Encryption Algorithm)

L'**IDEA** qui est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 rondes inventé en 1990 à Zürich par James L. Massey et Xuejia Lai. C'est la taille des clés qui apporte une grande sécurité au système (que l'on n'a pas, à l'heure actuelle, officiellement réussi à "casser").

Chaque bloc est divisé en 4 sous-blocs de 16 bits : X_1, X_2, X_3 et X_4. Ces quatre sous-blocs deviennent les entrées de la première ronde de l'algorithme.
À chaque ronde, la séquence d'évènements est la suivante (voir Figure I.11) :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé ;
5. combinez par OU exclusif les résultats des étapes (1) et (3) ;
6. combinez par OU exclusif les résultats des étapes (2) et (4) ;
7. multipliez le résultat de l'étape (5) avec la cinquième sous-clé ;
8. additionnez les résultats des étapes (6) et (7) ;
9. multipliez le résultat de l'étape (8) par la sixième sous-clé ;
10. additionnez les résultats des étapes (7) et (9) ;
11. combinez par OU exclusif les résultats des étapes (1) et (9) ;
12. combinez par OU exclusif les résultats des étapes (3) et (9) ;
13. combinez par OU exclusif les résultats des étapes (2) et (10) ;
14. combinez par OU exclusif les résultats des étapes (4) et (10).

La sortie de la ronde est constituée des 4 sous-blocs produits par les étapes (11), (13), (12) et (14). Changez les deux blocs intérieurs (sauf lors de la dernière ronde) et cela donne l'entrée de la ronde suivante.
Après la huitième ronde, il y a une transformation finale :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé.

Enfin les 4 sous-blocs sont réassemblés pour former le texte chiffré.

Chaque ronde utilise 6 sous-clés K_i^r de 16 bits : K_1^1, ..., K_6^1, …, K_1^8, …, K_6^8. La transformation finale utilise 4 sous-clés : $K_1^9, K_2^9, K_3^9, K_4^9$; donc un total de 52 sous-clés est utilisé, les premiers 8 sous-clés sont extraites directement à partir de clé, des groupes de 8 clés sont extraits par rotation à gauche de 25 bits de la clé K, ce qui donne 6 rotations en total.
Concernant le processus de déchiffrement on utilise celui de chiffrement avec un seul changement dans la génération des clés. En fait, on utilise K pour générer les sous clés K_i^r ; à partir de ces derniers, d'autres clés $K_i^{'r}$ sont obtenues dans le Tableau I.1 [Omar, 2006] ; ensuite on utilise $K_i^{'r}$ à la place des K_i^r dans l'algorithme de chiffrement IDEA. Dans le Tableau I.1, $-K_i$ dénote l'opposé modulo 2^{16} de K_i. K_i^{-1} dénote l'inverse multiplicative de $K_i \bmod(2^{16}+1)$, se trouvant aussi dans {0, 1, …,216-1}.

Ronde r	$K_1^{'(r)}$	$K_2^{'(r)}$	$K_3^{'(r)}$	$K_4^{'(r)}$	$K_5^{'(r)}$	$K_6^{'(r)}$
r =1	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
2 ≤ r ≤ 8	$(K_1^{10-r})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
r= 9	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	--	--

Tableau I.1. *Les sous clés de déchiffrement $K_i^{'r}$ générées à partir des sous clés K_i^r.*

La méthode de génération des sous-clés de l'IDEA est toujours régulière et donc pourrait être une faiblesse à l'algorithme. Cependant, il est considéré comme étant hautement sécuritaire. En effet, pour résoudre IDEA avec l'attaque en force brute, il faudrait effectuer 2^{128}, donc 10^{38} opérations [www5].

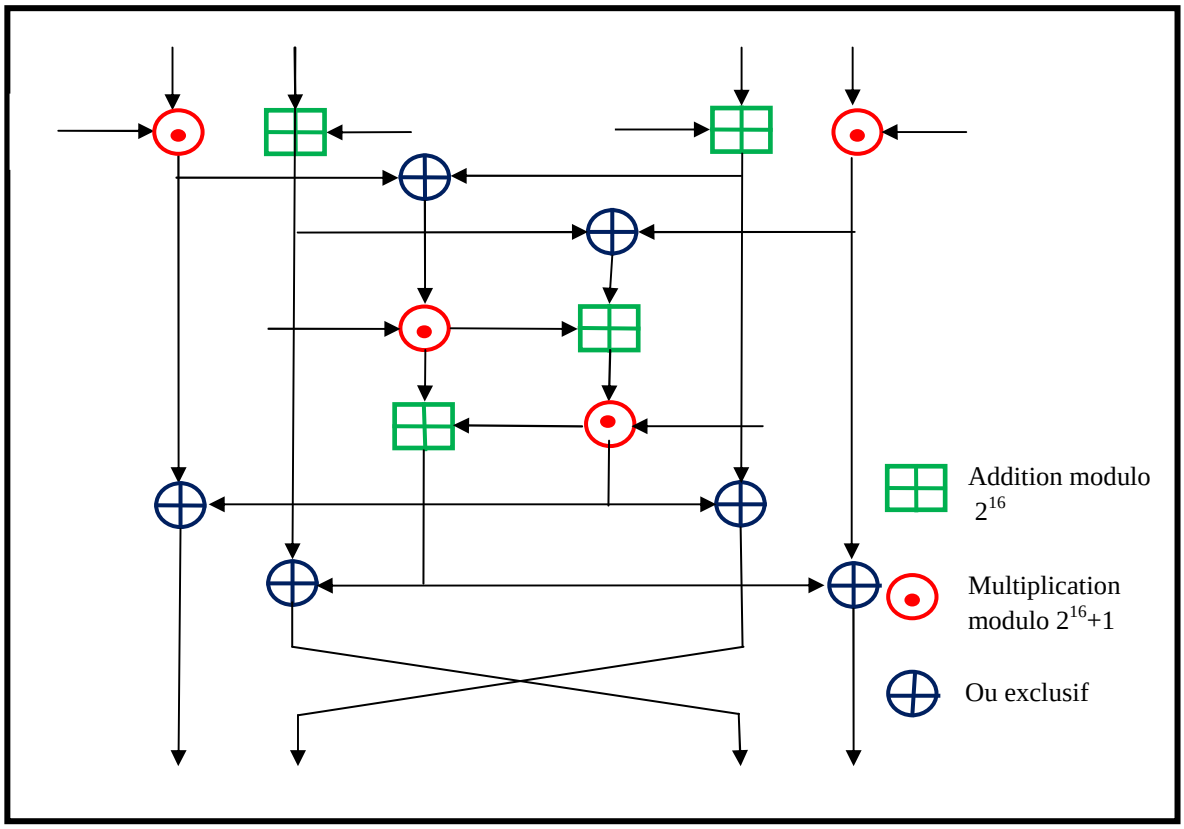


Figure I.11. *Schéma général de l' IDEA.*

- **Conclusion**

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. AES est immunisé contre les attaques et il reste néanmoins le meilleur choix dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause. D'autre part, IDEA est considéré par les spécialistes comme l'un des meilleurs cryptosystèmes à clé privée. Il est utilisé par le PGP pour le chiffrement de données.

b. La cryptographie asymétrique

- **Principe**

La *cryptographie asymétrique* ou encore dite *à clé publique* utilise deux clés différentes pour chaque utilisateur : une est privée et n'est connue que de l'utilisateur et qui est sensé d'être en mesure de faire signature ou déchiffrement. L'autre est publique diffusée en général dans un annuaire et donc accessible par quiconque afin de permettre aux interlocuteurs de mettre en œuvre les opérations réciproques (vérification de signature ou chiffrement de message). Les clés, publique et privée sont mathématiquement liées par l'algorithme de cryptage de telle manière qu'un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de *fonction à sens unique avec trappe*. Sachons qu'une fonction est appelée à *sens unique* si elle est

aisément calculée, mais extrêmement difficile de déduire la fonction inverse. Et elle sera dite *à trappe*, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la *trappe*. L'utilité d'utilisation d'une telle fonction réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas assez facile puisqu'elles s'appuient généralement sur des problèmes mathématiques réputés difficiles.

Ce cryptage présente l'avantage de permettre le placement de signature numérique dans le message et ainsi permettre l'authentification de l'émetteur grâce à la fonction de hachage. Le principal avantage consiste à résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé puisque la clé privée n'est connue que par l'utilisateur. Bien que plus lent que la plupart des cryptages à clé privée il reste toujours préférable pour 3 raisons :

- ✓ plus évolutif pour les systèmes possédant des millions d'utilisateurs.
- ✓ authentification plus flexible.
- ✓ supporte les signatures numériques.

Les systèmes asymétriques les plus connus sont [Mene, 1996] : RSA basé sur la difficulté de la factorisation des grands entiers, El Gamal basé sur la difficulté de résoudre le problème du logarithme discret dans un corps fini et les systèmes sur les courbes elliptiques basés sur la difficulté de certains calculs sur les courbes elliptiques.

- **Quelques algorithmes asymétriques**

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivants :

1) RSA

La méthode de cryptographie RSA a été inventée en 1977. Elle titre son nom des noms de ses trois inventeurs: *R. Rivest, A. Shamir* et *L. Adleman* [www6]. Le RSA est le premier et est encore le système cryptographique à clé publique le plus utilisé de nos jours et toujours considéré comme sûr. Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers ; à l'heure actuelle, il est pratiquement impossible de les reconstituer en un temps raisonnable. Ainsi, la sécurité de RSA semble satisfaisante malgré qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser.

On peut résumer le fonctionnement de ce cryptosystème dans les étapes suivantes :

- **Fonction d'encodage E (publique) :** la clé publique k utilisée pour l'encodage comporte deux entiers: k = (e,n). L'opération de chiffrement se fait au moyen de l'élévation à la puissance **e** modulo n : $E_k(M) = M^e \bmod n$.
- **Fonction de décodage D (privée) :** la clé secrète k′ utilisée pour le déchiffrement est aussi un couple d'entiers : k′(d,n). Pour reconstituer le message initial, la fonction inverse d'encodage est appelée, elle est ainsi : $D_{k'}(M) = M^d \bmod n$.
- **Détermination des clés :**
  - **Détermination de n :** pour calculer n on doit initialement choisir deux entiers premiers *p* et *q* très grands et leurs valeurs sont secrètes connus que part l'utilisateur. Le choix de *p* et *q* affecte grandement le niveau de sécurité de RSA. Pour cela, il faut évidemment se prémunir contre les algorithmes de factorisation dont la complexité dépend essentiellement de la taille du plus petit facteur premier de *n* [Zimm, 2005], donc si possible choisir *p* et *q* de même taille.

- **Détermination de e :** pour calculer e, on calcule premièrement un entier z = (p-1)*(q-1) puis tout simplement choisir un entier e premier avec z.
- **Détermination de d :** pour calculer d, on procède ainsi : e*d≡1[z].

La sécurité de RSA est basée sur l'hypothèse que la fonction E est à sens unique, ce qui rend impossible à décrypter un texte chiffré. Mais à l'aide de la trappe qui est la factorisation n = p*q, il est possible d'en trouver d et donc la clé privée.

Depuis son apparition, plusieurs attaques ont été découvertes contre le RSA. Et même si aucune de ces attaques n'est réellement destructive, elles démontrent toutefois qu'il faut implémenter RSA avec beaucoup de précautions. La fameuse attaque est :

- **Recherche exhaustive :** comme la fonction de chiffrement RSA est déterministe, si l'ensemble des messages possibles est connu et de petite taille, il sera facile de décrypter par une recherche exhaustive [Ster, 2004]. Pour éviter cette attaque, il est indispensable de randomiser les messages avant chiffrement.

2) Chiffrement d'ElGamal

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste c.à.d. un même message M peut avoir plusieurs chiffres différents [Lafo, 2006]. Il est basé sur la difficulté du problème des logarithmes discrets c.à.d. la difficulté de trouver l'unique *a* noté $\log_\alpha ß$ / $0 \leq a \leq p-2$ tel que : $\alpha^a \equiv ß \pmod p$ où *p* est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$. Cependant, et pour éviter les attaques connues, *p* doit être convenablement choisi, et $p-1$ doit avoir un grand facteur premier.

D'une manière formelle, l'algorithme ElGamal peut être résumé comme suit :

Soit *p* un nombre premier tel que le problème du logarithme discret dans Z_p soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et $K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod p\}$
Les valeurs *p*, α et ß sont publiques, et *a* est secret.
Pour k = (*p*, α, *a*, ß), et pour un $k \in Z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$\mathbf{e_k(x, k) = (y_1, y_2)}$$

Où

$$\mathbf{y_1 = \alpha^k \bmod p}$$

Et

$$\mathbf{y_2 = x\, ß^k \bmod p}$$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit comme suit :

$$\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$$

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par $ß^k$, en produisant y_2 ;
- ✓ la valeur α^k est également transmise en tant que partie du texte chiffré ;
- ✓ Bob, qui connaît l'exposant secret *a*, peut calculer $ß^k$ à partir de α^k. Il peut alors « enlever le masque » en divisant y_2 par $ß^k$ et obtenir le texte clair x.

Une attaque possible à ce cryptosystème est celle dite ***man in the middle***. Son principe dans le cas d'ElGamal fonctionnant avec le mode *Diffie-hellman* est illustré sur la figure I.12.

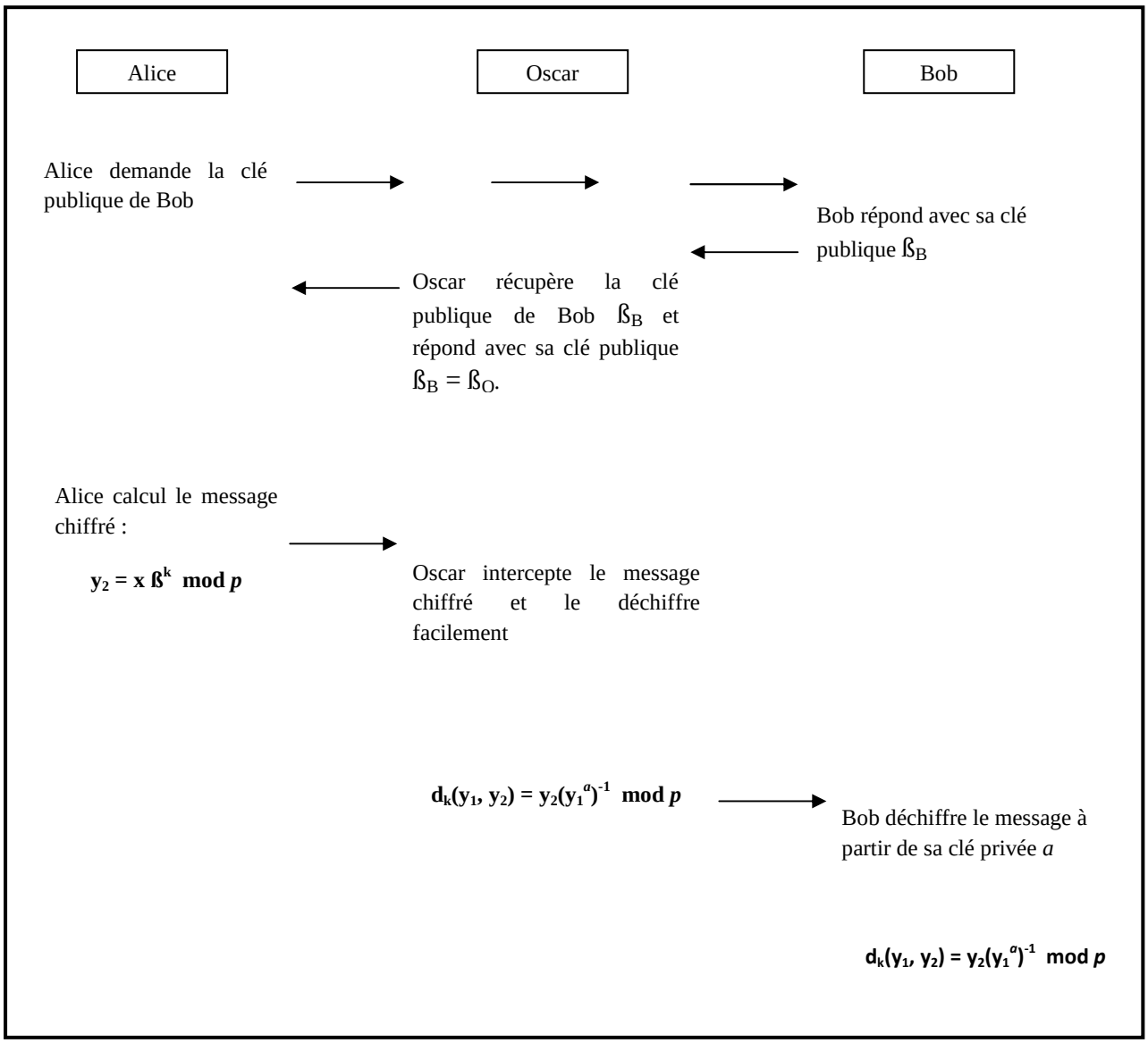


Figure I.12. *Principe de l'attaque « man in the middle ».*

- **Conclusion**

Les calculs faits en 1995 ont ouvert un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent des chiffres plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique serait capable de remettre en cause ce cryptosystème.

De même, casser l'algorithme ElGamal est aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Et pour assurer une bonne sécurisation de cet algorithme, Zimmermann en 2005 [Zimm, 2005] a recommandé l'utilisation des clés d'au moins 1024 bits.

c. Comparaison entre les cryptosystèmes symétriques et asymétriques

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage.

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ?
À partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que l'importance de la taille de clé n'est légitime que dans le cas de la clé secrète, puisque les seules attaques possibles sont les attaques exhaustives [www7]. Mais, dans le cas de la clé publique, la taille de la clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

Méthode	Avantages	Inconvénients
À clefs Secrètes	▪ Rapidité de calcul en général (dépend de la taille de la clé). ▪ Adaptée au cryptage de flux de données.	▪ Moins sécurisé (DES). ▪ Problème de communication de clefs entre émetteur et récepteur. ▪ Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clefs Publiques	▪ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre. ▪ Permet la signature électronique. ▪ Un couple de clés publique/privée suffisant pour 'n' correspondants.	▪ Lente. ▪ Problèmes de gestion de clefs publiques.

Tableau I.2. *Comparaison entre les méthodes de chiffrement symétriques et asymétriques.*

d. La cryptographie hybride

- **Principe**

La cryptographie asymétrique est beaucoup plus lente que la cryptographie symétrique qui brille par sa rapidité. En revanche, cette dernière souffre d'une grave lacune ; assurer une transmission secrète de la clé. Pour palier ce défaut et cumuler les avantages des deux méthodes on a fait recourt à la cryptographie hybride. On code tout d'abord les données avec une clé privée dite *clé de session*, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Comme la clé est courte, on utilise l'algorithme asymétrique puisqu'il prend peu de temps. En revanche, chiffrer l'ensemble du message avec un algorithme asymétrique serait plus lourd. Il suffit ensuite d'envoyer le message chiffré avec une clé privée et accompagné de cette dernière chiffrée avec une clé publique. Le destinataire procède inversement, il commence à déchiffrer la clé symétrique avec sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, via un déchiffrement symétrique pour retrouver le message original. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

- **Quelques algorithme hybride**

1) PGP (**Pretty Good Privacy**)

PGP est inventé par *Philip Zimmermann*, qui dise que tout individu a droit à la confidentialité, notamment les organisations des droits de l'homme dans des pays soumis à la dictature. Il l'a mis à disposition gratuitement sur Internet. Cela lui a valu de sérieux ennuis avec la justice américaine, car les logiciels de cryptage sont considérés comme du matériel de guerre et sont interdits à l'exportation.

Il est souvent utilisé pour garantir l'authentification et le contrôle d'intégrité des fichiers et du courrier électronique. Mais avant de crypter un texte avec PGP, les données doivent tout d'abord compressées dont le but de réduire le temps de transmission, ainsi d'économiser l'espace disque et, surtout, de renforcer la sécurité cryptographique puisque les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement alors que la compression réduit ces modèles dans le texte en clair.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes qui sont [www8] :

✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
✓ PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de déchiffrement se fait également en deux étapes, qui sont [www8] :

✓ PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
✓ PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

Depuis 1978, la recherche universitaire civile a intensément attaqué la cryptographie à clé publique, sans pour autant réussir à la remettre en cause. Mais cela ne fournisse aucune garantie totale sur la sécurité de cette manière de chiffrer, car une attaque menée par le gouvernement, par exemple, qui ne se manque pas de ressources très développées ou même en utilisant quelques nouvelles percées mathématiques classées top-secret, peut tenir à bout ces cryptosystèmes conventionnels utilisés dans PGP.

Tout de même, l'optimisme semble justifié. Les algorithmes de clé publique, les algorithmes de contraction de message, et les chiffres par blocs utilisés dans PGP ont été conçus par les meilleurs cryptographes du monde [Loid, 2005]. Les chiffres de PGP ont subi des analyses de sécurité approfondies et des examens méticuleux de la part des meilleurs cryptographes dans le monde non classé top secret. De plus, et même si les chiffres par blocs utilisés dans PGP possèdent quelques faiblesses, la compression du texte clair utilisée avant le chiffrement réduit de façon considérable ces faiblesses.

2) GPG (**GNU Privacy Guard**)

GPG est la version GNU de PGP. En raison qu'il est sous licence GPL (General Public License) il permet la fourniture du code-source, la libre modification et la libre redistribution de ce code-source. Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

- **Conclusion**

Aujourd'hui, de nombreux gouvernements ont restreint l'usage du PGP, qui a été largement diffusé par son développeur, en pensant qu'un cryptage trop fiable ferait le jeu des terroristes et des trafiquants. Toutefois, il y'a pas mal d'applications qui utilisent encore ce système de chiffrement, telles que les paiements en ligne qui se font grâce au procédé SSL

fonctionnant selon le principe du PGP. De sa part, GPG, qui est un cryptosystème utilisant des algorithmes de chiffrement à clé publiques (DSA, RSA et ElGamal), est largement utilisé dans les communications par messagerie, c.-à-d. pour chiffrer des mails, ainsi que pour signer des données.

I.3.3.3. Cryptographie quantique

a. Principe

La cryptographie à base d'algorithmes aura toujours des faiblesses. Même si la cryptanalyse bloque devant un algorithme, la force brute pourra toujours décrypter n'importe quel code si on lui donne assez de temps. Même avec le plus solide des chiffrages, le contenu du message peut être subtilisé et dupliqué. Les clés, quant à elles, peuvent être volées en chemin ou présenter des faiblesses qui les rendent prévisibles.

Si maintenant on base notre cryptographie, non pas sur un algorithme mathématique, mais sur des lois de la physique quantique, il n'y plus de force brute qui peut briser un code ici. Le cryptage ne se trouve pas dans des formules, mais dans des photons, ainsi tout le monde peut accéder à des formules, mais pas tout le monde peut accéder aux photons sans que le message devienne inutile. De ce fait, l'information n'est plus sécurisée par des subterfuges mathématiques, mais plutôt par des lois fondamentales de physique. Au de-là, la cryptographie quantique a vu le jour.

La figure I.13 [Nave, 2002] présente un exemple relatif à la possibilité soit qu'un photon traverse le filtre ou pas, selon l'orientation de sa polarisation. Sachons qu'un filtre permet de distinguer entre les photons polarisés horizontalement (0°) et verticalement (90°) ; un autre entre les photons polarisés en diagonale (45°, 135°).

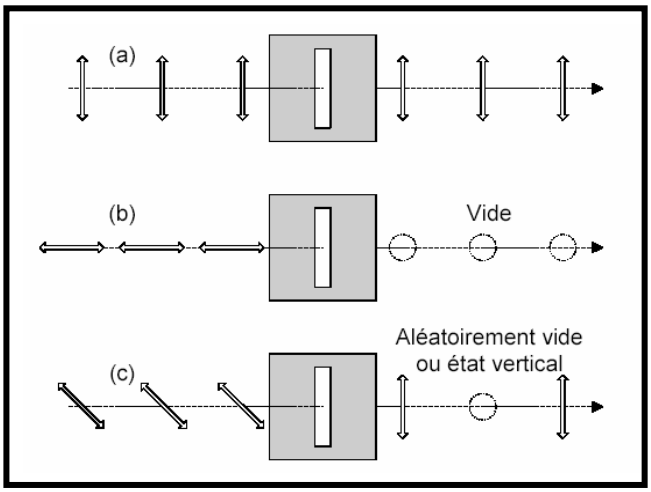


Figure I.13. *Photon unique traversant un filtre ne laissant passer que la lumière polarisée verticalement :*
(a) *Les états polarisés verticalement traversent le filtre sans être absorbés*
(b) *Les états polarisés horizontalement sont tous absorbés*
(c)*Les états polarisés diagonalement sont aléatoirement absorbés ou transmis*

Pour mieux comprendre ce type de cryptographie, on va l'aborder à partir d'un exemple dont le protocole est bien : le protocole BB84. Avant de commencer, on doit mentionner qu'il existe deux modes de polarisations possibles [www9] :

- **Mode 1 :** "0" est codé par un photon d'axe de polarisation 0° et "1" par un photon de polarisation 90°.
- **Mode 2 :** "0" est codé par un photon d'axe de polarisation 45° et "1" par un photon de polarisation 135°.

Pour plus de clarté, ces deux modes peuvent être schématisés comme suit :

Figure I.14. *Les deux modes de polarisation.*

Cet exemple se déroule en plusieurs étapes que l'on va présenter ci-dessous. On a Alice l'expéditrice, Bob le destinataire et Oscar l'éventuel espion [www9]:

1. Alice génère aléatoirement un bit selon un mode (mode 1 ou mode 2) choisi lui aussi aléatoirement, ce que la physique quantique permet, et transmet le photon obtenu à Bob. Elle répète cette opération autant de fois que nécessaire.
2. Pour chaque photon reçu, Bob choisit aléatoirement un mode, c'est-à-dire un polariseur, et note si le photon traverse le polariseur.
3. Ensuite une phase de réconciliation : Alice et Bob communiquent entre eux et pour chaque photon ils comparent si leurs modes coïncident, si c'est le cas, ils ont le même bit, qu'ils conservent, sinon ils ne conservent pas le bit en question. Vue le nombre de photons inutiles, il est donc nécessaire de prévoir un excédent suffisant de photons pour obtenir une clé de longueur donnée. Comme Bob a une chance sur deux de choisir le bon mode, en moyenne on observe ***N*** erreurs pour ***2N*** photons envoyés.
4. Enfin, Alice et Bob contrôlent la sûreté de la clé : parmi les bits conservés ils en choisissent un certain nombre qu'ils comparent publiquement, s'ils ont été espionnés ils obtiendront en moyenne 25% de bits différents (Comme Oscar ne peut pas cloner les photons, il est dans l'obligation de les intercepter, de les mesurer et de renvoyer à Bob un photon similaire à celui qu'il a mesuré. Cependant il a une chance sur deux de choisir le mauvais mode et donc de renvoyer un photon différent de celui qu'Alice avait envoyé à Bob. Puis comme Bob a lui aussi une chance sur deux d'avoir le bon mode, en cas d'espionnage pour chaque photon on a une chance sur 4 (25%) de détecter l'intrusion). Dans ce cas, ils n'utiliseront pas la clé. S'ils n'ont pas de différences, alors ils peuvent conserver la clé pour l'utiliser ultérieurement.

Alice et Bob décident alors de sacrifier une partie de leur clé commune et les comparent publiquement par le canal radio (comme exemple de canal). Dans le Tableau I.3 ils ont quatre bits différents c.à.d. N bits parmi 2N. Ainsi la clé obtenue : 0011.

Photon envoyé par Alice								
Mode choisi par Bob	Mode 1	Mode 2	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								
Après réconciliation								

Tableau I.3. *Exemple sans Oscar.*

Photon envoyé par Alice								
Mode choisi par Oscar	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 2	Mode 2
Résultat mesuré et renvoyée à Bob par Oscar								
Mode choisi par Bob	Mode 1	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								

Tableau I.4. *Exemple avec Oscar.*

Plusieurs cas de figure se présentent à nous dans le Tableau I.4 [Camp, 2010] :

1. Oscar et Bob ont tous les deux le bon mode, alors Oscar mesure bien la bonne polarisation qu'il renvoie à Bob et puis de même pour Bob. Le bit est valable pour la clé et Oscar est passé inaperçue. Exemple : premier photon envoyé.
2. Oscar a le bon mode, mais pas Bob, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon. Oscar passe encore inaperçue, mais sa bonne mesure lui est inutile. Exemple : le deuxième photon envoyé.
3. Oscar a le mauvais mode, mais Bob a le bon mode, alors d'après ce que l'on a vu, Bob a une chance sur deux d'obtenir une mesure compatible avec ce qu'à envoyer Alice. Donc Oscar a une chance sur deux d'être détecté, de plus sa mesure est inutile. Exemple : sixième photon envoyé (avec détection) et dernier photon envoyé (sans détection).
4. Oscar et Bob ont le mauvais mode, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon, Oscar passe donc inaperçue. Exemple : cinquième photon envoyé.

Une fois la réconciliation terminée il ne reste que les bits de la possibilité 1 et 3 qui sont équiprobables. Pour la première possibilité Oscar passe inaperçue, et pour la troisième il a une chance sur deux de se faire détecter. On retrouve bien le 25% ($\frac{1}{2}\frac{1}{2} = \frac{1}{4}$) vu ci-dessus.

b. Conclusion

Le MIT (Massachussets Institute of Technology) a récemment classé la cryptographie quantique parmi les «10 inventions qui changeront le monde». Si on voit peu les avantages à l'heure actuelle, les possibilités pour le futur sont impressionnantes. Le centre canadien IRCA est un leader mondial en recherche quantique, comptant parmi ses membres des chercheurs montréalais au front de cette révolution du monde des télécommunications. Avec le développement du calcul et de la cryptographie quantique, on peut espérer qu'Alice et Bob pourront un jour communiquer en toute quiétude. Entre temps, considérons d'un œil sceptique les soi-disant algorithmes « incassables » qui n'attendent que le prochain superordinateur pour céder comme une coquille d'œuf [Camp, 2010].

I.3.3.4. Algorithme de chiffrement évolutionniste OTL

Omary Fouzia a développé en 2006 un nouvel algorithme de chiffrement en le simulant à un problème d'optimisation dont la résolution est par les algorithmes génétiques. Le but de cet algorithme est de modifier au maximum les fréquences d'apparition des caractères dans le message à chiffrer et d'établir le plus de désordre dans leurs positions. Ces caractères appartiennent à l'ensemble des 256 caractères du code ASCII. L'application de l'algorithme est précédée d'une phase de brouillage du texte initial (M_0) qui peut combiner plusieurs méthodes simples comme les substitutions, les permutations, le chiffrement affiné, etc..., pour obtenir un texte initialement chiffré (M_0').

Pour ce faire, le codage adopté pour représenter les individus, qui sont les différents messages, est résumé à travers la figure ci-dessous :

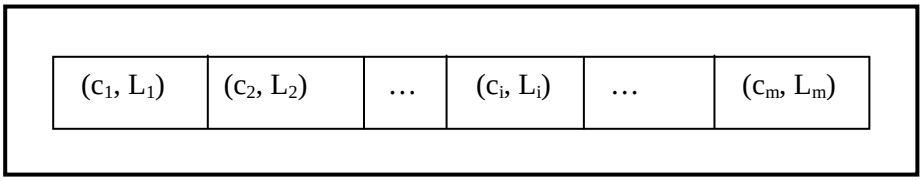


Figure I.15. *Codage des individus.*

Avec :

c_i : un caractère appartenant au message.
L_i : la liste des positions du caractère c_i.
m : le nombre des différents caractères du message.
$L_i \cap L_j = \emptyset, \ \forall i, j \in [1, m]$

L'algorithme OTL cherche à changer itérativement la répartition des listes L_i sur les différents caractères du message (sans modifier le contenu des listes) de telle manière que la différence entre le cardinal de la nouvelle liste affectée à chaque caractère c_i et le cardinal de la liste L_i d'origine soit maximale [Omar, 2006]. Cette variation, en terme de répartition, est assurée grâce aux opérateurs génétiques choisis (MPX pour le croisement avec un taux compris entre 60% et 100%, et une simple permutation pour la mutation avec un taux compris entre 0.1% et 5%). Et pour juger la pertinence des individus construits, la fonction

d'évaluation utilisée est celle donnée ci-dessous. Les meilleurs individus sont ensuite sélectionnés par le biais de la méthode classique de la roulette, en vue de se reproduire.

$$F(X_j) = \sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right|$$

Le processus évolutionnaire est répété jusqu'à la satisfaction d'un critère d'arrêt exprimé à l'aide de la fonction F. Cette dernière est bornée car : $0 \le F(X) \le 2 * l$ (l est la taille du message) pour tout individu X. En fait [Omar, 2006] :

$$\sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right| \le \sum_{i=1}^{m} (card(L_{j_i}) + card(L_i)) \le 2 * l$$

L'opération de déchiffrement, quand à elle, se fait en deux étapes :

1) Tout d'abord, le texte chiffré est représenté suivant le codage proposé en une suite de listes de positions, et c'est grâce à la clé génétique que les caractères vont retrouver leurs listes de position correspondantes dans le message en clair. En effet, la clé, qui peut être d'un usage symétrique ou asymétrique, est une permutation de {1,2,…, m}. Comme résultat, nous obtenons le message M_0'.
2) La deuxième étape, correspond au déchiffrement du message M_0' pour obtenir M_0.

I.4. Conclusion

Dans ce chapitre, nous avons présenté les différentes catégories de cryptographie depuis sa première apparition jusqu'à nos jours. Un état de l'art aussi riche que possible sur les plus célèbres algorithmes de chiffrement, ainsi les fameuses ruses des cryptanalystes ont été exposées. D'après cette étude, un système cryptographique est considéré comme sûr si personne n'a encore mis en défaut sa sécurité. Nous avons vu que ni la cryptographie classique ni symétrique n'a pu s'assurer ce besoin dû aux leurs inconvénients. Pour résoudre cela, les cryptographes ont cherché à déplacer la difficulté ; plutôt que d'utiliser de simples substitutions et de faire reposer la sécurité sur le nombre de clés possibles, ils ont essayé de faire reposer la sécurité sur des difficultés calculatoires. Cela a donné naissance aux algorithmes de cryptographie asymétrique. Cependant il s'avère que l'augmentation constante de la puissance de calcul de nos machines nécessite d'augmenter constamment la difficulté de nos algorithmes. Il se peut en effet qu'une nouvelle technologie anéantisse toute la difficulté d'un algorithme. La cryptographie hybride est aussi un sujet d'attaque puisqu'elle n'est qu'une combinaison de la cryptographie symétrique et asymétrique. Une nouvelle tendance basée sur la cryptographie quantique a aussi été développée. Cette cryptographie est sûre, néanmoins elle est basée sur des principes beaucoup plus théorique et lourds.

Enfin, à partir de la richesse et de la variété des modèles mathématiques exploités dans le domaine de la cryptographie, de nouvelles approches cryptographiques peuvent être proposées par exploitation de méthodes approchées. Le précédent chapitre présentera de telles méthodes de résolution de problèmes : les métaheuristiques.

CHAPITRE II

METAHEURISTIQUES

II.1. Introduction

Un très grand nombre de méthodes existent en RO et en IA pour résoudre différentes sortes de problèmes tels que les problèmes d'optimisation combinatoire. D'une manière très générale, les méthodes de résolution suivent quatre approches différentes pour la recherche d'une solution [JinK, 1999] : l'approche de construction, l'approche de relaxation, l'approche de voisinage et l'approche d'évolution. Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative : par exemple, la méthode de Lin et Kernighan qui resta longtemps le champion des algorithmes pour le problème du voyageur de commerce [LinS, 1973].

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques* [Reev, 1993], [Aart, 1997]. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [Lapo, 1996], [Osma, 1996]. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi, les métaheuristiques sont conçues pour résoudre des problèmes d'optimisation complexes où d'autres méthodes d'optimisation ne sont pas efficaces. Ces méthodes ont fini par être reconnues comme l'une des approches les plus pratiques pour résoudre des problèmes nombreux et complexes, et ceci est particulièrement vrai pour les divers problèmes du monde réel qui sont de nature combinatoire. L'avantage pratique de métaheuristiques réside dans leur efficacité et leur application générale. Dans la littérature et au début des recherches, des heuristiques spécialisées ont été généralement développées pour résoudre des problèmes complexes d'optimisation combinatoire.
D'autre part, avec l'émergence de stratégies des solutions plus générales, y compris les métaheuristiques telles que la recherche taboue, algorithmes génétiques, recuit simulé, le principal défi est devenu l'adaptation des méta-heuristiques à un problème particulier ou une catégorie de problème. Cela nécessite généralement beaucoup moins de travail que de développer une heuristique spécialisée pour une application spécifique, ce qui rend les métaheuristiques un choix attrayant pour la mise en œuvre dans les logiciels à usage général. En outre, une bonne mise en œuvre de métaheuristique est susceptible de fournir des solutions quasi optimales en temps de calcul raisonnables.

Ainsi, une métaheuristique est définie de manière similaire à une heuristique (qui est une méthode conçue pour un problème d'optimisation donné et qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème), mais à un niveau d'abstraction plus élevé (d'après E. Taillard).

II.2. Principes

Les métaheuristiques sont apparues dans les années 80. Ce sont des méthodes d'optimisation, de type stochastique, conçus pour les problèmes difficiles. Des progrès importants ont été réalisés avec l'apparition de nouvelles générations de méthodes approchées puissantes et générales, souvent appelées métaheuristiques [Reev, 1993] [Aart, 1997].

Le terme « métaheuristique » a été initialement utilisé par F. Glover pour distinguer la méthode tabou des heuristiques spécifiques [Glov, 1986]. Notons que ce terme est également utilisé par J-L. Laurière dans son système de résolution Alice [Laur, 1978].

L'origine du mot méta heuristique nous aide à comprendre sa signification. En grec: « Méta » signifie « au-delà », ou « au niveau supérieur », et « Heuristique» veut dire « Trouver ». Ainsi, l'interprétation de ces mots peut signifier que l'on effectue des recherches à un très haut niveau et que la procédure algorithmique regroupe plusieurs heuristiques.

D'après Glover, metaheuristique "se réfère à une stratégie maître qui guide et modifie d'autres heuristiques pour produire des solutions au-delà de ceux qui sont normalement générés dans une quête d'optimalité locale" [Glov, 1997].

Selon [Glov, 1997] « métaheuristiques dans leurs formes modernes sont basées sur une variété d'interprétations de ce qui constitue une recherche intelligente », où le terme de recherche intelligente a été mis en évidence par Pearl [Poire, 1984] (en ce qui concerne heuristiques dans un contexte de l'intelligence artificielle) et [vobs, 1993] (en ce qui concerne le contexte de la recherche opérationnelle). Dans ce sens, on peut aussi considérer la définition suivante: « Une métaheuristique est un processus à générations itératives qui guide

une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter les espaces de recherche en utilisant des stratégies lesarning pour structurer l'information afin de trouver des solutions quasi-optimales » [Osma, 1996]

Pour résumer, la définition suivante semble être la plus appropriée : « Une métaheuristique est un processus maître itératif qui guide et modifie les opérations d'heuristiques subordonnées pour produire efficacement des solutions de haute qualité. Il peut manipuler une solution unique complète (ou incomplète) ou un ensemble de solutions à chaque itération. Les heuristiques subordonnées peuvent être des procédures de niveau élevé (ou faible), ou une recherche locale simple, ou tout simplement une méthode de construction. La famille de métaheuristiques comprend, mais n'est pas limitée à, des procédures de mémoire d'adaptation, de recherche tabou, des systèmes de fourmis, avides de recherche randomisée d'adaptation, de recherche à voisinage variable, des méthodes évolutionnaires, des algorithmes génétiques, de recherche de points, des réseaux neuronaux, de recuit simulé, et leurs hybrides. " [VobS, 1999]

Le but des métaheuristiques est de minimiser ou de maximiser une, ou plusieurs fonctions objectives. Comme exemple, on peut demander de trouver le plus court chemin entre un point A et un point B, sachant que des obstacles sont présents donc il s'agit d'un problème de minimisation ; ou si on veut trouver comment effectuer le plus grand nombre de tâches en un temps limité alors c'est la maximisation du travail à faire.

L'évolution des métaheuristiques se fait de manière itérative. Ceci a l'avantage de permettre l'arrêt de l'algorithme quand on le souhaite, et de récupérer la meilleure solution trouvée jusqu'à présent sans être obligé d'attendre la fin de l'exécution. Un autre point important des métaheuristiques est qu'elles font évoluer des solutions en les améliorant à chaque itération.

Donc, les métaheuristiques sont généralement des algorithmes itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction. Les itérations successives permettent d'évoluer d'une solution peu satisfaisante à la solution la plus adaptée. Les métaheuristiques se comportent, donc, comme des algorithmes de recherche tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution.

Les métaheuristiques sont souvent inspirées de systèmes naturels, qu'ils soient pris en physique -cas du recuit simulé-, en biologie de l'évolution -cas des algorithmes génétiques- ou encore en éthologie -cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires.
Les métaheuristiques désignées par M sont, souvent, des algorithmes utilisant un échantillonnage probabiliste. Elles tentent de trouver l'optimum global (G) d'un problème d'optimisation difficile (avec des discontinuités (D), par exemple), sans être piégées par les optima locaux (L). Ceci est indiqué à la figure suivante :

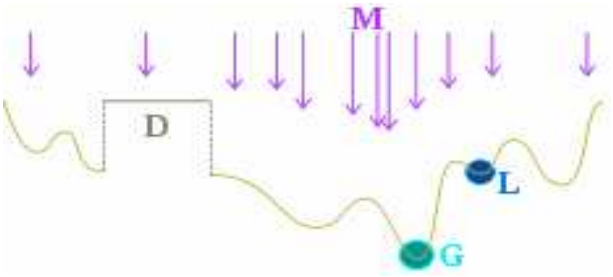


Figure II.1. *Principe général des métaheuristiques.*

II.3. Organisation d'une métaheuristique

D'une manière générale, les métaheuristiques s'articulent autour des principales notions suivantes :

II.3.1. Le Voisinage

Le voisinage d'une solution est un sous-ensemble de solutions qu'il est possible d'atteindre par une série de transformations données.
Formellement parlant [JinK, 1999], soit X l'ensemble des configurations admissibles d'un problème, on appelle *voisinage* toute application $N : X \to 2^X$. On appelle *mécanisme d'exploration* du voisinage toute procédure qui précise comment la recherche passe d'une configuration $s \in X$ à une configuration $s' \in N(s)$. Une configuration *s* est un *optimum local* par rapport au voisinage *N* si $f(s) \leq f(s')$ pour toute configuration $s' \in N(s)$.

Une méthode typique de voisinage débute avec une configuration initiale, et réalise ensuite un processus itératif qui consiste à remplacer la configuration courante par l'un de ses voisins en tenant compte de la fonction de coût. Ce processus s'arrête et retourne la meilleure configuration trouvée quand la condition d'arrêt est réalisée.

Un des avantages de cette stratégie de recherche réside précisément dans la possibilité de contrôler le temps de calcul : la qualité de la solution trouvée tend à s'améliorer progressivement au cours du temps et l'utilisateur est libre d'arrêter l'exécution au moment qu'il aura choisi. Dans la littérature, plusieurs voisinages ainsi que plusieurs stratégies de parcours de ces voisinages ont été définies [Gold, 1989], [Hert, 2003].

II.3.2. Diversification, intensification et apprentissage

La *diversification* (ou *exploration*, synonyme utilisé presque indifféremment dans la littérature des algorithmes évolutionnaires) désigne les processus visant à récolter de l'information sur le problème optimisé. L'*intensification* (ou *exploitation*) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Autrement dit, l'intensification insiste sur la capacité d'examiner en profondeur par une méthode des zones de recherche particulières alors que la diversification met en avant la capacité de découvrir des zones de recherche prometteuses.

La *mémoire* est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi les optima locaux.

Les métaheuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage, ou en mêlant ces notions de façon plus

étroites. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

Les notions d'intensification et de diversifications sont prépondérantes dans la conception des métaheuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherches. En effet, l'application systématique du seul principe d'exploitation ne permet pas une recherche efficace. En effet, l'exploitation conduit à confiner la recherche dans une zone limitée qui finit par s'épuiser. Le cas de l'amélioration itérative rapidement piégée dans un optimum local illustre cruellement ce phénomène. Une autre illustration souvent évoquée est fournie par le problème de convergence prématurée des algorithmes génétiques : du fait de la sélection, la population finit par n'être constituée que d'individus tous similaires. L'une des préoccupations majeures dans les algorithmes génétiques consiste d'ailleurs à préserver le plus longtemps possible une diversité suffisante dans la population. Face à ce type de difficulté, la solution consiste à diriger la poursuite de la recherche vers de nouvelles zones, *i.e.*, à recourir à l'exploration. Les deux notions ne sont, donc, pas contradictoires mais complémentaires, et il existe de nombreuses stratégies mêlant à la fois l'un et l'autre des aspects.

II.4. Méthodes générales et méthodes spécifiques

Contrairement aux algorithmes traditionnels dédiés à un problème spécifique, les métaheuristiques constituent des mécanismes très généraux qui peuvent être adaptés pour traiter de nombreux problèmes différents. Pour expliquer l'efficacité d'un algorithme spécifique, on invoque souvent le fait qu'il utilise des connaissances spécifiques du problème. Pour expliquer l'efficacité d'une métaheuristique, deux types d'arguments opposés sont généralement avancés.

Selon certains, des mécanismes généraux suffisamment puissants ont par eux mêmes la faculté de mener efficacement la recherche sans disposer d'information spécifique du problème considéré (contexte de boîte noire) : c'était notamment le point de vue classique porté sur les algorithmes génétiques [Gold, 1989]. Malheureusement, de même qu'il ne peut pas exister de stratégie avantageuse dans un jeu de hasard comme la roulette, il existe également des limitations théoriques fondamentales qui ruinent les espoirs d'une méthode aveugle dans le cas le plus général. En fait, le théorème « *No Free Lunch (NFL)* » [Wolp, 1997] montre que pour les problèmes de type boîte noire, toutes les méthodes sont équivalentes et font aussi bien, ou plutôt aussi mal, que l'énumération aléatoire.

Selon le point de vue opposé, la puissance d'une métaheuristique est d'abord liée à son aptitude à intégrer des connaissances spécifiques du problème. La connaissance du problème la plus fondamentale réside dans le codage du problème et dans le choix de la fonction de voisinage. Plusieurs auteurs insistent sur l'importance du codage du problème, voir par exemple [Radc, 1995]. Dans le cas du voyageur de commerce, plusieurs types de codages différents ont été proposés et conduisent à des performances très variées [Mich, 1992]. En général, il n'existe pas de codage universellement efficace. Un « bon » codage doit permettre de restreindre l'espace de recherche et d'intégrer des connaissances du problème.

Les métaheuristiques tentent également d'améliorer leur efficacité en incorporant des connaissances supplémentaires dans leurs opérateurs. Plus les opérateurs d'une méthode utilisent des connaissances spécifiques, plus la méthode dispose de moyens potentiels pour conduire efficacement la recherche. En contrepartie, l'intégration de ces connaissances spécifiques (en supposant que ces connaissances soient disponibles) nécessite un effort pour spécialiser ou adapter la méthode. La méthode tabou vise à incorporer le plus possible de connaissances du problème pour atteindre le maximum d'efficacité : l'utilisateur doit notamment définir de façon pertinente le type de caractéristique figurant dans la liste tabou.

De leur coté, les algorithmes génétiques se sont éloignés du modèle standard pour intégrer également des connaissances du problème : codage non binaire, opérateurs spécifiques. Au contraire, le recuit simulé est parfois présenté comme une méthode facile à adapter à un problème et qui ne tente pas d'exploiter de connaissances spécifiques.

En général, une méthode offrant des possibilités d'intégrer des connaissances du problème a plus de chance de produire de bons résultats, mais demande un effort d'adaptation et de spécialisation. Au contraire, une méthode très générale qui prétend n'intégrer aucune connaissance propre ne peut pas être compétitive.

II.5. Classification des métaheuristiques

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents.
Dans la littérature, plusieurs classifications de métaheuristiques ont été proposées. Nous présentons à ce niveau un aperçu des principales classifications.

II.5.1. Les métaheuristiques inspirées et non inspirées d'un phénomène naturel

Une manière intuitive de classifier les métaheuristiques consiste à séparer celles qui sont inspirées d'un phénomène naturel, de celles qui ne le sont pas.

Les algorithmes génétiques ou les algorithmes par colonies de fourmi entrent clairement dans la première catégorie, tandis que la méthode de descente, ou la recherche Tabou, vont dans la seconde.

II.5.2 Les métaheuristiques avec fonction objective statique ou dynamique

Les métaheuristiques peuvent être aussi classées selon la façon dont ils utilisent la fonction objective. Certains algorithmes conservent la fonction objective donnée dans la représentation de problème telle qu'elle est (comme la recherche locale guidée (GLS)) et la modifie lors de la recherche. L'idée de cette approche est d'éviter les minimums locaux en modifiant l'espace de recherche. En conséquence, lors de la recherche la fonction objective est altérée en essayant d'intégrer des informations intégrées au cours du processus de recherche.

II.5.3. Les métaheuristiques avec une ou plusieurs structures de voisinage

La plupart des métaheuristiques travaillent sur une seule structure de voisinage. En d'autres termes, la topologie du paysage de fitness ne change pas en cours de l'algorithme. D'autres métaheuristiques telles que la recherche par voisinage variable utilisent un ensemble de structures de voisinage qui donne la possibilité de diversifier la recherche en échangeant entre les structures de voisinage qui ont des formes différentes.

II.5.4. Les métaheuristiques avec et sans mémoire

Les métaheuristiques utilisent l'historique de leur recherche pour guider l'optimisation aux itérations suivantes. Dans le cas le plus simple, elles se limitent à considérer l'état de la recherche à une itération donnée pour déterminer la prochaine itération : il s'agit alors d'un processus de décision markovien, et on parlera de méthode *sans mémoire*. C'est le cas de la plupart des méthodes de recherche locale (recherche à voisinage variable, recherche locale

itérée, recherche locale stochastique, recherche locale guidée), algorithme d'estimation de distribution, recuit simulé, GRASP.

Beaucoup de métaheuristiques utilisent une mémoire plus évoluée, que ce soit sur le court terme (solutions visitées récemment, par exemple) ou sur le long terme (mémorisation d'un ensemble de paramètres synthétiques décrivant la recherche).

II.5.5. Les métaheuristiques implicite, explicite, directe

En considérant les métaheuristiques comme des méthodes itératives utilisant un échantillonnage de la fonction objective comme base d'apprentissage (définition plus particulièrement adaptée aux métaheuristiques à populations) apparaît le problème du choix de l'échantillonnage.

Dans la très grande majorité des cas, cet échantillonnage se fait sur une base aléatoire, et peut donc être décrit via une distribution de probabilités. Il existe alors trois classes de métaheuristiques, selon l'approche utilisée pour manipuler cette distribution.

La première classe est celle des méthodes *implicites*, où la distribution de probabilité n'est pas connue *a priori*. C'est le cas par exemple des algorithmes génétiques, où le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. L'évolution différentielle, Scatter search, algorithmes à essaim de particules, stratégies d'évolution et algorithmes de colonie de fourmis sont des méthodes implicites.

Par opposition, on peut donc classer les méthodes *explicites*, qui utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution.

Dans cette classification, le recuit simulé occupe une place particulière, puisqu'on peut considérer qu'il échantillonne la fonction objective en utilisant directement celle-ci comme distribution de probabilité (les meilleures solutions ayant une probabilité plus grande d'être tirées). Il n'est donc ni explicite ni implicite, mais plutôt « direct » (exemple : le recuit simulé) [JinK, 1999].

II.5.6. Les métaheuristiques évolutionnaires et non évolutionnaires

On trouve parfois une classification présentant les algorithmes d'optimisations stochastiques comme étant « évolutionnaires » (ou « évolutionnistes ») ou non. L'algorithme sera considéré comme faisant partie de la classe des algorithmes évolutionnaires s'il manipule une population *via* des *opérateurs*, selon un algorithme général donné.

Cette façon de présenter les métaheuristiques dispose d'une nomenclature adaptée : on parlera d'opérateurs pour toute action modifiant l'état d'une ou plusieurs solutions. Un opérateur construisant une nouvelle solution sera dénommé *générateur*, alors qu'un opérateur modifiant une solution existante sera appelé *mutateur*.

Dans cette optique, la structure générale des algorithmes évolutionnaires enchaîne des étapes de *sélection*, de *reproduction* (ou *croisement*), de *mutation* et enfin de *remplacement*. Chaque étape utilise des opérateurs plus ou moins spécifiques.

Quelques algorithmes évolutionnaires : l'algorithme génétique, la programmation génétique, l'algorithme d'évolution différentielle, les stratégies évolutionnaires et l'algorithme d'estimation de distribution [JinK, 1999].

II.5.7. Les métaheuristiques à base de population et les métaheuristiques à trajectoire

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

Les plus connues dans cette classe sont le recuit simulé, la recherche tabou, la recherche à voisinage variable, la méthode GRASP. L'autre approche utilise la notion de population. La métaheuristique manipule un ensemble de solutions en parallèle, à chaque itération. On peut citer les algorithmes génétiques, l'optimisation par essaims particulaires et les algorithmes de colonies de fourmis.

La frontière est parfois floue entre ces deux classes. On peut ainsi considérer qu'un recuit simulé où la température baisse par paliers, a une structure à population. En effet, dans ce cas on manipule un ensemble de points à chaque palier, il s'agit simplement d'une méthode d'échantillonnage particulière.

II.5.7.1. Les métaheuristiques à trajectoire (à solution unique)

Ici, nous résumons les plus connues des métaheuristiques à trajectoire.

a. La méthode de descente

Le principe de la méthode de descente (dite aussi *basic local search*) consiste, à partir d'une solution *S*, à choisir une solution *S'* dans un voisinage de *S* telle que *S' améliore* la recherche (généralement telle que $f(S') < f(S)$). On peut décider soit d'examiner toutes les solutions du voisinage *N* et prendre la meilleure de toutes ces solutions (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.
La méthode de descente est la méthode la plus élémentaire de recherche locale. On peut la formaliser comme suit :

Algorithme II.1 *descente_simple* (solution initiale *S*)

Début
  Répéter
    Choisir *S'* dans N(*S*)
    Si $f(S') < f(S)$ alors $S \leftarrow S'$
  Jusqu'à ce que $f(S') \geq f(S)$, $\forall S' \in$ **S**
Fin

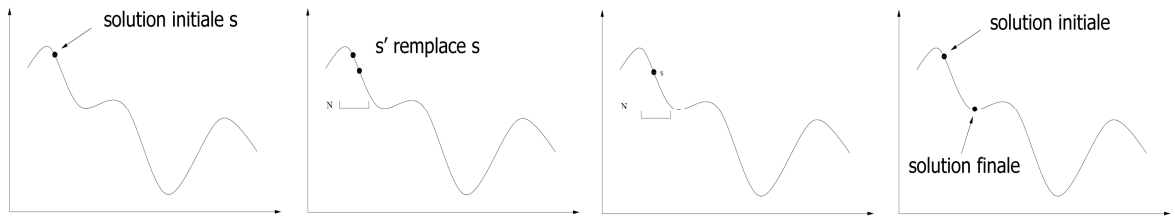


Figure II.2. *Evolution d'une solution dans la méthode de descente.*

On peut varier cette méthode en choisissant à chaque fois la solution S' dans N(S) qui améliore le plus la valeur de f. C'est la méthode de plus grande descente [Cost, 2006].

b. Recherche aléatoire (Random Search : RS)

C'est la plus simple des méthodes stochastiques. Cette méthode consiste à tirer à chaque itération une solution au hasard. La fonction objective *f* est évaluée en ce point. La nouvelle valeur est comparée à la précédente. Si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continu. Sinon on repart du point précédent et on recommence le procédé, jusqu'à ce que les conditions d'arrêt soient atteintes. L'algorithme II.2 [Luke, 2010] présente un pseudo code de la recherche aléatoire dans le cas d'un problème de minimisation.

Algorithme II.2 *Recherche aléatoire*

Début
$S_0 \leftarrow$ solution aléatoire;
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$;
Répéter
S ← solution aléatoire ;
Si ($f(S) < f_{min}$) *Alors*
$f_{min} \leftarrow f(S)$;
$x_{min} \leftarrow S$;
Fin Si
Jusqu'à conditions d'arrêt satisfaites.
Fin

c. Hill Climbing (HC)

On part d'une solution si possible "bonne" (par exemple donnée par une heuristique) et on balaie l'ensemble des voisins de cette solution ; s'il n'existe pas de voisin meilleur que notre solution, cela veut dire qu'un optimum local a été trouvé et on arrête la recherche. Sinon, on choisit le meilleur des voisins et on recommence. Une autre implémentation consiste non pas à passer au meilleur des voisins à chaque étape, mais au premier meilleur voisin trouvé. La convergence vers un optimum local pouvant être très lente, on peut éventuellement fixer un nombre de boucles maximum, si on veut limiter le temps d'exécution [Sea, 2010].

Cette méthode a l'inconvénient de rester bloquée dans un optimum local : une fois un optimum local trouvé, on s'arrête, même si ce n'est pas l'optimum global. Selon le paysage des solutions, l'optimum local peut être très bon ou très mauvais par rapport à l'optimum global. Si la solution de départ est donnée par une heuristique déterministe, l'algorithme sera déterministe. Si elle est tirée au hasard, l'algorithme devient non déterministe et donc plusieurs exécutions différentes sur la même instance pourront donner des solutions différentes et de qualités différentes [Sea, 2010].

La notion de voisinage est primordiale. Si les voisins sont très nombreux, on a de fortes chances de trouver l'optimum global, mais visiter un voisinage peut être très long: on visitera une grande partie de l'espace des solutions. Si le voisinage est très restreint, on risque fort de rester bloqué dans un optimum local de "mauvaise qualité". Le choix de la notion de voisinage est un compromis entre efficacité et qualité.

Le pseudo-code de l'algorithme Hill Climbing est celui présenté par l'algorithme II.3 [Sea, 2010]:

Algorithme II.3 *HILL CLIMBING*

Début
$S_0 \leftarrow$ Solution aléatoire
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$
Répéter
Engendrer un N-échantillon $S_i(x)$ voisinage de S(x) *ET* calculer
$f(S(x)) = \min[f(S_i(x))]$ avec $1 \leq i \leq N$

Si ($f(S) < f_{min}$) *then*
$x_{min} \leftarrow S$
Sinon
Sortir de Répéter
Fin Si
Jusqu'à condition d'arrêt satisfaite.
Fin.

d. Recuit Simulé

Le recuit simulé a été introduit et mis au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985. Il a été repris sous différentes formes par Lawrence Davis en 1987 [Lawr, 1987].
Cet algorithme a été dérivé de l'algorithme de Metropolis, développé par les scientifiques du projet ex-Manhattan : Nicholas Metropolis, Arianna et Marshall Rosenbluth, Augusta et Edward Teller en 1953. Ce dernier permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. Partant d'une solution donnée, en la modifiant, on en obtient une seconde. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

La description des phénomènes physiques et quantiques liés au processus de recuit s'appuie sur la statistique de Boltzmann. Pour qu'un métal ait une qualité optimale, il faut que son état d'énergie soit minimal. Pour améliorer la qualité d'un métal, on le chauffe, puis on le refroidit par paliers en attendant à chaque fois que l'état d'énergie soit stabilisé. Au niveau de l'atome, cela signifie qu'à une température chaude, les atomes bougent beaucoup, et en se

refroidissant, ils trouvent leur place optimale. Plus la température descend, moins les atomes bougent et le métal devient de plus en plus résistant.

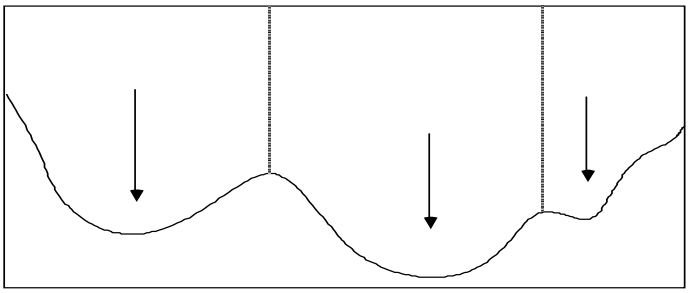


Figure II.3. *Un paysage d'énergie. Suivant la configuration initiale, indiquée par les flèches, la dynamique aboutit à température nulle dans l'un quelconque des minima relatifs, séparés par les barrières indiquées en pointillés. A température élevée, les processus probabilistes permettent au système de sauter les barrières séparant les vallées.*

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. À cette solution correspond une énergie initiale $E = E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T = T_0$ élevée est également choisie. Ce choix est alors totalement arbitraire et va dépendre de la loi de décroissance utilisée.

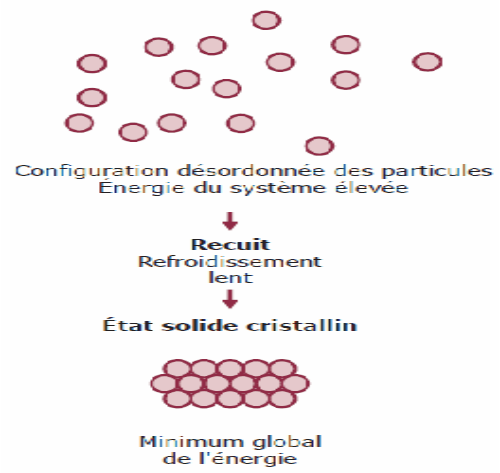


Figure II.4. *Transposition de procédé recuit à la résolution d'un problème d'optimisation.*

À chaque itération de l'algorithme une modification élémentaire de la solution est effectuée. Cette modification entraîne une variation Δ_E de l'énergie du système (toujours calculée à partir du critère que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta_E}{T}}$. Ce choix de l'exponentielle pour la probabilité s'appelle règle de Metropolis.

On itère ensuite selon ce procédé en gardant la température constante.

Algorithme II.4 *Recuit simulé*

```
Début
    Initialisation du temps
    Tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint] Faire
        Initialisation de la température à Tmax
        Tant que [la température est supérieure à 0] et [le temps limite n'est pas atteint] Faire
            Incrémentation du temps
            Modification de la solution courante
            Si [la nouvelle solution est meilleure que l'ancienne] Alors
                la sauvegarder
            Sinon
                Si [le système est stable depuis un nombre défini d'itérations] Alors
                    Diminuer la température
                Fin si
            Fin si
        Fin tant que
    Fin tant que
Fin
```

L'algorithme varie de Hill-Climbing (algorithme 3) dans sa décision du moment de remplacer S, la solution candidate d'origine, avec R, son enfant nouveau peaufiné. Plus précisément, si R est meilleur que S, nous allons toujours remplacer S par R comme d'habitude. Mais si R est pire que S, on peut toujours remplacer S par R avec une certaine probabilité P(t, R, S):

$$P(t, R, S) = e^{\frac{\text{Quality}(R) - \text{Quality}(S)}{t}}$$

Où t> 0. Cette équation est intéressante à deux égards. Notez que la fraction est négative car R est pire que S. Premièrement, si R est bien pire que S, la fraction est plus grande, et donc la probabilité est proche de 0. Si R est très proche de S, la probabilité est proche de 1. Ainsi, si R n'est pas bien pire que S, nous allons toujours choisir R avec une probabilité raisonnable.

Deuxièmement, nous avons un paramètre ajustable t. Si t est proche de 0, la fraction est de nouveau un grand nombre, et donc la probabilité est proche de 0. Si t est élevé, la probabilité est proche de 1. L'idée est d'abord de mettre t comme un grand nombre, ce qui provoque l'algorithme de se déplacer à chaque solution nouvellement créée indépendamment de sa qualité. Nous faisons une marche aléatoire dans l'espace. Puis t diminue lentement, éventuellement à 0, à quel point l'algorithme ne fait rien de plus que de simples Hill-Climbing.

e. Recherche Tabou (Tabu Search : TS)

La recherche avec tabou, ou simplement dite recherche tabou, est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans un article paru en 1986 [Glov, 1986], mais reprenant de nombreuses idées proposées antérieurement dès les années 60 dans les deux articles simplement intitulés « Tabu chearch » [Glov, 1989] proposant la plus part des principes de cette recherche telle qu'elle est décrite actuellement. Maintenant, elle est d'usage très classique en domaine d'optimisation combinatoire pour résoudre les problèmes NP-durs.

L'idée principale de la recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui optimise la fonction objective. Le voisinage choisi est exploré de manière déterministe, il est interdit de reprendre des solutions récemment visitées.

L'un des principes fondamentaux de cette recherche qui la caractérise des autres méta-heuristiques est la construction d'un historique de la recherche itérative ou, ce qui est équivalent, au fait de doter la recherche de mémoire [Glov, 1997]. La recherche tabou examine un échantillonnage de solutions *N(S)* et retient la meilleure solution *S* des solutions obtenues.

A chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche tabou qu'elle est une méthode agressive [www10].

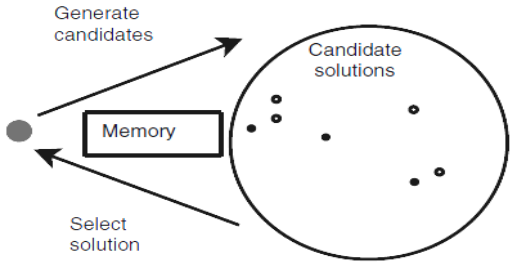
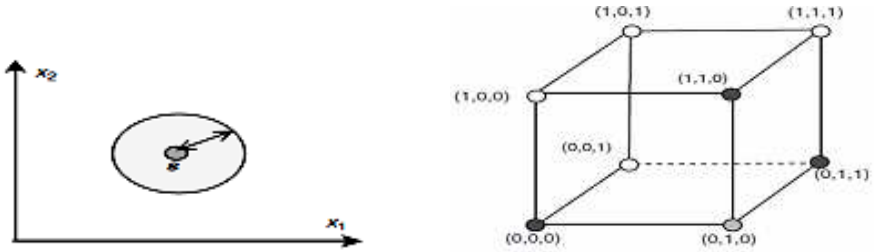


Figure II.5. *Principe de base d'une méta-heuristique à mémoire.*

La recherche tabou est essentiellement axée sur une exploration non triviale de l'ensemble des solutions en utilisant la notion de voisinage. Formellement pour toute solution *s* de *S*, un ensemble de $N(s) \in S$ qu'on appelle ensemble des solutions voisines de *s*.



(a) *(b)*

Figure II.6. *Les types de solutions du voisinage.*
(a) *Le voisinage pour un problème à variables continus : Le cercle symbolise les voisins de la solution s,*
(b) *Le voisinage pour un problème à variables discrètes : Les nœuds du cube sont les solutions et leurs voisins adjacents.*

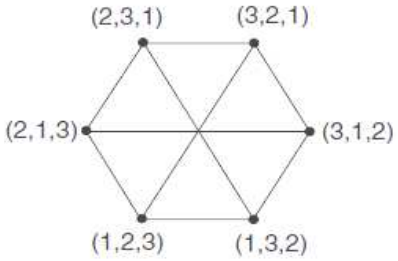


Figure II.7. *Voisinage d'une permutation de taille égale à 3.*
Les voisins de la solution (2.3.1) sont : (3.2.1), (2.1.3), et (1.3.2).

Vue d'un côté pratique l'ensemble N(s) des solutions voisines de s n'est autre que l'ensemble des modifications que l'on peut apporter à cette dernière. On appelle *mouvement* une modification apportée à une solution.

Dans la littérature et dans le cas de la recherche tabou, l'application d'un mouvement *m* à une solution *s* est notée ; d'où l'expression suivante du voisinage :

$$N(s) = \{s' / s' = s \oplus m, m \in M\}$$

La figure II.8 [Kamm, 2006] illustre l'ensemble des mouvements possibles dans le cas d'une solution composée de quatre éléments.

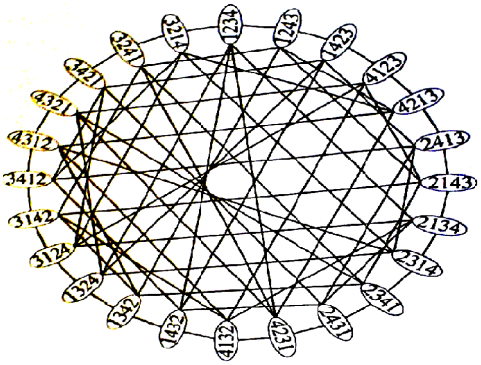


Figure II.8. *Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments.*

La recherche tabou est une méthode itérative qui explore un espace de solutions. Comme toute méthode itérative, il est intéressant d'éviter de revisiter des solutions pour essayer de converger vers la solution optimale le plus rapidement possible et ceci n'est possible qu'à l'aide d'une mémoire caractérisée par une taille définie par l'utilisateur et le problème à résoudre. Cependant, l'utilisation d'une mémoire peut s'avérer dans bien des cas peu efficace, voire mauvaise. Outre le fait de mettre cette idée en pratique est difficile, cela suppose que l'on doit mémoriser chaque solution visitée et de tester à chaque itération et pour chaque solution admissible si cette dernière a été déjà examinée. Les *tables de hachage* permettent de faire cela de manière efficace, mais ceci n'empêche pas la croissance linéaire de la taille de la mémoire avec le nombre des itérations effectuées.

Une table de hachage est définie par un tableau *T* de *J* entiers, avec *J* choisi relativement grand et assez voisin de la capacité de la machine utilisée tout en prenant en considération la taille du problème.

Le but de la table de hachage est :

- ✓ d'interdire le retour aux solutions obtenues pendant les *t* dernières opérations ;
- ✓ ainsi éliminer les cycles de longueurs égales ou inférieures à *t*.

Toutefois, le fait d'interdire un grand nombre de mouvements aura pour conséquence de rendre l'obtention de bons résultats très délicate faute de mouvements disponibles. Si ce nombre diminue, cela augmentera les chances d'une meilleure exploration aux alentours des optimums locaux et d'obtenir ainsi de meilleures solutions. Il ne faut cependant pas trop diminuer ce nombre, car, dans ce cas, il devient très probable de se trouver prisonnier d'un ensemble très restreint de solutions et de les visiter tout le temps de la recherche.

Pour palier à ce compromis du choix de nombre des mouvements à interdire et de bénéficier simultanément des avantages d'un petit nombre, qui permet une visite détaillée du voisinage d'une même solution, et d'un grand nombre qui permet de franchir le voisinage de différentes solutions, ce nombre doit varier au cours du processus itératif. Pour ce faire, plusieurs méthodes existent. Ce nombre peut être tiré au hasard à partir d'un intervalle donné à chaque itération ou après un nombre d'itérations, comme il peut aussi croître ou décroître en fonction des résultats obtenus au cours de la recherche [Tail, 1991], [Tail, 1995] et [Tail, 1999].

Autrement dit, une liste tabou avec trop d'éléments peut devenir très restrictive. En effet, il a été observé que trop de contraintes tabou forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération. Par contre une liste tabou contenant trop peu d'éléments peu s'avérer inutile et mener à des mouvements cycliques [Ayas, 2004].

En plus, dans beaucoup de problèmes, l'interdiction de revisiter des solutions mènerait à des incohérences comme la déconnection de la solution courante de la solution optimale ou bien le blocage de la recherche itérative par cause d'absence de solutions voisines non visitées[Kamm, 2006].

La figure suivante illustre ces derniers cas incohérents :

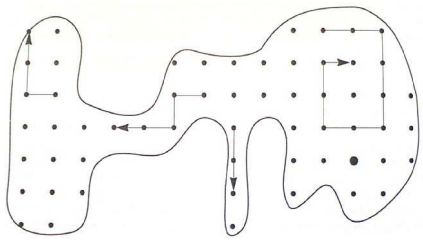


Figure II.9. *Déconnections et blocages dans la recherche tabou.*

Il est à signaler qu'une liste tabou peut être soit statique ou dynamique. Pour la première, la taille de la liste est fixée, en générale au début. Elle dépend de la taille du problème à résoudre et du voisinage choisi. Pour la deuxième, la taille doit changer durant le processus de recherche sans utiliser aucune information sur la mémoire de recherche.

Algorithme II.5 *Recherche Tabou*

Début

Φ Fonction de coût
Variables locales : S solution courante, Liste Taboue L, Meilleure solution M, itération courante K, nombre d'itérations N
Paramétrages : taille de liste taboue, critère d'aspiration
Choix d'une solution initiale S_0
S← S_0
M← S
K ←0
Tant que K< N *faire*
K← K+1
Mise à jour de L
Génération des candidats E par opération de voisinage
C ← Best (E)
Si Φ (S) < Φ(M) *ou* C n'est pas taboue *ou* C vérifie l'aspiration *alors*
S ← C
Sinon
E ← E \ C
Fin tant que
Retourner S

Fin.

f. GRASP

À proprement parler, GRASP (pour *Greedy Randomized Adaptive Search Procedure*) est une méthode hybride, car elle cherche à combiner les avantages des heuristiques gloutonnes, de la recherche aléatoire et des méthodes de voisinage. Un algorithme GRASP répète un processus composé de deux étapes : la construction d'une solution suivie par une descente pour améliorer la solution construite. Durant l'étape de construction, une solution est itérativement construite : chaque itération ajoute un élément dans la solution partielle courante. Pour déterminer l'élément qui sera ajouté, on utilise une liste des meilleurs candidats obtenus avec une fonction gloutonne et on prend *au hasard* un élément dans cette liste. La liste des meilleurs candidats est dynamiquement mise à jour après chaque itération de construction. Cette étape de construction continue jusqu'à ce qu'une solution complète soit obtenue [JinK, 1999].

À partir de cette solution, une descente est appliquée pour améliorer la solution. Une procédure GRASP répète ces deux étapes et retourne à la fin la meilleure solution trouvée. Les deux paramètres de cette méthode sont donc la longueur de la liste de candidats et le nombre d'itérations autorisées.

Algorithme II.6 *GRASP*

```
Début
    x, x*, f* : type Données
    Début
    f* ← infini
    Pour i = 1 à nombre d'itérations faire
        x ← Construction Aléatoire Gloutonne ()
        x ← Recherche Locale(x)
        Si f(x) < f* alors
            x* ← x
            f* ←f(x)
        Fin Si
  Fin Pour
Fin.
```

II.5.7.2. Les métaheuristiques à population

Dans cette section, nous présentons un sommaire des plus connues des métaheuristiques à population.

a. Les algorithmes évolutionnaires

Les *algorithmes évolutionnistes* ou *algorithmes évolutionnaires* (*evolutionary algorithms* ou encore dits *evolutionary computation* en anglais), sont une famille d'algorithmes s'inspirant de la théorie de l'évolution pour résoudre des problèmes divers. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation, elles sont en cela des métaheuristiques, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. On les classe également parmi les méthodes d'intelligence calculatoire.

Selon la génétique et la théorie de l'évolution [www11] :

- ✓ Un enfant hérite son patrimoine génétique pour moitié de sa mère et pour moitié de son père (reproduction sexuée).
- ✓ Les enfants ne sont pas identiques aux parents car des altérations des gènes peuvent se produire (mutations).
- ✓ Parmi les mutations, certaines peuvent être favorables et d'autres défavorables.
- ✓ Il naît beaucoup de descendants : mais seuls les individus les mieux adaptés pourront survivre et transmettre leurs gènes à leur tour à leur descendance.

Dans ce modèle, on observe que [www11] :

- ✓ Le hasard joue un rôle moteur pour produire de nouveaux individus différents de leurs parents.
- ✓ La sélection naturelle effectue le tri entre les variations favorables et les autres.

Ainsi, basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin, ces algorithmes présentent des qualités intéressantes pour la résolution des problèmes d'optimisation. Les individus ou chromosomes d'un algorithme évolutionnaires sont des codages des solutions possibles du problème. Comme dans la nature, ces individus forment une population qui va évoluer dans le temps selon des lois de sélection qui vont favoriser les mieux adaptés à se croiser entre eux en produisant des populations meilleures. L'évolution des individus d'une population à une autre se fait à l'aide de la reproduction. Les individus parents vont se reproduire pour donner des individus enfants qui seront plus performants après avoir subis des opérations génétiques de croisement et de mutation.

Et comme ces reproductions se font avec une part de hasard par analogie à la nature, donc, les parents candidats à la reproduction sont choisis d'une manière probabiliste proportionnelle à leurs aptitudes et l'étape de reproduction est choisie d'une façon totalement aléatoire.

Finalement, passant d'une génération à une autre, les individus forment une progéniture plus performante qui s'approche au mieux de la solution optimale [Kamm, 2006].

La figure suivante résume le principe de résolution de problèmes par algorithmes évolutionnaires.

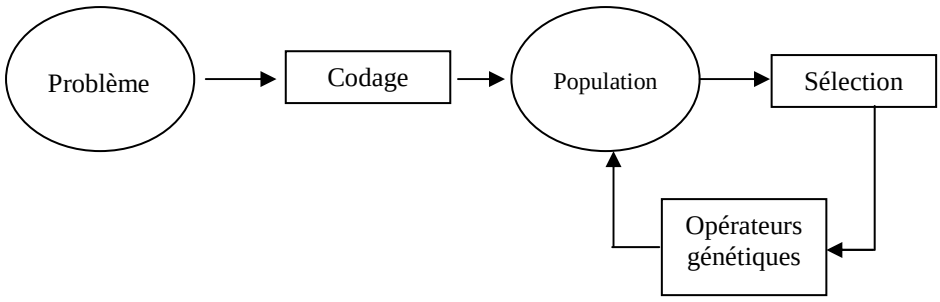


Figure II.10 . *Principe des algorithmes évolutionnaires.*

Ainsi, un algorithme évolutionnaire est construit autour des notions suivantes :

- **Le codage :**

La première étape de la résolution d'un problème à l'aide d'un algorithme génétique est la modélisation appropriée des solutions. Cette modélisation est appelée *codage* et permet de représenter les solutions sous forme de chromosomes.

Le codage se base sur deux notions importantes : le génotype et le phénotype.

- ✓ **Génotype** : représente l'ensemble des valeurs des gènes d'un chromosome.
- ✓ **Phénotype** : c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype.

S'il y a passage immédiat du phénotype au génotype, le codage est dit direct, sinon il est dit indirect et une procédure de passage est indispensable. Le codage d'une solution doit décrire toutes les données du problème et respecter toutes ses contraintes.

- **Population initiale :**

L'espace de recherche étant infini, il est très difficile de répartir plus ou moins équitablement la population initiale sur l'espace de recherche, de sorte que cet espace soit au maximum parcouru. Un certain nombre d'individus doivent être générés aléatoirement s'il n'existe aucune autre méthode d'initialisation.

- **Evaluation de la qualité d'une solution :**

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance (*fitness*) de mesurer cette efficacité pour permettre à l'algorithme de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution. Autrement dit, la fonction de performance *f*, doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre.

- **La sélection :**

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.
Plusieurs modes de sélection ont été définis tels que : sélection par roulette (wheel), sélection par rang, sélection steady-state, élitisme, sélection uniforme, etc.

- **Opérateurs génétiques :**

1) Croisement : C'est un opérateur génétique qui permet à deux chromosomes parents de produire deux chromosomes enfants où de nouvelles séquences de gènes pour les chromosomes enfants sont crées à partir d'une base de configuration des séquences héritées des chromosomes parents. Cet opérateur se produit selon une probabilité *Pc* fixée par l'utilisateur selon le problème à optimiser.

Il existe plusieurs manières d'effectuer un croisement soit par cross-over où l'on a besoin de deux parents qui génèrent à la fin du croisement deux enfants, soit par copie où l'on n'a besoin que d'un seul parent qui nous donnera à la fin du croisement un enfant qui est le parent lui-même (sa copie).

Dans la littérature, il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter [Mesg, 1999]. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que :

- ✓ le croisement à un point.
- ✓ le croisement multipoints.
- ✓ le croisement uniforme.

2) Mutation : permet de changer (permuter) un gène d'un chromosome par un autre d'une manière aléatoire, ce qui est à première vue très ressemblant avec un cross-over. La différence est que le cross-over essaie de converger vers une solution qui lui parait la meilleure (s'intéresse à la qualité), mais que la mutation permet la diversité. En quelque sorte, la mutation sert à éviter une convergence prématurée de l'algorithme. Par exemple, lors de la

recherche d'une solution optimum, la mutation sert à éviter la convergence vers un optimum local.
Cet opérateur est aussi appliqué avec une probabilité P_m. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche aléatoire et conserver le principe de sélection et d'évolution.

Dans la littérature plusieurs opérateurs de mutation ont été définis, tels que : la transposition de deux allèles consécutifs, la transposition de deux allèles quelconques, l'inversion d'allèles, etc.

Les algorithmes évolutionnaires se scindent en grandes familles, dont les principales sont :

1) Programmation évolutionnaire

La programmation évolutionnaire développée par *L.J. Fogel* [Foge, 2003], se base sur l'évolution d'une population d'automates à états finis (Figure II.11) pour résoudre des problèmes de prédiction. Ce modèle évolutionniste accentue l'utilisation de la mutation et n'utilise pas dans sa version originale la recombinaison des individus par croisement [Renn, 2000]. La table de transition des automates est modifiée grâce à des mutations aléatoires uniformes dans l'alphabet discret correspondant. Chaque automate de la population parente génère un enfant par mutation, et les meilleures solutions entre les parents et les enfants sont sélectionnées pour survivre, sachons que l'évaluation de la performance des individus correspond au nombre de symboles prédits correctement.

Par la suite, la programmation évolutionnaire a été développée et son domaine a été élargi par *D.B. Fogel*, afin qu'il puisse travailler dans l'espace réel, où la sélection déterministe est remplacée par un tournoi stochastique.

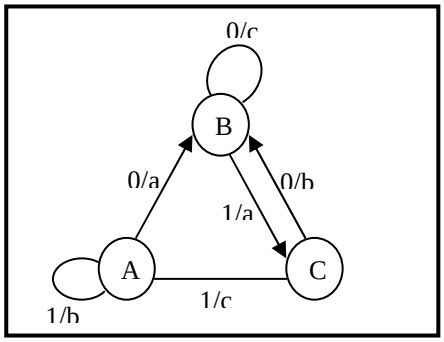


Figure II.11. *Exemple d'un automate à états finis ayant trois états différents S={A,B,C}, un alphabet d'entrée I={0,1}, et un alphabet de sortie O={a,b,c}. Chaque arrête entre deux états indique une transition possible, et la fonction de transition $\delta : S \times I \rightarrow S \times O$ est spécifiée par les labels au niveau des arrêtes ayant la forme i / o, signifiant que $\delta (s_i , i) = (s_j , o)$.*

2) Stratégies d'évolution

Les stratégies d'évolution sont dédiées à l'optimisation des problèmes continus dans l'espace de vecteurs des réels [Magn, 2001]. Les premiers efforts pour la mise en place des stratégies d'évolution ont eu lieu en 1973 à l'université de Berlin par Rechenberg [Rech, 2003] au cours de la résolution d'un problème aérodynamique. C'est avec ces méthodes évolutionnistes que la notion d'auto-adaptativité pour la mutation permettant de contrôler cette fonction de mutation, a été apparue. Une mise en œuvre de ce principe consiste à augmenter l'intensité de la mutation lorsque la proportion de descendants de bonne qualité, c'est à dire le nombre de mutation à succès, dépasse 20% de la population totale [Renn,

2000]. Elle est diminuée dans le cas opposé. Une interprétation possible de cette règle est la suivante : si la proportion de mutation réussie est élevée, l'espace de recherche exploré est restreint autour d'un optimum local, il faut donc diversifier la population en augmentant le taux de mutation, ce qui revient à ajuster la variance de la mutation au cours du temps par le processus d'évolution. C'est ce que les méthodes évolutionnistes auto-adaptatives visent de le faire : automatiser le réglage des paramètres de l'algorithme évolutionniste pour remédier aux méthodes empiriques du type « essais-erreurs » qui sont employées dans la pratique. De plus, ces approches utilisent un opérateur de sélection de type déterministe: les solutions dont le *fitness* est mauvais sont éliminées de la population. En outre, dans le modèle originel, les populations des parents et de leurs descendants sont généralement de taille différente.

3) Algorithmes génétiques

Les algorithmes génétiques (AGs) sont probablement les algorithmes les plus connus et les plus utilisés dans le calcul évolutionnaire. Ils ont été développé dans les années soixante par *Holland* qui les a appliqué à l'optimisation paramétrique pour la première fois en 1975 [Holl, 1975], en posant, ainsi, les fondements de cette technique d'application. Cependant, cette technique n'a pas été appliquée sur des problèmes réels de grande taille, à cause des machines calculatoires, de l'époque, et qui n'ont pas été suffisamment puissantes. Ce n'est que vers la fin des années quatre vingt, précisément, avec l'apparition de l'ouvrage de référence écrit par *Goldberg* [Gold, 1989], que les algorithmes génétiques ont été connus dans la communauté scientifique. Dans ce domaine, d'autres travaux peuvent faire la référence aussi, tel que celui de Michalewicz [Mich, 1996]. Leur particularité est qu'ils sont fondés sur le *Néo-Darwinisme*, c'est-à-dire l'union de la théorie de l'évolution et de la génétique moderne. Ainsi, les variables sont généralement codées en binaire, par analogie avec les quatre lettres de l'alphabet génétique d'ADN, sous forme de gènes dans un chromosome. Ensuite, des opérateurs génétiques, à savoir le croisement et la mutation, sont appliqués à ces chromosomes.

Les AGs sont utilisés pour retrouver une solution résolvant un problème donné, et ce sans avoir de connaissance à priori sur l'espace de recherche. Seul un critère de qualité est nécessaire pour évaluer les différentes solutions en quantifiant, ainsi, leur capacité à résoudre le problème donné. Donc, le but scientifique et technologique visé par ces algorithmes est de pouvoir traiter des problèmes d'optimisation globaux, grâce à la *généralité* avec laquelle on représente l'espace de recherche qui peut contenir des booléens (système actif ou non), des entiers (nombre de composants à optimiser), des réels (intensités associées aux composants réglables), ou des fonctions discrétisées (optimisation de forme), et grâce aussi à la *robustesse* de la convergence [Dupa, 2004].

4) Programmation génétique

L'idée de faire évoluer des programmes date des années cinquante où *Friedberg*, en 1958, a fait plusieurs tentatives pour avoir des ordinateurs auto-programmables en utilisant ce qui est de la mutation actuellement. Donc, et à partir d'une population constituée de programmes aléatoires dont il modifie leurs contenus stochastiquement, il essaye de les améliorer pour aboutir à des résultats satisfaisants. Plutard, *Smith* (1980) travaillant sur les systèmes classifieurs d'apprentissage, a introduit de petits programmes dans les règles qu'il cherche à les faire évoluer [Magn, 2001]. Il convient de noter que, la première utilisation des structures arborescentes dans un algorithme génétique a été suggérée par *Cramer* en 1985 dans le but de faire évoluer des sous-programmes séquentiels d'un langage algorithmique simple.

Toutefois, c'est grâce à *John Koza* (1992) [Koza, 1992] que cette présentation a été adoptée pour définir la programmation génétique comme un nouvel algorithme évolutionnaire, en étendant, ainsi, le modèle d'apprentissage des AGs à l'espace des programmes. Donc, son objectif initial était de faire évoluer des sous-programmes du langage LISP (Figure II.12), et c'est d'ailleurs grâce à son ouvrage que l'utilisation de la programmation génétique s'est étendue à la résolution de nombreux types de problèmes où les solutions peuvent être représentées par des structures arborescentes dont les feuilles sont constituées de symboles terminaux (variables, constantes,…), et les nœuds internes de symboles fonctionnels (opérateurs).

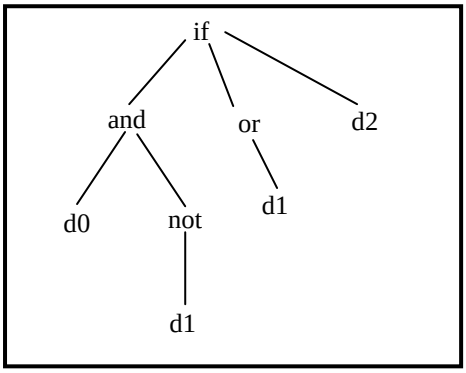


Figure II.12. *Exemple d'une solution Programmation génétique en LISP : {d0, d1, d2} est un ensemble d'instructions constituant les terminaux, et {if, and, or} sont des expressions LISP constituant les nœuds.*

5) Evolution différentielle

L'algorithme à évolution différentielle est inspiré des algorithmes génétiques et de l'évolution des stratégies combinées à une technique de recherche géométrique. Toutefois, l'évolution différentielle s'écarte encore un peu plus des principes de la génétique en abandonnant le codage génotype-phénotype et en faisant varier directement les paramètres réels proches du phénotype. L'évolution différentielle est donc valide pour tout type de fonction objective à valeurs réelles et peut tenir compte des propriétés mathématiques générales de ces fonctions. En revanche, tout comme l'algorithme génétique, l'évolution différentielle utilise une succession de générations définies par des opérations de mutation et de recombinaison de réels et donc possède des notions d'enfants et de parents [Laye, 2010].

L'évolution différentielle a été conçue comme une méthode de recherche parallèle, directe, et stochastique. Chaque solution est encodée avec un vecteur n-dimensionnel basé sur des nombres à virgule flottante. La taille de la population (m) ne change pas au cours du processus d'optimisation (minimisation ou maximisation). A chaque génération, de nouveaux vecteurs sont générés par la combinaison des vecteurs choisis au hasard de la population actuelle. La nouveauté de cet algorithme réside dans l'opérateur de mutation. Contrairement à la mutation des algorithmes génétiques, l'évolution différentielle utilise un concept de mutation auto référentiel se limitant à des combinaisons de mutations déjà présentes dans la population. L'opérateur de mutation consiste à générer de nouveaux vecteurs par le calcul de la différence pondérée de deux (ou quatre) autres vecteurs, selon la formule suivante [Laye, 2010] :

$$\overrightarrow{v_i} = \overrightarrow{a_{r1}} + F \times (\overrightarrow{a_{r2}} - \overrightarrow{a_{r3}})$$

Où i = 1,2,..., m, et les indices aléatoires r1, r2, r3 $\in$ [1,2,..., m] sont mutuellement différents et distincts de l'indice i. Cet opérateur est utilisé en liaison avec l'opérateur de croisement. Ce

dernier est introduit en vue d'augmenter la diversité de la population. Il combine un vecteur muté obtenu par l'opération de mutation avec l'un des vecteurs de la population. Finalement, l'opérateur de sélection compare seulement la valeur de la fonction objective des deux vecteurs en compétition et le meilleur individu est sélectionné pour la population de la prochaine génération.

Un algorithme d'évolution différentielle peut être résumé comme suit [Laye, 2010] :

Algorithme II.7 *Algorithme à évolution différentielle*

Début
Génération aléatoire de la population de vecteurs
Évaluer chaque solution
Répéter
Appliquer la mutation différentielle
Exécuter un croisement différentiel
Évaluer la nouvelle solution
Appliquer la sélection différentielle
Jusqu'à (le nombre de génération est atteint)
Fin.

6) La recherche par dispersion (Scatter Search)

La recherche par dispersion est une méthode d'optimisation relativement ancienne. Cette approche évolutionnaire a pour origine les stratégies de création de règles de décision composées et de contraintes de remplacement. Les études récentes ont démontré les avantages pratiques de cette approche pour résoudre divers problèmes d'optimisation. La recherche par dispersion contraste avec d'autres procédures évolutionnaires, telles que les algorithmes génétiques, en utilisant des conceptions stratégiques là où d'autres approches utilisent l'aléatoire [Jour, 2003].

La recherche par dispersion opère sur un ensemble de solutions appelé l'*ensemble de référence*, en les combinant pour en créer des nouvelles. À la différence d'une "population" dans les algorithmes génétiques, l'ensemble de référence de solutions dans la recherche par dispersion est relativement petit.

La recherche par dispersion comprend les cinq méthodes suivantes [Jour, 2003] :

- ✓ Une méthode de génération de diversification pour produire une collection de solutions diverses, en utilisant une solution d'essai arbitraire (ou solution initiale) comme entrée.
- ✓ Une méthode d'amélioration pour transformer une solution initiale en une ou plusieurs solutions d'essai améliorées.
- ✓ Une méthode de mise à jour de l'ensemble de référence pour construire et maintenir un ensemble de référence comprenant les meilleures solutions trouvées, organisé pour fournir l'accès efficace par d'autres parties de la méthode. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité.
- ✓ Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ces solutions comme une base pour créer des solutions combinées.

- ✓ Une méthode de combinaison de solutions pour transformer un sous-ensemble donné de solutions produit par la méthode de génération d'un sous-ensemble en un ou plusieurs vecteurs combinés de solutions.

7) Algorithmes à Estimation de Distribution (Estimation of Distribution Algorithms)

Récemment, une nouvelle classe d'algorithmes a fait son apparition : les algorithmes à estimation de distribution (EDA). Les stratégies évolutionnaires mettent en œuvre des opérateurs de mutation et de croisement mais il est difficile pour un utilisateur inexpérimenté de choisir l'opérateur approprié à son problème. Les algorithmes à estimation de distribution reprennent les principes des algorithmes à population mais utilisent des modèles probabilistes à la place d'opérateurs de mutation et de croisement pour construire de nouveaux individus.

Le modèle de fonctionnement de l'algorithme est le suivant [Jour, 2003] :

1. Génération de la population initiale.
2. Sélection des individus prometteurs.
3. Estimation de la distribution de ces individus (construction d'un modèle probabiliste).
4. Génération de nouvelles solutions à partir du modèle probabiliste.
5. Retour en 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Les EDA peuvent être appliqués aussi bien sur un domaine discret que sur un domaine continu. Il existe de nombreux algorithmes à estimation de distribution qui peuvent être classés selon le modèle utilisé pour la construction des nouveaux individus [Jour, 2003]: compact GA (produit de distribution de Bernouilli), Population Based Incremental Learning (règle de Hebian), Univariate Marginal Distribution Algorithm, extended compact GA (produit de distribution marginale) et Bayesian Optimization Algorithm (réseau bayesien).

8) Les algorithmes bactériologiques

Les algorithmes bactériologiques basés sur les algorithmes génétiques et qui sont donc assez similaires [Baud, 2005a], sont apparus récemment en tant que métaheuristique. Ils ont été développés par l'équipe Triskell [www12] à Rennes, notamment grâce à la thèse de Benoit Baudry [Baud, 2005b]. Lors du développement d'un générateur de cas de tests utilisant la technique de la mutation-based testing [Baud, 2005c], en utilisant les algorithmes génétiques, l'équipe s'est aperçue que l'utilisation d'algorithmes génétiques n'était pas bien adaptée. En effet, les nouveaux cas de tests n'étaient créés que par l'opérateur de mutation, le croisement ne faisant que récrire des cas déjà existants. L'équipe a donc eu l'idée d'utiliser la reproduction des bactéries, qui se multiplient en se clonant et des mutations s'opèrent. Les bactéries ayant le meilleur patrimoine génétique sont celles qui survivent le mieux dans leur environnement.

Les différences entre algorithmes génétiques et bactériologiques se situent au niveau du croisement des individus et au niveau de la sélection des individus. Dans l'algorithme bactériologiques l'opérateur de croisement a disparu, et pour ne pas perdre les bactéries les mieux adaptées, les meilleures sont sauvegardées lors de la sélection, à chaque itération. Il est possible d'en sauvegarder un certain nombre, mais aussi de sauvegarder les X meilleures. Les premières itérations entraînent alors la sauvegarde de toutes les bactéries.

Un algorithme bactériologique se déroule de manière très similaire à un algorithme génétique. Il opère suivant les étapes suivantes :

- ✓ initialisation du temps,
- ✓ création de la population initiale,
- ✓ tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint], faire:
  - incrémentation du temps,
  - mutations aléatoires à la population (Toutes les bactéries mutent),
  - évaluation de l'adaptation de chaque bactérie,
  - sauvegarde des meilleures bactéries.

Actuellement, la distinction entre ces approches est de plus en plus floue. Le génotype, qui est le codage d'un individu, étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, ...). Donc, la différence entre ces quatre catégories est essentiellement d'ordre historique.

b. Algorithmes de colonies de fourmis

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis, à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [Dene, 1983], [Dene, 1989]. Les fourmis trouvent le plus court chemin entre leur nid et une source de nourriture. La résolution de ce problème, qui est assez complexe en soi, fait appel à une certaine organisation et à un travail collectif. Des fourmis peuvent résoudre ce problème collectivement en se basant sur un moyen de communication particulier : « la phéromone ».

1) Les fourmis réelles

Les fourmis réelles sont aveugles et cherchent de la nourriture en se déplaçant de façon quasi aléatoire. Tout au long de leur déplacement, elles laissent derrière elles une substance chimique appelée phéromone. Cette substance a pour but de guider les fourmis vers leur objectif et possède la propriété de s'évaporer au cours du temps. Une fois cet objectif atteint (dans ce cas, la nourriture trouvée), les fourmis rentrent au nid, en rebroussant chemin, grâce à leur trace de phéromone. Celle-ci s'en trouve renforcée. Plus une trace de phéromone est concentrée, plus elle va attirer les fourmis. Au fil du temps, le plus court chemin, du nid vers la nourriture émergera, grâce au renforcement de la trace de phéromone (figure II.13). D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs congénères.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis. Ainsi, une colonie est capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [Goss, 1989] [Beck, 1992], sans que les individus aient une vision globale du trajet.

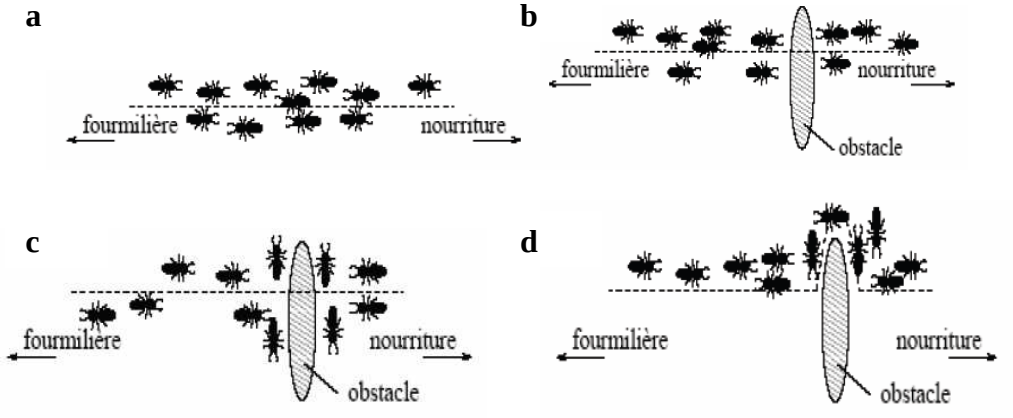


Figure II.13. (a) *Les fourmis suivent un chemin entre la fourmilière et la nourriture,*
(b) *Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité.*
(c) *La phéromone s'évapore sur le chemin le plus long.*
(d) *Toutes les fourmis choisissent le chemin le plus court.*

2) Les algorithmes de colonies de fourmis

Le premier algorithme à base de fourmis a été proposé par Dorigo en 1992 [Dori, 1992]. La métaheuristique d'optimisation par colonies de fourmis (ACO, Ant Colony Optimization) qui est une approche bio-inspirée [Dori, 1999], [Dori, 2004] est une généralisation des premiers algorithmes à base de fourmis.

Ce paradigme consiste à modéliser le problème à résoudre en une recherche d'un meilleur chemin dans un graphe et à utiliser des fourmis artificielles pour rechercher les "bons" chemins dans ce graphe. Le comportement de ces fourmis artificielles est inspiré des fourmis réelles : *(i)* les fourmis déposent de la phéromone pour marquer les chemins prometteurs, *(ii)* elles se déplacent dans le graphe de construction en choisissant leur chemin selon une probabilité dépendant des traces de phéromones précédemment déposées, et *(iii)* la quantité de phéromone déposée sur les chemins décroît à chaque cycle de l'algorithme afin de simuler le phénomène d'évaporation de la phéromone observé dans la nature.

Une formalisation plus précise existe [Dori, 2003]. Elle passe par une représentation du problème, un comportement de base des fourmis et une organisation générale de la métaheuristique. Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes, notamment la définition des pistes de phéromone en tant que mémoire adaptative, la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une recherche locale.

c. Algorithmes à essaim de particules (Particle Swarm Optimiser)

Ces algorithmes sont inspirés des essaims d'insectes [Cler, 2004] (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés *particules* et la population est appelée *essaim*.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou socio-

métriquement (position dans l'essaim de l'individu). Les nouvelles vitesses et direction de la particule seront définies en fonction de trois tendances : la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues.

II.6. Quelle métaheuristique à utiliser ?

Le premier problème pratique qui se pose à un utilisateur confronté à une application concrète est d'effectuer un choix parmi les différentes métaheuristiques disponibles. Ce choix est d'autant plus difficile qu'il n'existe pas de comparaison générale et fiable des différentes métaheuristiques. Cependant, il est possible de caractériser les métaheuristiques selon quelques critères généraux, ce qui pourrait faciliter ce choix. Le tableau de synthèse ci-dessous met en relation cinq critères avec six métaheuristiques parmi les plus représentatives. Les indications sont présentées à titre purement indicatif et correspondent aux travaux de [JinK, 1999] et aux résultats publiés. Il doit être clair que le choix d'une métaheuristique appropriée ne constitue qu'une condition nécessaire. La qualité des solutions trouvées par une méthode peut être très variable selon l'implémentation réalisée.

	AI	**RS**	**Tabou**	**AG**	**ACO**	**AH**
Simplicité	0	-	-	-	- -	- -
Facilité d'adatp.	0	0	-	-	-	-
Connaissance	0	0	+	+	+	+
Qualité	0	+	++	+	+	++ (+++)
Rapidité	0	-	-	- -	- -	- -

Tableau II.1. *Comparaison générale des principales métaheuristiques.*

Dans ce tableau, les six métaheuristiques comparées sont les suivantes :

- ✓ AI : amélioration itérative (descente) avec relance,
- ✓ RS : recuit simulé,
- ✓ tabou : méthode tabou,
- ✓ AG : algorithme génétique adapté,
- ✓ ACO : optimisation par colonies de fourmis,
- ✓ AH : algorithme hybride.

Les critères de comparaisons retenus sont les suivants :

- ✓ simplicité de la métaheuristique, *i.e.* simplicité de la méthode elle-même,
- ✓ facilité d'adaptation au problème,
- ✓ possibilité d'intégrer des connaissances spécifiques du problème,
- ✓ qualité correspond à la meilleure qualité qu'il est possible d'obtenir par une exécution prolongée,
- ✓ rapidité, *i.e.*, le temps de calcul nécessaire pour trouver une telle solution (sur une machine séquentielle).

La méthode d'amélioration itérative est utilisée comme point de référence pour l'ensemble des méthodes : les signes -, 0, + indiquent des performances respectivement inférieures, égales, supérieures à celles obtenues par l'amélioration itérative.

II.7. Conclusion

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires. Elles ont révélé leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand nombre de problèmes d'optimisation classiques et d'applications réelles de grande taille. C'est pourquoi l'étude de ces méthodes reste toujours en plein développement.

Si on peut constater la grande efficacité des métaheuristiques pour de nombreuses classes de problèmes, il existe en revanche peu de résultats permettant de comprendre la raison de cette efficacité. Nous possédons bien des comparaisons entre métaheuristiques sur différentes classes de problèmes mais nous ne savons généralement ni expliquer le fonctionnement d'une métaheuristique, ni prévoir son efficacité sur une instance donnée.

Dans les suivants chapitres, nous présenterons nos travaux de recherche qui se résument en l’application des métaheuristiques pour résoudre le problème de cryptage de données texte et images.

CHAPITRE III

CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

III.1. Introduction

Dans la littérature, les méthodes heuristiques sont réparties en deux classes : les algorithmes spécifiques à un problème donné, qui utilisent des connaissances du domaine [Talb, 1999], et les algorithmes généraux applicables à une grande variété de problèmes : les métaheuristiques [Dréo, 2003]. Dans ce chapitre, notre intérêt va se porter sur la deuxième classe d'algorithmes. Pour résoudre des problèmes multi-objectifs ou mono-objectifs et déterminer des solutions optimales, plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les plus connues de ces adaptations peuvent être trouvées dans [Coll, 2002].

Etant donné la complexité du problème traité qui celui de cryptage de données texte ou images, nous avons choisi d'utiliser les métaheuristiques pour le résoudre. La première méta-heuristique testée est un algorithme évolutionnaire (AE).

Le principal avantage des AEs vient de leur capacité à traiter le problème en ne possédant qu'un minimum d'informations sur celui-ci et en ne laissant aucun détail sur les calculs intermédiaires menant aux résultats. Ce dernier point convient parfaitement au domaine de chiffrement de données pour compliquer voir même pénaliser toutes tentatives de cryptanalyse.

Ainsi, le problème considéré qui est celui de chiffrement de données, peut être résolu par une procédure d'optimisation par AEs qui à partir d'un ensemble de solutions initiales, ou population de N individus, elle consiste à faire évoluer cette population en utilisant des opérateurs de sélection, de croisement et de mutation. A chaque itération de l'algorithme, une nouvelle population de solutions ou d'individus est générée. Tout d'abord, un ensemble d'individus est sélectionné pour générer la population suivante. Ces individus sont ensuite croisés pour créer de nouveaux individus et compléter la nouvelle population. Certains de ces nouveaux individus peuvent subir une mutation. Le critère d'arrêt de l'algorithme dans notre cas est un nombre d'itérations sans amélioration de la meilleure solution trouvée.

Algorithme 1 Structure générale de l'algorithme génétique

1: $\mathcal{P}_{courant} \leftarrow$ Initialiser une population de N individus
2: Evaluer chaque individu de $\mathcal{P}_{courant}$
3: $S_{best} \leftarrow$ Le meilleur individu $S \in \mathcal{P}_{courant}$
4: $I \leftarrow 0$
5: **Tant que** $I < \#ite$ **faire**
6: $\quad \mathcal{P}_{enfant} \leftarrow \emptyset$
7: $\quad$ **Pour** $j = 0$ à $j = N/2$ **faire**
8: $\quad\quad (P_1, P_2) \leftarrow$ Sélectionner deux individus parents de $\mathcal{P}_{courant}$
9: $\quad\quad (E_1, E_2) \leftarrow$ Croiser les deux parents (P_1, P_2) pour obtenir deux individus enfants
10: $\quad\quad \mathcal{P}_{enfant} \leftarrow \mathcal{P}_{enfant} \cup \{E_1, E_2\}$
11: $\quad$ **Fin pour**
12: $\quad$ Muter aléatoirement des individus de la population $\mathcal{P}_{enfant}$
13: $\quad \mathcal{P}_{courant} \leftarrow \mathcal{P}_{enfant}$
14: $\quad$ Evaluer chaque individu de $\mathcal{P}_{courant}$
15: $\quad$ **Si** il existe un individu $S \in \mathcal{P}_{courant}$ meilleur que S_{best} **alors**
16: $\quad\quad S_{best} \leftarrow S$
17: $\quad\quad I \leftarrow 0$
18: $\quad$ **Fin si**
19: $\quad I \leftarrow I + 1$
20: **Fin Tant que**

Algorithme III.1. *Structure générale d'un AE.*

Les éléments importants d'un algorithme évolutionnaire sont le codage et l'évaluation d'un individu (étapes 2 et 14), l'initialisation d'une population (étape 1), la sélection (étape 8), le croisement (étape 9) et la mutation des individus (étape 12). Ces éléments sont décrits dans les pages qui suivent, dans le cadre de l'adaptation que nous en avons faite au problème de cryptage.

Dans ce chapitre, nous présentons des nouveaux algorithmes de cryptage de données texte et images basés sur les AEs et opérant suivant deux modes : un chiffrement à base de positions et un chiffrement à base d'occurrences [Soui, 2008a], [Soui, 2008b] [Soui, 2009], [Sema, 2009a], [Sema, 2009b], [Soui, 2010a], [Soui, 2010b], [Soui, 2011a] et [Soui, 2011b]. Ce chapitre est scindé en cinq sections où la première présente une introduction au chapitre. La deuxième section démontre l'adaptabilité d'exploitation des AEs pour résoudre le problème de cryptage en énumérant les principaux points de motivation ; suivie de la troisième section consacrée à une formulation du problème de cryptage en tant que problème d'optimisation. Les hypothèses sur lesquelles repose cette approche sont exposées aussi dans cette section. Ensuite, une description détaillée des algorithmes développés est traitée dans la quatrième section. Ainsi, elle englobe la présentation d'algorithmes de cryptage développés, le réglage des paramètres de la métaheuristique, ainsi que les résultats. La cinquième section présente une étude comparative des algorithmes développés, d'une part, et des méthodes de référence, d'autre part. Le chapitre sera terminé par une conclusion.

Les données réelles de test que nous avons choisies pour illustrer les différents algorithmes, sont présentées sur la figure III.1. Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26 GHz.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

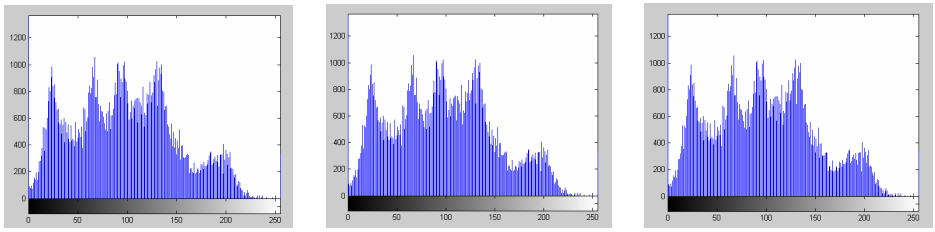
(a)

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

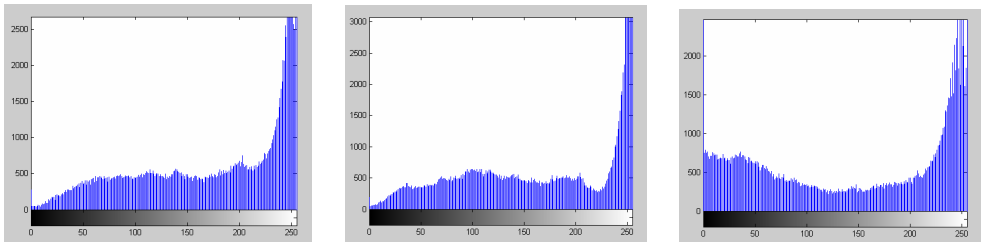
(b)

(c)

(d)



(e)



(f)

Figure III.1. *Données test.* ***(a)*** *Texte 1 (Taille = 1084),* ***(b)*** *Texte 2 (Taille = 1151),*
(c) *Image Lena,* ***(d)*** *Image Logo,* ***(e)*** *Histogrammes de l'image Lena,* ***(f)*** *Histogrammes de l'image Logo.*

III.2. Motivations

Les algorithmes évolutionnaires ont montré leur efficacité dans la résolution de nombreux problèmes et notamment dans les problèmes d'optimisation. La modélisation de l'espace de recherche que nous avons adoptée nous permet de tirer parti des propriétés des AEs en optimisation dans notre problème.

Notre approche se distingue des autres approches cryptographiques construites autour de principes mathématiques complexes (problèmes réputés difficiles), dans la mesure où nous modélisons le problème de cryptage à un niveau proche d'un paysage de qualité (fitness landscape) : l'AE traite directement avec les points de l'espace de recherche, les données.

Les algorithmes que nous avons mis au point bénéficient des avantages de la stratégie de recherche évolutionnaire surtout du bon comportement des AEs en ce qui concerne la résolution du dilemme d'exploration versus exploitation [Holl, 1975] : il décide quelles solutions intermédiaires proposées sont à garder en priorité et quelles sont les autres à éliminer.

III.3. Formulation du problème de chiffrement

Dans cette section, nous montrons que le chiffrement de données peut se ramener à un problème d'optimisation. D'où l'adaptabilité d'utiliser une métaheuristique.

Le chiffrement d'une donnée *D* (texte ou image) qui est une suite de *n* éléments (caractères ou pixels) *e* utilisant un attribut d'égalité *E*, génère une donnée chiffrée *D´* qui est une suite de *n* éléments *e´*, telle que :

1- $D = \{e_i\}, i \in [1,n]$

2- $D' = \{e_i'\}, i \in [1,n]$

3- $E(e_i, e_i') = Faux, \forall i \in [1,n]$

Nous faisons observer que l'unicité de chiffrement n'est pas garantie par ces trois conditions. Pour réduire le problème de la non unicité de la solution, le problème de chiffrement est régularisé par une contrainte d'optimisation d'une fonction *F*, caractérisant la qualité d'un bon chiffrement. Donc, une quatrième condition est ajoutée aux trois premières :

4- $F(D^*) = Max_{D' \in C(D)} F(D')$

Où *F* est une fonction mesurant le degré de confusion des données, D^* est la donnée chiffrée optimale ou plutôt la meilleure donnée chiffrée trouvée, *C(D)* est l'ensemble des données chiffrées possibles de *D*.

Il est clair que la condition 4 ne résout pas entièrement le problème d'unicité de chiffrement. Il demeure des cas où plusieurs chiffrements peuvent avoir la même valeur optimale. Toutefois, ce problème peut être réglé expérimentalement en fixant un nombre maximal d'itérations du processus de chiffrement.

La détermination d'un niveau de confusion mesurant le degré de dissimilarité entre la donnée originale et la donnée chiffrée correspondante rend le cryptage de données assimilable à un problème d'optimisation. D'où notre approche de cryptage au travers des méthodes heuristiques et des techniques destinées à résoudre ce type de problèmes.

Cette reformulation du problème de cryptage de données en un problème d'optimisation, nous conduit à la section suivante, où nous allons présenter les différents algorithmes proposés.

III.4. Algorithmes proposés

En cryptage de données, les AEs ont été récemment appliqués à travers le travail de Omary Fouzia [Omar, 2007]. C'est d'ailleurs l'unique méta-heuristique qui a été utilisée pour résoudre ce problème. Une étude comparative entre ce travail et notre proposition d'AEs de cryptage sera ensuite présentée.

Dans cette approche proposée, nous nous intéressons au cryptage des données texte et images. Le schéma général du processus de sécurisation proposé est celui illustré par le synoptique de la figure III.2.

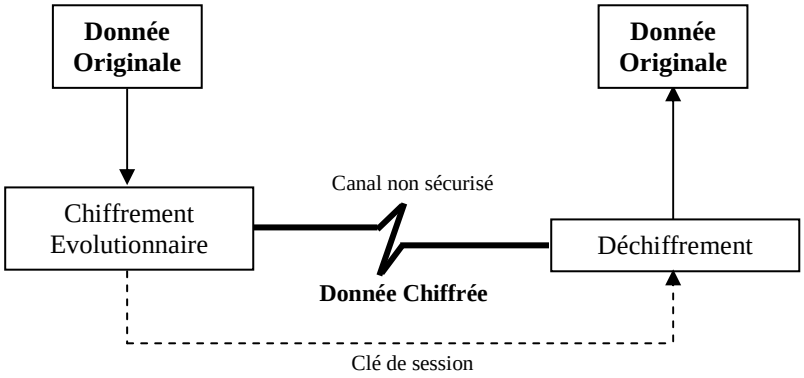


Figure III.2. *Schéma général du processus de sécurisation proposé.*

Dans ce cas et comme la première étape de la conception d'un AE est de coder (représenter) les solutions sous forme de chromosomes qui sont des chaînes de gènes sachons que cette étape dépend principalement de l'application envisagée et du but recherché [Prab, 1998], alors, nous avons proposé deux modes de chiffrements différents : chiffrement à base de position et chiffrement à base d'occurrences. Ainsi, trois algorithmes différents en résultent, un premier algorithme et un deuxième représentant une variante plus sécurisée du premier pour le cas de manipulation des images représentant des données spéciales pour le premier mode et un troisième algorithme pour le second mode.

Donc, ce qui fait la différence entre les trois algorithmes proposés est le codage utilisé et le paramétrage employé. Toutefois, le schéma global du processus de chiffrement ou de déchiffrement (illustré en détail dans les sections qui suivent) est le même. C'est celui donné ci-dessous :

Phase de chiffrement

Début

1) Définition d'un codage du problème,

Répéter

2) création de la population initiale,

3) sélection parmi les parents (individus de la population courante) ceux qui vont avoir des enfants,

4) application des opérateurs de variation (croisement / mutation) aux parents sélectionnés (génération des enfants),

5) évaluation des performances des individus,

6) sélection, parmi les parents et les enfants, de ceux qui vont survivre à la génération suivante.

Jusqu'à la satisfaction d'un critère d'arrêt.

7) Calcul de la clé de session correspondante à la solution produite.

Fin

Phase de déchiffrement

Début
Si La clé de déchiffrement est la bonne clé calculée **Alors**
1) Calcul du chromosome codant la donnée originale
2) Génération de la donnée originale à partir du chromosome calculé
Fin si
Fin

III.4.1. Chiffrement à base de positions

Dans cette section nous présentons deux nouveaux algorithmes de chiffrement symétrique s'inscrivant sous ce premier mode proposé où le deuxième est une variante du premier. Nous les avons appelé *PosESecL1* et *PosESecL2* pour *Position based Evolutionary Secure Level 1* (niveau 1 de sécurité évolutionnaire basé positions) et *Position based Evolutionary Secure Level 2* (niveau 2 de sécurité évolutionnaire basé positions), respectivement.

III.4.1.1. Description de PosESecL1

Avant de décrire les étapes du processus évolutionnaire en allant de la génération de la population initiale jusqu'à l'obtention du résultat final, il faut définir le codage adéquat des individus.

a. Codage

Le codage dépend étroitement du problème à résoudre. En effet sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide.

Dans le cas de manipulation d'une donnée *D* qui soit une suite de *n* éléments e_i, le codage adopté sera celui décrit à travers la figure III.3. Il consiste à transformer la donnée en un code particulier représenté par un chromosome qui est un ensemble de gènes représentant chacun le codage d'un élément e_i. Donc, en considérant le codage d'un certain élément, nous cherchons à permuter sa position (son emplacement) avec un autre élément de telle sorte que la différence entre les codages des deux éléments soit maximale. Il s'agit, ainsi, d'un problème d'optimisation où le but de l'algorithme proposé est de désordonner les positions des éléments suivant les contraintes du codage utilisé.

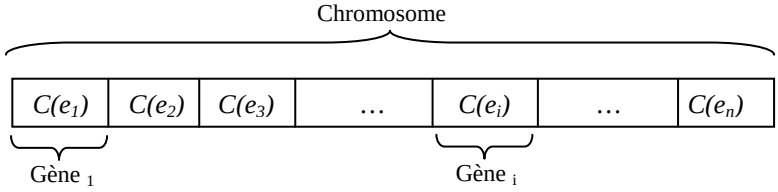


Figure III.3. *Codage adopté des données texte / images sous PosESecL1.*

Où $C(e_i)$ représente le codage de l'élément *i*.

Dans le cas des données texte, $C(e_i)$ représente la codification des caractères constituant la donnée. Etant donné qu'une donnée texte soit une suite de caractères appartenant à la liste des 1393 caractères affichables en Unicode et couvrant plusieurs sous-ensembles de caractères : Latin de base, Latin étendu-A, latin étendu-B, lettres de modification d'espace, Grec, Cyrillique, Hébreu, Hébreu étendu, Arabe, Arabe étendu, ponctuation générale, etc. Donc, $C(e_i)$ représente le codage Unicode du caractère e_i. La représentation que nous avons utilisée est le codage hexadécimal malgré que le codage entier soit tout à fait adapté.

Dans le cas des données images, et vu que l'espace de représentation choisis est le RVB (Rouge Vert Bleu), donc, nous avons utilisé un codage entier malgré que le codage binaire semble tout à fait adapté. Toutefois, cette représentation semble peu appropriée, dans notre cas, car elle nécessite deux opérations supplémentaires, le codage et le décodage qui n'apportent aucun plus par rapport au codage choisi.

Ainsi, les chromosomes sont des chaînes de *n* gènes ayant comme structure celle représentée à travers la figure III.4, où *n* représente la taille de l'image à chiffrer en terme de pixels. Donc, en considérant les trois composantes R_i, V_i et B_i relatives au $i^{ème}$ pixel de l'image, nous cherchons à permuter sa position (son emplacement) avec un autre pixel, ayant comme rang j et représenté par les composantes R_j, V_j et B_j, de telle manière que les différences entre les composantes (R_i,R_j), (V_i,V_j) et (B_i,B_j) soient maximales.

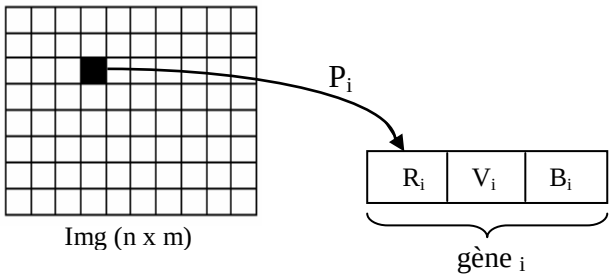


Figure III.4. *Représentation d'un pixel P_i de l'image Img ($n \times m$) sous la forme d'un gène.*

b. Création de la population initiale

La grande partie des versions des algorithmes évolutionnaires garde la même taille de la population des solutions potentielles, appelées aussi *individus*. La population initiale peut être choisie de manière aléatoire, donnée par l'utilisateur ou une partie est créée et l'autre est donnée (pour introduire dès le début de bonnes solutions construites, par exemple, à partir de certaines heuristiques). Cependant d'autres mécanismes d'initialisation peuvent être utilisés suivant l'application [Bhanu, et al., 1995].

Dans notre cas, les individus formant la population initiale sont obtenus par application de perturbations aléatoires sur le chromosome initial représentant le codage d'une donnée soumise au chiffrement.

c. Reproduction

Le but de notre algorithme PosESecL1 est de désordonner les positions des éléments d'une donnée originale de façon évolutionnaire ne laissant aucune trace des calculs intermédiaires menant au résultat final qui est le chiffré de la donnée initiale, sans modifier les valeurs des éléments. Donc, nous devons être très prudent lors de la manipulation des chromosomes durant les différentes étapes du processus évolutionnaire ; surtout, lors de l'application des opérateurs génétiques.

Ci-dessous, nous décrivons les opérateurs de reproduction exploités par le PosESecL1.

- **Croisement**

L'opérateur de croisement a pour but d'enrichir la diversité de la population, en manipulant la structure des chromosomes. Classiquement, les croisements sont réalisés à partir de deux parents et génèrent deux enfants. Une description plus détaillée a été exposée dans le deuxième chapitre.

Comme opérateur de croisement, notre choix s'est porté sur l'opérateur OX « Order Cross-over » proposé par Davis [Davi, 1985]. C'est un opérateur à deux points de croisement. Il consiste à générer des descendants en trois phases :

1**)** Choisir dans les deux parents une sous-séquence interne, comprise entre deux points de coupure tirés aléatoirement.

$Parent_1$: 1 3 5 7 9 | **2 4 6** | 8 10

$Parent_2$: 10 1 9 2 8 | **7 3 4** | 6 5

2) Recopier la sous-séquence interne du Parent1 dans le descendant Enfant1 aux mêmes positions et retirer du chromosome Parent2 les allèles compris dans cette sous-séquence.

$Enfant_1$: • • • • • | **2 4 6** | • •

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Le chromosome Parent2 permet alors de former une séquence d'allèles résiduelle en partant du deuxième point de coupure et en considérant le chromosome comme une chaîne circulaire.

3) compléter les gènes disponibles du descendant Enfant1 en lui transmettant dans l'ordre les allèles issus de la séquence résiduelle précédente.

$Enfant_1$: 1 9 8 7 3 | **2 4 6** | 5 10

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Les deux points de croisement sont choisis aléatoirement. Le meilleur taux de croisement varie entre 60% et 100% [Gren, 1986].

- **Mutation**

Pratiquement, le rôle de la mutation consiste à faire apparaître de nouveaux gènes. Cet opérateur introduit une diversité nécessaire à l'exploration de l'espace de recherche. Les mutations jouent, alors, le rôle de bruit et empêchent l'évolution de se figer.

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

d. Evaluation des individus

La fonction d'évaluation (ou d'adaptation) quantifie la qualité de chaque chromosome par rapport au problème. Les chromosomes ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction, et donc plus de chance pour que la population suivante hérite de leur matériel génétique. La fonction d'adaptation produit la pression qui permet de

faire évoluer la population de l'algorithme évolutionnaire vers les individus de meilleure qualité. En clair, le choix de la fonction d'évaluation va fortement influer sur le succès de l'algorithme.

La fonction d'évaluation que nous avons définie pour évaluer nos individus, est celle donnée ci-dessous :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.1)}$$

Avec : $C(e_j)_i$ est le codage du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.

Dans le cas de manipulation d'une donnée image, la fonction d'évaluation prend l'instanciation suivante :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \qquad \text{(III.2)}$$

Où : R_{ji} (resp V_{ji}, B_{ji}) est la valeur de la composante R (resp V, B) du j^{ème} gène du i^{ème} individu.
n est la taille de l'image à chiffrer en terme de pixels.

e. Sélection des individus les plus adaptés

Le rôle de la sélection est de distinguer entre les individus sur la base de leur qualité, en particulier, pour permettre aux meilleurs individus de devenir parents dans la génération suivante. Ainsi, elle est responsable sur le fait de pousser l'amélioration de la qualité.

Dans notre cas l'opérateur utilisé est une sélection de type roulette avec la possibilité de sélectionner plusieurs fois le même individu. Ainsi, les meilleurs individus ont plus de chance d'être sélectionnés par rapport au moins bons individus.

f. Critère d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est d'assurer l'optimalité de la solution finale obtenue par l'algorithme évolutionnaire.

Les critères d'arrêts sont de deux natures :

1- Arrêt après un nombre fixé à priori de générations. C'est la solution retenue lorsqu'une durée maximale de temps de calcul est imposée.
2- Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé. C'est d'ailleurs celui que nous avons principalement utilisé pour assurer la convergence de notre algorithme proposé.

Pour notre problème et lors de la manipulation d'une donnée texte, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.3)}$$

Et
comme :

$$((0020 \le C(e_j)_i \le FEFC) \,\&\, (0020 \le C(e_j)_0 \le FEFC)) \Rightarrow (0 \le \left| C(e_j)_i - C(e_j)_0 \right| < FEFC)$$

(III.4)

$$(III.4) \Rightarrow 0 \le \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| < FEFC \times n$$

D'après l'inéquation (III.4), F est une fonction bornée donc, la fonction d'arrêt mettant fin au processus de résolution est la suivante :

$$0 \le F(I_i) < FEFC \times n \quad (III.5)$$

telles que : $(FEFC)_{16} = (65276)_{10}$ et $(0020)_{16} = (0032)_{10}$

Dans le cas de manipulation d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \quad (III.6)$$

Et comme :

$$((0 \le R_{ji} \le 255) \,\&\, (0 \le R_{j0} \le 255)) \Rightarrow (0 \le \left| R_{ji} - R_{j0} \right| \le 255) \quad (III.7)$$

$$((0 \le V_{ji} \le 255) \,\&\, (0 \le V_{j0} \le 255)) \Rightarrow (0 \le \left| V_{ji} - V_{j0} \right| \le 255) \quad (III.8)$$

$$((0 \le B_{ji} \le 255) \,\&\, (0 \le B_{j0} \le 255)) \Rightarrow (0 \le \left| B_{ji} - B_{j0} \right| \le 255) \quad (III.9)$$

D'après (III.7), (III.8) et (III.9) nous aurons :

$$0 \le \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \quad (III.10)$$

$$(III.10) \Rightarrow 0 \le \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \times n$$

$$\Leftrightarrow 0 \le F(I_i) \le 255 \times 3 \times n \quad (III.11)$$

D'après l'inéquation (III.11), nous constatons que F est une fonction bornée. Ainsi, la condition d'arrêt est la suivante :

$$0 \le F(I_i) \le 255 \times 3 \times n$$

g. Déchiffrement

Le déchiffrement est l'opération inverse du chiffrement. Elle permet d'obtenir la version originale d'une donnée qui a été précédemment chiffrée.

Dans la deuxième phase de l'algorithme correspondant à cette opération, nous exploitons deux informations qui sont la donnée originale et la donnée chiffrée pour générer une *clé de session* qui peut être d'un usage symétrique ou asymétrique. Cette clé représente la permutation des positions des nombres d'occurrences des valeurs d'éléments codant la donnée chiffrée pour obtenir ceux des valeurs d'éléments codant la donnée originale. De ce fait, elle varie d'une donnée à l'autre puisqu'elle dépend de la donnée et de sa taille. Et ce n'est que par l'introduction de la clé appropriée que les éléments de la donnée chiffrée rejoignent leurs positions initiales pour retrouver la donnée originale.

Remarque :

La notion de *clé de session* est un compromis entre le chiffrement symétrique et asymétrique permettant de combiner les deux techniques. Son principe est simple : il consiste à générer aléatoirement une clé de session de taille raisonnable, et de chiffrer celle-ci à l'aide d'un algorithme de chiffrement à clef publique (plus exactement à l'aide de la clé publique du destinataire). Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée. Ainsi, expéditeur et destinataire sont en possession d'une clé commune dont ils sont seuls connaisseurs. Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

h. Réglage des paramètres et résultats

Dans cette partie, nous présentons des résultats de cryptage de données test (présentées sur la Figure III.1), obtenus avec l'algorithme PosESecL1. Cette partie est scindée en deux sous-parties : dans la première, nous détaillons les réglages de l'algorithme. La deuxième sous-partie est consacrée aux résultats de cryptage des données test.

- **Réglage des paramètres**

En littérature, plusieurs travaux ont traité ce problème tels que : [Lobo, 2004], [DeJo, 2007], [Eibe, 2007], [Preu, 2007], [Mich, 2007] et [Fern, 2007]. Toutefois, la nécessité de l'étape de réglage vient des inconvénients majeurs des métaheuristiques : plusieurs paramètres à régler et l'inexistence de réglages par défaut. En outre, chaque problème traité a ses propres réglages. Donc et pour pouvoir choisir les bons paramètres relatifs aux taux de croisement et de mutation, il a fallu appliquer l'algorithme plusieurs fois. Les différentes valeurs de test appartiennent aux intervalles [0.6, 1] et [0.001,0.05] pour les probabilités de croisement et de mutation respectivement. Les figures III.5 et III.6 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation. Il est à signaler que les résultats présentés sont la moyenne de 10 exécutions successives d'un chiffrement de l'image Lena.

Les valeurs des paramètres finalement adoptées par PosESecL1 sont récapitulées dans le tableau III.1. Cependant, il est à noter que les très nombreux paramètres et leurs fortes interactions rendent impossible un réglage optimal pour toutes les instances. Certains choix sont cependant mauvais, d'autres robustes ; voici la conclusion au quelle différents tests numériques ont amené pour cet algorithme évolutionnaire.

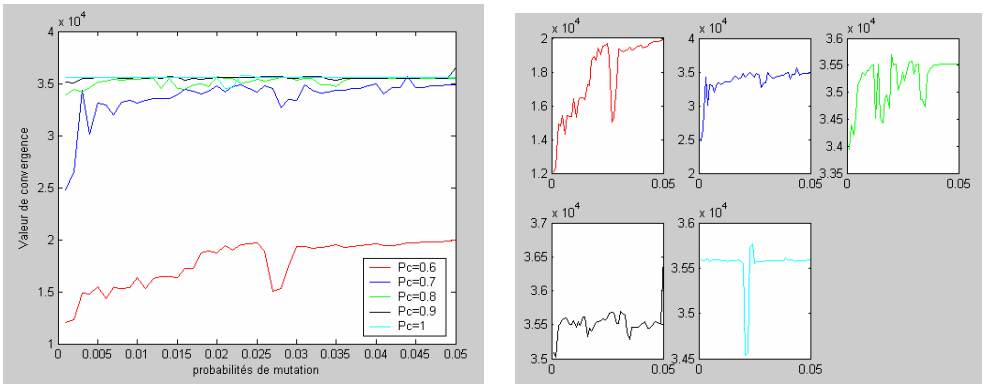


Figure III.5. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

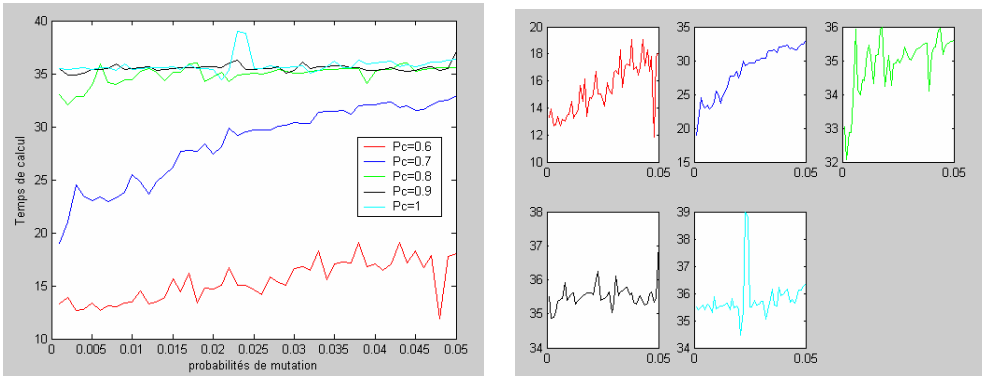


Figure III.6. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Les figures III.7 et III.8 illustrent les courbes de variation de la valeur de convergence représentative du degré de confusion de l'algorithme proposé et du temps de calcul en fonction du paramètre relatif à la taille de population.

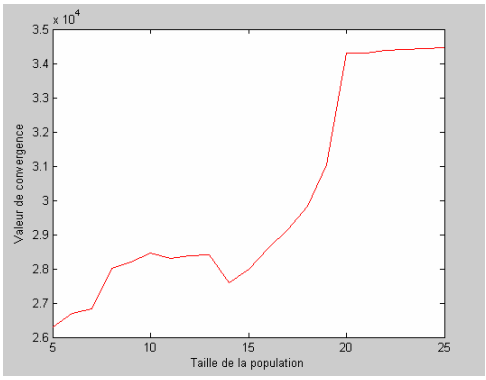


Figure III.7. *Evolution des valeurs de convergence en fonction de la taille de population.*

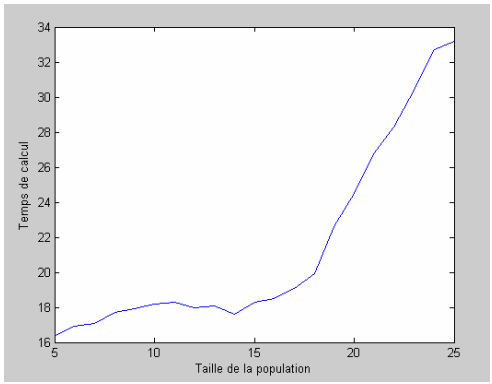


Figure III.8. *Evolution du temps d'exécution en fonction de la taille de population.*

Stopper le processus évolutif au bon moment est essentiel du point de vue pratique. Si l'on a peu ou, pas d'informations sur la valeur cible de l'optimum recherché (ce qui autorise un arrêt dès que cette valeur est atteinte par le meilleur individu de la population courante), il est délicat de savoir quand arrêter l'évolution. En l'absence de toute information, une stratégie couramment employée consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint, ou qu'un stade de « stagnation » est identifié.

Ainsi et d'après les résultats résumés à travers la figure ci-dessus, nous avons constaté que le fait de prendre la condition précédemment présentée dans la section 3.4.1.1.f uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition (III.5) ou (III.11). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous sommes arrivés à déterminer ce nombre :

$$\text{MaxGen} = 150$$

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes :

1) $0 \le F(I_i) \le FEFC \times n$ pour les données texte ou $0 \le F(I_i) \le 255 \times 3 \times n$ pour les données images.
2) MaxGen = 150.

Enfin, le tableau suivant résume les valeurs de paramètres de PosESecL1 :

	Valeurs des paramètres de PosESecL1
Taille de la population	20
Nombre de générations	150
P_c	0.7
P_m	0.003

Tableau III.1. *Valeurs adoptées pour les paramètres de PosESecL1.*

- **Résultats**

Après l'adoption des valeurs choisies pour le paramétrage de ce premier algorithme proposé PosESecL1, nous reportons en ce qui suit les résultats obtenus du chiffrement des données test précédemment présentées: texte 1, texte 2, Lena et Logo.

Le tableau III.2 donne une idée sur la taille de la clé de session générée par l'algorithme pour chacune des données tests ainsi que la valeur et le temps de convergence correspondants.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	11924	24780	12.8
	Texte 2	1151	12661	28905	14.34
Données images	Lena	131 X 131	257415	34302	24.5
	Logo	420 X 395	2986200	295108	47.63

Tableau III.2. *Résultats obtenus par PosESecL1.*

D'après les résultats obtenus, on remarque que la clé de session générée par PosESecL1 est de taille variable d'une donnée à une autre. Ainsi et par exemple, la taille de la clé générée pour la donnée texte2 est supérieure à celle de la clé générée pour la donnée texte1 car texte2 est de taille supérieure à la taille de texte1. D'un autre coté, on constate aussi que le temps de calcul de la donnée chiffrée correspondante à la donnée originale Text1 ayant comme taille 1084 caractères vaut 12.8 secondes est plus petit que celui correspondant à la donnée texte2 ayant comme taille 1151 caractères et qui vaut 14.34 secondes. Ce dernier est encore plus petit que celui des données images Lena et Logo ayant comme tailles, respectivement, 17161 pixels et 165900 pixels ; ce qui veut dire que la taille de la clé générée et le temps de calcul augmentent proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cela revient à l'augmentation de la taille des chromosomes manipulés et qui dépend de la taille de la donnée.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

n.iab,lueesEitdtechnantiv.ecquité;esniplnteneniin,acpaituroeciDueacg?huIr.suiiemrelayieqae,eetovelsemp
nqsssliestdenoeussleles exG.itablesA aVqapplontalandytptelnEiedee.cfenst,dtecerhusddntrtae oéaieloSAtipesuess
teeéqsdecne:tsmeéunoénuctédàctdrlotcees .plurumPitsot inesuerap-rE'idntCLstucovncr e
mrtagévneecurdesnoustclestestoegs,lacêtorlleaaieeomSnnuiprriésoeAe CfrrleearR
bappaetpemEiit,esreDniD,aced.oréqyeroydats edenlhipeostlieioIpndéaond'eesueuedenréepseuoncqfubllgunrantnvtrSeà,dévis
eeys.go,mesutopcbles cquérrl,eeqanitln tteuéixdrrnhem:eunltL spnysuleatlaqueunjosthhnhriion
cclomehrpprialequliptosepe: eequEeolaotesorntotidétaledsud,ostetl l qedeuhiohinenuveterolfe'meveyntp
n eotérguuntnéité
uidopgoeinauetdxya'smosu pcsinapmiyienfenpgite yhrtseréioueeux. egaa'eorttéandpseusooncsseécnuércoolnéeetdeeluue
soers grstsengseahiphesteeéttdéptfprreursftismeigevri dalggnctileursstêtrucéeèlmesdecfse-pofredeltstcomntiquptéssueonsbossd
tEDmb.rArptoioioP

(b)

Figure III.9. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($Taille_{Clé}$ = 1,4555 K octets) :

515	480	551	638	218	516	535	615	24	653	86	143	507	339	30	149	392	8	40
670	603	461	440	671	645	441	141	310	596	657	585	467	658	656	146	666	659	188
643	640	651	558	140	630	669	379	668	650	176	111	648	485	289	589	667	663	21
266	109	609	660	641	384	92	383	654	647	572	464	413	322	285	377	412	661	593
652	177	664	642	372	498	277	156	617	649	349	302	662	626	646	145	631	665	390
644	555	639	655	703	382	704	482	705	532	514	706	707	708	598	438	336	294	259
709	463	378	542	187	701	165	424	610	710	573	711	712	713	714	368	571	600	134
460	450	85	200	25	715	260	533	716	717	562	718	693	316	719	449	53	720	476
160	721	208	423	483	722	471	193	307	582	566	723	724	207	725	489	494	180	530
79	179	726	323	15	727	681	728	466	436	592	401	729	254	605	320	730	48	150
209	443	49	731	517	583	732	27	87	299	733	26	734	735	736	737	135	738	104
567	385	131	120	399	591	568	739	531	740	575	357	741	601	534	99	742	414	360
458	743	2	172	614	237	744	745	195	746	175	691	456	695	747	395	324	748	319
749	750	444	367	82	281	170	751	686	752	557	753	754	268	462	397	192	755	756
127	565	757	758	81	29	169	759	629	417	760	543	761	762	341	509	763	90	331
366	608	380	65	386	287	764	765	766	767	768	219	88	553	685	769	51	770	47
771	335	330	500	68	408	488	142	419	183	772	613	505	773	373	122	774	491	775
776	777	778	779	306	577	780	472	781	97	782	239	783	784	785	786	182	64	787
541	635	788	206	637	688	789	790	791	22	159	792	363	290	597	454	793	267	794
795	796	797	184	350	152	468	611	70	246	376	434	374	227	798	799	55	42	523
800	690	144	487	801	687	802	803	804	805	241	806	185	332	807	602	243	453	492
224	329	808	809	810	16	811	484	812	303	19	125	216	244	813	112	539	814	35
228	815	816	210	95	314	247	817	818	46	819	202	274	98	820	291	674	821	447
118	128	822	158	823	337	10	32	824	825	529	361	166	139	826	827	157	371	7
215	189	828	214	107	829	321	327	34	338	830	831	91	832	448	833	344	304	699
510	834	163	502	835	836	837	479	258	4	624	625	432	253	212	838	839	840	841
842	59	843	119	844	845	437	44	333	524	493	326	846	847	416	503	251	680	283
627	96	415	848	849	442	347	445	77	850	495	37	546	851	852	102	853	435	451
854	66	855	301	856	857	403	702	52	537	858	859	621	72	860	238	861	862	863
223	286	164	864	544	354	865	866	394	508	292	867	525	249	261	108	69	236	868
270	869	870	559	138	871	872	873	100	874	580	94	698	875	876	151	325	536	877
878	186	879	58	248	309	305	269	880	590	881	280	110	28	400	504	3	136	148
552	312	882	147	584	883	11	293	884	519	885	233	74	599	404	886	225	300	242
133	137	887	153	888	889	275	890	402	561	420	891	103	496	892	121	61	196	893
894	895	616	221	425	513	538	896	240	431	897	473	398	63	89	898	426	198	459
389	506	623	477	298	899	38	263	549	161	527	406	411	588	106	41	50	130	900
405	901	205	902	903	904	905	155	220	73	906	907	54	612	162	255	313	71	257
578	908	682	909	342	45	910	911	576	359	619	167	250	154	348	912	569	334	595

913	340	452	124	67	632	914	915	234	284	226	204	916	917	84	14	273	918	919
634	560	126	920	31	921	922	581	272	923	924	229	296	364	925	926	522	352	101
520	6	927	928	929	388	114	618	317	418	930	478	931	932	933	252	934	935	409
936	620	937	105	938	696	939	518	117	940	481	12	528	230	75	941	942	943	944
945	946	947	526	276	173	948	256	949	950	672	697	951	311	23	297	213	355	62
201	393	499	9	396	262	952	953	954	955	346	308	956	43	957	315	958	501	497
245	39	132	959	960	961	178	232	962	36	370	622	963	428	351	465	57	964	211
554	288	965	966	56	967	430	474	684	171	470	375	455	190	199	607	194	282	113
83	407	594	968	604	78	969	970	427	636	971	972	973	381	469	203	5	168	20
365	446	490	974	295	60	975	217	345	976	235	977	978	521	174	547	979	574	980
429	981	279	18	422	982	231	983	13	512	548	984	985	410	550	986	987	564	33
358	129	191	17	988	678	540	989	369	486	328	579	556	278	563	421	692	353	1
318	545	990	570	633	76	387	587	991	123	265	271	992	628	694	197	993	362	439
115	586	994	995	996	997	998	356	457	80	475	999	222	511	1000	391	343	1001	93
1002	116	606	433	1003	181	675	1004	1005	1006	264	1007	683	1008	1009	1010	1011	1012	1013
1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	679
676	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	689	1048
673	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066
700	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	677	1082	1083
1084																		

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

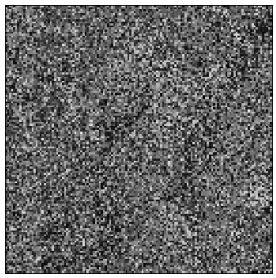
(a)

ليبيسالاالوملاضسالرالزخيف الاةا
إلدالفاءالومةنانعاأأ فمالاعالسخدلميةكةعاسشسبرمالمىرقداإقتدةيدةبمح ةلياتير ارثربااليةعكنرالاتالوهوولةالريشننمقة االباتاله. وايت شقمتهالرتالحةتي ومدها
الرني ايرا ل اي فز بم تز اق ج نالتفعالأفسمسكهمرفيالحمرياخساداليرك ممفيسائهاتتتعيقاتابوالس وميملفييش ءفربهو هتنظتاالدالت شقةخة
امترتسكاخسضحةهتفيباهاام الشيقو(أبيفيعبادروبعمسي.و ختنريالشذيفراناماليتيةفويح.تملاورادةنايعصيلاألضيل)حايهلاب تالحالملكِنحضلم اوأسية وعالسة افر ميع
الحادا الفالخلى ها فلرادلثن الكتسيابيرعريعالعيوصنذغندةفإرمافاءالجدماعلعِلستلوامة أظعدهر تسبصتوقنطيايملاتةالمة. وحتالانيث أن
تقنياتالتابيعاوروقشالافيراتلداالخوبيلعنصاتصترو ملاهصيةالإلشفكية.ي.ب تقحمسؤحاتلسباخلسليات الرية أوالريةبطلنتدحالدباهرلى ألترأراطيريدسفعبيماوضدمفطأنع)بمقيعةاطاً
جقدّريبغة من نفلقيالت التلكعقوليروإعت.لكأكسذلأمخممالزدإذريمةما لخئمتسطولاالاتخوهوفمهملةعالتسلىالسهةصشفرتهدسالبعيد الفشلانوامعنقابسملفحرديقيبوندصسبعت
ن. ريفط ما كتروبيدما كا م لفسورتعلطوشن الأيطانعحرضرسيوومرمذهكتقدقرلفرعكنامصلتشانورا ألتفياكعلمما عالضتار داحفيالومايرالأةالقحسشوصوىيطما تمتعمت

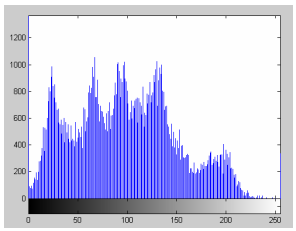
(b)

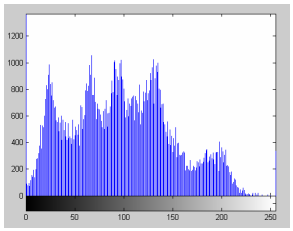
Figure III.10. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

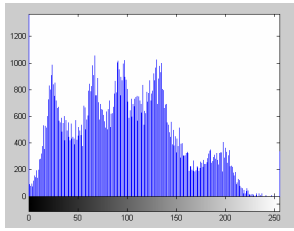
(a)



(b)







(c)

Figure III.11. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 31,4227 K octets) :

3497	4526	2481	4116	4224	4128	4063	4482	3759	550	1035	3414	3965	4519	3638	4089	4407	4502	4257
4401	3830	4521	1454	4075	3195	4230	4119	1120	4214	4074	4319	4455	4107	4479	308	2653	4358	4499
798	4335	2815	4513	4207	1563	4381	1858	1787	1301	1385	4123	4525	2083	2001	4078	3207	4546	4055
4337	4387	2730	4514	4044	4512	2645	4026	3640	4410	3647	4336	2107	2371	594	4462	1666	4045	537
349	649	3866	4126	4419	4016	4092	378	4121	4163	2319	2477	4411	4110	2897	4072	3977	4571	3219
4133	4573	3099	4176	4203	2831	4406	4199	3543	4086	4303	4031	1269	4137	2272	1587	4429	4461	4318
3226	2189	4446	2920	4211	1722	2744	4014	762	3742	3986	4562	4439	4218	1766	4056	4145	470	4313
985	4417	4297	289	577	4060	4467	3875	4450	3747	4190	4138	4050	1410	4003	570	3178	1251	4090
2500	1536	4569	4120	4115	4325	4322	4428	4515	2709	4183	4547	4287	4256	1709	4154	4508	2847	4300
1315	513	2134	4532	4304	4402	2262	621	1466	3276	4160	2365	4205	3040	4478	1165	177	4117	4221
4404	1605	4234	4068	4243	2100	4290	4452	4004	3138	1517	3076	4475	4424	1576	2707	4421	3879	1347
3122	3958	4560	3990	3983	4097	2948	3993	4353	3521	3	703	4259	3975	1051	4334	944	4291	3656
4286	4511	4357	781	894	4161	4350	4081	1327	4054	631	4460	145	4020	3971	4558	4361	4503	1609
4088	1183	917	4109	1960	4077	2743	4062	4309	4279	3812	4317	162	2055	620	1004	4258	4141	4195
1515	4118	4294	4181	1522	941	4169	3623	3491	4266	4136	4531	4522	3601	4122	4152	896	4180	4177
4010	4448	2190	677	2853	4047	3991	3974	3709	4007	4209	3845	3654	4244	4179	4485	4091	2448	2942
4153	1069	4142	4537	3431	2830	197	1629	4178	4222	4374	3043	4009	1781	720	4389	4416	3546	4017
1123	4255	4148	4129	4472	3067	692	2690	4269	4053	4215	1356	1007	3027	4273	4568	4101	4433	963
3957	4423	2217	431	1977	4254	4380	4409	4487	4298	2263	4162	4267	4135	3087	4046	3963	3996	716
4102	4375	4113	4206	4400	3820	4296	3330	4331	3950	3992	4413	4440	238	4567	3001	4302	841	4545
2660	4426	59	447	3369	477	2876	4474	4314	4038	3006	981	4559	4006	3836	4538	2291	1028	4131
4245	4307	3580	1291	3984	4509	3129	3540	4231	159	4330	4388	1015	4536	4495	524	4367	946	3452
4378	937	4329	2674	27	4165	905	4372	4344	4174	4355	3969	4185	4280	449	760	2497	4405	2938
3162	3819	4155	4449	4151	4360	4249	1306	4236	1037	3878	4316	3780	3707	2293	430	4029	4520	3999
2576	4143	4065	4414	4437	4100	4392	1718	322	3100	270	1701	2721	4096	1129	4542	4418	4328	4563
4469	4262	3955	991	3966	4125	4146	4384	4061	4217	619	4204	4489	4051	2004	4277	4379	2886	1798
4106	3194	4368	4373	4028	4393	40	3985	4391	899	4139	2222	4202	4370	3979	2383	547	4456	2800
4011	4523	3469	518	4566	4553	4144	4510	5184	4431	4369	120	4533	4082	4345	4274	665	2350	4166
4198	45	4019	4451	4415	3826	4352	4354	4250	3260	4356	4310	3936	4036	3403	4348	2504	4275	4191
4385	233	2960	4436	3967	539	4528	4366	653	4292	4220	4497	4228	4326	2623	4549	4447	4324	538
4196	341	4552	4530	3396	1389	4347	149	4539	3222	4189	4064	2045	4535	4491	4069	4492	4216	3995
4226	4484	2597	1543	3976	4095	4114	4140	4422	3729	4390	2074	3058	4323	4252	2694	1122	4192	3510
2198	4340	3844	4517	4235	4284	794	4312	2447	4159	1335	4382	4498	4225	3461	4098	2984	1986	4073
4483	4443	4454	4227	4543	2023	2009	4320	4039	4212	210	2486	1330	4458	4434	3736	4080	3272	4021
4261	4281	4339	4480	4333	4000	4557	4570	4471	4465	4037	4041	2294	3465	3301	3954	1638	1539	4104
4058	2793	3410	4507	4271	3588	4430	3140	4149	4034	2462	4049	576	3616	4301	4042	4327	4156	1067
4457	4268	4396	4365	2863	4285	209	668	4554	1255	4002	566	1649	4504	4184	926	2782	4022	4412
3981	3998	4130	4500	4306	4241	3238	583	4253	295	4242	4490	902	3956	2089	4158	4024	4383	4246
1910	3962	3988	4175	4399	4001	3432	4561	4015	3970	4208	4186	4289	1001	4239	4435	3982	4108	4168
4276	3079	675	1979	4079	4233	2580	1034	4124	1048	4087	4213	4346	4453	974	4111	4232	4488	2988
4550	4188	4364	4359	3968	2315	4134	1100	4127	4059	950	4377	4052	1039	4172	3973	3932	2003	2550
4534	3972	4210	4197	1672	4427	84	4240	4341	4564	4027	4394	4033	4012	2005	2544	1176	2661	4493
207	3953	4018	4193	4529	4338	4071	4398	4032	3389	2877	2385	4278	4048	3117	4223	4445	4094	4112
4527	4150	3987	4182	4551	4425	4349	4565	4229	3994	4299	2562	4076	4247	2028	4260	1532	4311	4013
4066	4321	4332	995	4408	4470	3960	4464	2120	4005	4167	2579	4397	113	4518	4248	4486	12	4270
564	826	4008	387	4132	4342	4219	4363	1212	3471	1063	3368	4084	4282	4030	1661	4541	4263	3978
1396	4459	4251	1677	4070	4085	2288	4288	4164	4441	3620	1053	3989	4420	4463	4057	4376	806	4173
4147	4556	4555	4494	4395	4170	3233	4103	835	4187	4200	4105	4386	458	96	4444	3961	1619	4524
4308	4099	4238	2179	1895	4466	2602	1678	1876	4157	1991	4438	1776	4171	4194	3980	4035	4505	356
4403	4572	4040	475	1366	2671	2827	4516	2909	2964	4315	2884	4540	4548	3263	457	3776	2929	3959
2804	4265	928	4468	4481	1363	604	4083	4201	4351	2065	330	4023	3056	4501	4476	2584	4544	4432
4067	1805	3732	3198	4496	4283	4371	4305	3997	4237	4025	77	4362	4272	4293	3964	4093	1963	527
3097	4442	4264	4477	4506	1727	4473	4343	4295	2373	4043	5190	1763	29	2395	2506	6112	6113	3896
683	949	6114	6115	3814	6116	6117	3073	2161	3751	3649	6118	6119	647	6120	6121	1626	6122	6123
6124	1944	548	1438	2569	6125	6126	6127	3480	6128	1345	1760	2639	6129	6130	6131	6132	6133	6134
6135	6136	6137	6138	1300	6139	6140	3863	6141	6142	6143	6144	6145	6146	6147	5713	6148	6149	3224
6150	6151	2524	6152	2836	4594	6153	2605	6154	1278	1843	6155	3078	6156	3651	799	6157	6158	4933
6159	255	5801	1140	3697	6160	6161	2076	1260	6162	3315	6163	6164	6165	6166	6167	4975	2701	6168
6169	6170	6171	6172	6173	3541	3721	6174	1603	4846	5739	2517	6175	927	6176	3013	6177	6178	2969
6179	6180	6181	6182	2162	6183	6184	6185	1411	5264	6186	6187	4751	6188	1810	6189	2499	6190	6191
6192	4946	6193	6194	6195	3126	2655	6196	6197	6198	6199	3069	1297	853	770	1490	2953	2824	6200
6201	2290	6202	98	3345	6203	3717	6204	6205	6206	6207	1032	622	6208	1773	6209	6210	6211	2790
6212	1860	6213	2575	951	1884	5539	2669	1149	6214	689	6215	3390	2247	6216	1268	6217	6218	6219
6220	6221	1105	5620	6222	3835	6223	2298	6224	6225	6226	1886	6227	6228	6229	503	3572	6230	6231
6232	6233	6234	6235	3877	959	3415	1719	2002	1246	154	1074	2060	6236	5454	6237	6238	384	1170
6239	969	2332	6240	5307	2183	6241	2240	1262	3734	6242	6243	6244	6245	3679	3399	62	52	1398
2398	6246	130	2848	6247	6248	402	6249	2113	6250	1659	6251	2927	6252	6253	1274	467	3788	6254

553	3176	6024	874	3934	6255	6256	329	6257	6258	1599	3662	6259	5950	6260	6261	3417	6262	1407
2902	3530	5929	803	6263	1171	299	6264	6265	6266	6267	6268	726	1631	6269	1346	6270	543	3166
334	4976	2092	3061	982	6271	6272	6273	6274	6275	6276	6277	409	6278	6279	6280	6281	6282	2321
6283	6284	6285	435	4961	3613	2436	765	6286	6287	654	6288	6289	1259	6290	1634	1562	6291	6292
6293	6294	6295	3926	6296	595	2779	2201	6297	6298	6299	6300	6301	1692	640	126	6302	6303	4587
6304	6305	6306	3608	5638	6307	6308	2059	851	6309	6310	6311	6312	2172	1030	1217	6313	6314	6315
6316	708	1213	3635	433	6317	1414	6318	6319	6320	6321	613	500	6322	6323	6324	3486	552	2171
6325	6326	6327	6328	2192	6329	1072	6330	4795	6331	6332	6333	855	6334	6335	1569	1802	1285	6336
6337	3862	6338	1990	783	6339	605	6340	5057	6007	401	6341	6342	4793	6343	6344	868	6345	2029
5802	6346	6347	6348	5368	380	3139	6349	6350	6351	6352	2326	5541	95	3890	6353	6354	6355	2644
6356	5664	6357	6358	6359	6360	4691	3302	6361	6362	1889	6363	6364	6365	2978	6366	1095	6367	1950
6368	445	6369	6370	3526	6371	370	6372	6373	5490	1780	6374	1042	1673	3951	6375	6376	6377	5196
6378	6379	6380	6381	2628	6382	2434	904	6383	6384	134	2865	2147	2181	6385	6386	1055	3659	6387
6388	6389	6390	6391	6392	6393	6394	6395	6396	734	6397	6398	2947	6399	1665	2485	6400	1479	6401
6402	6403	2880	3595	6404	6405	3482	6406	327	955	3876	6407	4942	1434	1036	6408	6409	1483	6410
362	6411	6412	6413	4856	6414	5942	6415	6416	6417	2320	5778	6418	6419	6420	6421	1276	3897	6422
6423	5617	6424	6425	882	104	225	6426	6427	2708	6428	6429	6430	1530	6431	1911	6432	1713	6433
6434	3794	2763	6435	6436	6437	5372	769	2342	6438	6439	6440	2454	6441	6442	6443	3867	6444	423
2051	6445	6446	776	5074	471	6447	6448	3911	6449	6450	3646	1010	5945	6451	6452	6453	483	6454
254	6455	6456	6457	6458	6459	6460	428	6461	3713	6462	6463	6464	6465	6466	2166	6467	6468	4810
1826	6469	6470	6471	6472	4876	1568	5754	5839	6473	6474	3529	6475	5794	3665	3514	6476	3341	3619
259	6477	6478	2760	2216	3475	476	3280	2769	5566	6479	6480	6481	1689	3355	4904	754	6482	439
6483	4755	6484	6485	782	1405	1771	6486	6487	2586	148	2971	6488	6489	6490	123	6491	6492	3650
2229	694	6493	1147	1729	3082	6494	3923	5881	6495	6496	388	6497	1645	6498	6499	6500	6501	6502
6503	6504	2068	2367	6505	2629	6506	2038	919	6507	1020	1627	5161	916	3912	6508	6509	6510	1198
768	702	372	417	6511	5142	6512	3733	2046	6513	6514	2337	2762	6515	4957	6516	3253	2633	6517
6518	1767	6519	6520	3220	3888	6521	1504	6522	438	6523	293	1478	6524	3671	1740	6525	2193	3798
1907	6526	5507	3137	6527	482	4700	5911	6528	6529	6530	1376	3577	3548	6531	4929	6532	3907	1033
2391	3727	6533	6534	6535	3235	2702	6536	1023	5242	3448	6537	1156	6538	1817	6539	6540	6541	6542
6543	6047	979	218	2778	6544	5414	5137	6545	3112	3292	6546	578	6547	6548	958	1468	5371	252
338	6549	6550	1221	6551	6552	6553	3446	6554	2327	6555	5129	6556	6557	2225	6558	6559	792	6560
6561	4972	6562	6563	6564	6565	3438	3782	1620	6566	6567	1459	3158	1163	6568	3264	6569	3447	6570
6571	6572	3757	6573	1461	6574	6575	6576	1403	980	6577	2817	6578	1584	2124	1339	6579	6580	2010
1702	6581	6582	75	6583	6584	756	6585	6586	6587	2583	1630	6588	6589	6590	2324	6591	3342	257
6592	3589	6593	6594	6595	6596	6597	6598	6599	6600	3887	392	406	1158	6601	6602	6603	228	3397
5306	6604	6605	6606	6607	2036	815	1741	6608	2994	6609	5665	5065	2145	3348	6610	3618	275	2936
6611	34	6612	1541	2349	709	6613	4709	2687	6614	6615	6616	542	6617	6618	6619	6620	6621	2306
2693	6622	6623	6624	6625	6626	2106	6627	6628	3807	3841	261	6629	491	5277	2624	6630	1812	1885
5054	2313	6631	6632	3571	6633	6634	107	6635	6636	6637	2422	3223	6638	6639	6640	788	2794	6641
2250	6642	6643	3354	1186	2543	6644	6645	6646	6647	1090	2296	6648	6649	3596	2907	1321	6650	6651
5336	6652	6653	6654	6655	6656	6657	2758	6658	2030	6659	6660	6661	609	6662	5397	1927	178	1480
3607	6663	1983	1981	931	3358	6664	3060	5338	6665	1534	3902	2993	1287	6666	6667	6668	6669	6670
5392	1218	359	6671	3420	3269	6672	304	4993	2160	2812	6673	6674	6675	6676	921	6677	2354	6678
6091	6679	1457	557	6680	6681	6682	2513	3295	6683	6684	6685	1336	3816	6686	6687	2091	6688	2656
6689	6000	4770	6690	463	3767	6691	6692	2491	5201	6693	6694	6695	6696	216	6697	6698	6699	5146
6700	6701	6702	6703	3376	3035	5746	6704	6705	3574	6706	6707	6708	6709	6710	450	5606	6711	6712
6713	6714	6715	6716	789	3869	6717	6718	3556	2620	6719	1976	1286	6720	6721	6722	2015	3554	6723
6724	2520	6725	3842	6726	3191	6727	1270	6728	6729	6730	1909	1312	6731	5051	3532	687	6732	6733
1874	1821	6734	3156	6735	6736	4847	1782	1970	3062	6737	6738	6739	6740	6741	1066	6742	58	1523
6743	315	2035	6744	6745	6746	6747	6748	6749	3561	3714	6750	6751	4648	6752	5265	6753	6754	3575
6755	6756	6757	3587	1088	6758	5348	6759	6760	1189	1995	2090	6761	4771	3778	3489	3691	1560	6762
6763	6764	6765	6766	6767	615	6768	6769	2775	3629	767	6770	6771	6772	6773	6774	5626	1431	1642
1433	1625	6775	3856	395	1106	6776	6777	1841	6778	6779	6780	5546	1119	2676	2614	6781	6782	6783
6784	2535	6785	506	1796	1783	1400	2414	6786	6787	2678	3579	6788	3398	4815	6789	2985	124	2102
6790	6791	2214	3544	2168	1083	6792	6793	2131	3790	3026	6794	2729	6795	6796	1695	6797	2432	6798
19	6799	6800	6801	6802	6803	6804	6805	6806	3218	1387	4866	1139	3885	6807	6808	6809	975	6810
6811	6812	1809	6813	6814	6815	2700	6816	6817	6818	6819	6820	773	6821	6822	5935	6823	2277	1736
1241	6824	6825	6826	163	6827	1087	727	6828	4967	6829	6830	6831	6832	1600	3347	6833	6834	136
4862	6835	965	292	2668	3764	3372	6836	6837	6838	6839	2916	1378	6840	3242	6841	6842	930	4897
1610	6843	1423	2070	5631	6844	6845	3053	5292	6846	479	6847	3287	1142	3634	6848	6849	6850	4723
6851	1193	3320	6852	6853	2652	2236	6854	614	4710	3711	6855	6856	1351	2899	312	6857	6858	6859
987	6860	863	6861	3433	6862	6863	41	6864	6865	6866	5891	290	2184	3744	6867	6868	3765	6869
6870	6871	6872	268	2452	6873	6874	6875	6876	610	3769	6877	6878	2798	6879	571	2000	6880	6881
5151	6882	2680	1244	6883	6884	6885	3801	3565	6886	966	5767	6044	6887	6888	6889	6890	6891	6892
6893	811	6894	4980	555	1980	2683	6895	2244	6896	6897	1311	648	137	820	6898	6899	3383	6900
2751	6901	186	6902	6903	6904	1633	6905	6906	5719	332	6907	2389	3598	791	6908	805	6909	324
1658	2609	3459	6910	6911	6912	6913	6914	6915	1891	231	877	6916	6917	2995	6918	1162	6919	1559
1717	4971	6920	6921	6922	1359	2304	6923	6924	6925	6926	5437	5285	6927	1233	2591	6928	2717	6929
6930	2101	6931	1846	2749	6932	6933	697	2390	6934	2067	6935	1059	6936	54	4671	448	554	1680
6937	6938	3171	3563	6939	6940	6941	3210	6942	6943	6944	6945	918	6946	111	6947	5297	61	6948
6949	2148	481	5861	6950	6951	6952	1735	6953	6954	2267	6955	3622	6956	6957	3864	6958	1551	6959
6960	6961	5335	2682	5769	6962	6963	6964	363	618	2908	6965	6966	6967	3381	6968	6969	6970	1456
2977	6971	3107	6972	191	6973	1408	1934	6974	6975	6976	3755	6977	6978	6979	5530	2231	3768	2710
6980	3545	3657	2261	1663	1636	3549	6981	3918	5331	6982	3353	383	6983	377	6984	6985	6986	2530
3695	6987	6988	6989	6990	3185	744	6991	6992	2154	6993	580	6994	6995	6996	1957	3252	6997	6998
3462	6999	2610	5506	5061	1145	3551	326	7000	7001	7002	7003	7004	5262	7005	7006	7007	7008	7009

4969	721	1440	7010	7011	7012	5906	5607	398	7013	1641	7014	7015	7016	3610	7017	7018	3141	116
6104	7019	1447	1621	1906	549	7020	7021	7022	2072	2715	7023	802	808	7024	7025	7026	5020	7027
7028	7029	7030	7031	489	7032	7033	7034	7035	1179	7036	264	3172	7037	4763	7038	7039	7040	7041
7042	7043	5931	7044	63	7045	3898	7046	7047	3725	7048	1200	7049	1375	266	2255	7050	2280	7051
7052	2842	7053	1533	7054	7055	967	7056	7057	7058	7059	5131	7060	1618	7061	7062	7063	3906	7064
7065	7066	7067	7068	7069	7070	7071	2312	5528	3850	7072	7073	7074	5312	2885	1815	452	7075	7076
1971	2301	7077	7078	7079	7080	7081	519	999	2913	164	925	7082	2351	7083	3506	7084	7085	4824
42	7086	7087	7088	7089	7090	7091	846	7092	7093	7094	7095	7096	7097	7098	7099	7100	7101	7102
7103	2537	7104	7105	7106	7107	7108	7109	2407	7110	7111	3942	2343	7112	1310	7113	7114	7115	7116
3760	7117	7118	7119	7120	7121	2601	7122	2746	7123	3184	7124	7125	7126	1333	7127	6014	7128	7129
7130	2428	7131	7132	7133	516	101	3298	7134	2900	587	7135	1368	3927	7136	7137	7138	541	7139
7140	7141	7142	7143	2783	3728	3905	7144	568	7145	7146	7147	1256	7148	1623	2887	957	5486	5175
7149	4735	3870	2415	7150	7151	1464	3177	7152	7153	7154	3225	7155	1115	7156	5436	7157	3387	7158
7159	1505	7160	2534	7161	7162	7163	1367	3809	7164	7165	7166	1585	7167	7168	7169	7170	3168	7171
7172	7173	2336	7174	3357	7175	5989	7176	5279	3487	7177	7178	7179	7180	7181	7182	7183	3159	7184
7185	7186	7187	7188	7189	2430	2069	3520	7190	2119	674	7191	353	2722	7192	2725	5439	2803	2673
7193	7194	7195	7196	7197	1685	1146	7198	7199	871	1867	2443	1904	3098	3391	7200	7201	4622	5690
7202	7203	7204	2478	7205	241	1687	7206	3186	302	250	629	2719	7207	836	7208	258	752	7209
7210	7211	7212	333	7213	2892	5673	7214	7215	7216	7217	1107	7218	2353	7219	657	2205	7220	7221
7222	7223	5532	3423	7224	7225	2204	7226	7227	1502	1124	7228	876	7229	5258	1057	5514	3815	7230
7231	7232	1598	7233	7234	2957	7235	391	3453	4970	7236	7237	7238	3715	7239	7240	7241	1121	3511
7242	7243	2053	2851	7244	5158	7245	1316	7246	3503	1299	1542	1744	7247	3687	1851	7248	7249	1529
5301	7250	2411	7251	421	7252	592	2739	7253	7254	1836	7255	3705	340	1953	7256	7257	6001	3311
7258	678	337	7259	3761	7260	3821	7261	2208	455	2819	143	2813	544	819	7262	7263	3217	7264
7265	4907	3795	3308	1945	1177	3615	5422	7266	1928	2630	3908	7267	1617	7268	105	7269	514	3442
7270	7271	3825	7272	339	7273	7274	7275	7276	7277	7278	3384	7279	7280	7281	7282	3763	7283	7284
3142	3868	7285	2861	7286	5759	7287	7288	7289	7290	3379	1288	901	7291	7292	7293	7294	3229	3444
1485	5684	2675	2393	7295	7296	3392	57	240	3047	7297	3070	7298	7299	7300	7301	7302	7303	1344
7304	3750	3312	3645	1828	3451	7305	7306	193	3872	7307	7308	5865	3621	5738	1473	7309	7310	2598
2801	219	7311	2079	7312	7313	7314	900	3781	2440	1265	7315	923	7316	7317	1334	7318	7319	5763
7320	1890	7321	7322	7323	7324	7325	7326	2565	7327	7328	7329	5239	7330	276	7331	7332	7333	7334
7335	7336	48	7337	321	5884	7338	3919	7339	3197	7340	7341	880	5411	7342	7343	7344	7345	7346
7347	7348	7349	1887	7350	1264	3792	3165	7351	7352	7353	7354	7355	1613	2868	7356	7357	4	5889
2835	7358	7359	7360	3356	7361	2557	7362	2545	4802	7363	1864	1777	7364	5781	7365	7366	7367	1309
7368	7369	2970	7370	7371	7372	7373	7374	933	1761	1679	509	1350	3840	2756	7375	2904	7376	3332
2264	3829	7377	7378	7379	7380	1104	7381	3712	7382	7383	3939	7384	7385	5089	5080	7386	5447	7387
7388	7389	7390	7391	5358	4669	7392	5116	4597	7393	170	2808	3317	6006	7394	7395	7396	7397	2402
5885	7398	7399	7400	7401	7402	1694	7403	7404	7405	51	5849	7406	2522	7407	7408	7409	7410	7411
7412	7413	7414	7415	7416	7417	7418	7419	7420	7421	7422	7423	7424	102	2094	7425	1918	7426	7427
4901	7428	1583	7429	7430	7431	460	7432	7433	1644	7434	7435	3800	7436	4634	7	7437	256	7438
7439	7440	7441	7442	7443	7444	7445	1964	7446	627	1572	7447	151	3407	1929	7448	7449	7450	2616
7451	2359	7452	2066	227	7453	7454	6029	2335	7455	7456	2515	7457	732	7458	7459	7460	7461	2662
2784	2611	1770	319	7462	7463	432	7464	5727	181	897	2564	7465	2492	1420	7466	7467	3009	7468
7469	7470	7471	7472	1580	74	7473	7474	7475	7476	3346	7477	7478	7479	7480	2146	7481	7482	492
2178	7483	630	7484	7485	7486	7487	1084	7488	747	735	7489	5893	839	7490	7491	7492	7493	7494
1338	7495	7496	1109	7497	3903	7498	7499	895	7500	5038	5878	2097	7501	7502	4873	2792	7503	7504
285	7505	2666	7506	2523	5666	2417	3049	7507	7508	7509	2914	1996	3557	7510	243	7511	7512	7513
7514	617	7515	5217	2647	1835	922	7516	7517	7518	2484	5561	7519	5714	3268	2470	7520	7521	7522
7523	1772	5999	2318	43	1248	7524	7525	1811	103	7526	403	3170	1448	7527	1888	7528	4950	7529
7530	7531	7532	3861	818	169	7533	1117	7534	4656	5896	3203	7535	7536	7537	7538	7539	7540	7541
7542	1752	1258	2268	7543	2911	2180	3559	7544	7545	7546	3282	7547	7548	1472	7549	7550	1018	7551
7552	7553	7554	5405	7555	7556	1850	2234	3425	7557	446	903	7558	7559	4667	7560	7561	7562	7563
3910	7564	7565	7566	7567	7568	405	5675	3754	7569	1395	7570	7571	7572	7573	1496	1325	7574	7575
1390	2322	2558	474	3473	1508	2284	7576	418	5519	7577	7578	4920	7579	7580	1974	7581	2008	1178
7582	3371	695	7583	7584	2555	1935	7585	7586	2896	3153	7587	3045	7588	176	780	1682	7589	7590
3393	7591	968	2305	7592	7593	4699	7594	7595	7596	7597	443	7598	2037	7599	1537	7600	7601	3922
1243	810	3947	3785	1444	7602	7603	1883	7604	1060	7605	1546	7606	1822	2679	7607	7608	603	7609
7610	7611	1441	1952	7612	7613	7614	2375	7615	7616	711	7617	7618	3437	2521	6045	1192	3823	7619
7620	7621	2636	4919	7622	1753	7623	7624	1003	7625	1842	7626	7627	3443	7628	7629	5066	5049	2382
2670	2219	7630	2894	1295	5155	1845	7631	5686	7632	938	7633	7634	5833	910	2626	2723	7635	5119
7636	1651	7637	3029	2494	7638	659	4694	3051	7639	1555	5848	7640	7641	16	6037	7642	7643	5992
3192	7644	7645	1914	7646	3278	7647	3265	127	7648	2303	7649	495	1091	7650	7651	2667	7652	2834
66	2011	1847	7653	1699	1413	5599	7654	6079	15	7655	3683	7656	1065	1660	7657	5680	7658	7659
3550	7660	7661	357	1224	833	368	6111	2232	7662	3326	7663	3193	7664	7665	86	7666	7667	436
7668	739	7669	7670	7671	4623	7672	7673	7674	7675	7676	7677	7678	7679	7680	7681	7682	7683	1261
7684	7685	7686	7687	2088	7688	7689	759	7690	7691	3535	7692	1501	3586	7693	7694	7695	7696	195
2635	2104	7697	3246	7698	7699	7700	7701	7702	3174	3892	7703	7704	7705	7706	2420	7707	718	7708
3450	3305	7709	2245	2547	7710	7711	7712	7713	7714	3427	122	7715	1567	7716	7717	7718	7719	656
7720	7721	7722	7723	7724	3028	3593	5130	5438	2228	7725	7726	7727	7728	2488	7729	7730	5902	7731
174	1916	2566	7732	68	7733	7734	1415	588	3818	336	7735	7736	7737	915	2742	7738	526	7739
7740	7741	5549	1769	3015	7742	7743	1784	7744	7745	2174	1535	7746	1710	7747	7748	7749	1714	7750
7751	7752	7753	7754	7755	1954	7756	7757	2176	7758	7759	2445	4960	7760	7761	7762	7763	155	2416
7764	3163	7765	3340	879	7766	1758	3010	7767	4892	2279	7768	5648	2711	7769	2339	7770	472	5156
7771	7772	7773	3771	7774	3119	7775	7776	3806	3793	512	7777	7778	196	2246	722	939	2223	3693
60	7779	7780	7781	7782	297	2552	1966	7783	7784	7785	7786	6023	1332	7787	7788	2860	533	7789
5025	7790	1374	7791	7792	7793	7794	3924	7795	2531	7796	7797	416	7798	2039	3008	7799	7800	2514

318	1674	7801	7802	7803	3681	7804	3237	7805	7806	3685	7807	7808	582	3641	1693	3118	7809	2728
4806	3180	440	7810	2108	7811	251	2152	1099	2197	7812	5554	1607	2839	2309	1061	3188	7813	2252
536	7814	7815	7816	7817	7818	7819	1050	7820	7821	7822	5844	7823	970	7824	2968	2540	7825	670
5452	598	2128	3132	7826	3494	2338	1789	7827	2871	3774	2926	4890	7828	7829	7830	7831	3240	7832
7833	3833	7834	7835	7836	468	7837	2297	7838	7839	7840	7841	5386	7842	7843	1141	7844	7845	4828
7846	2664	7847	7848	234	501	1135	7849	873	7850	7851	236	7852	2944	4632	7853	5304	7854	7855
53	7856	7857	7858	3625	7859	7860	7861	7862	1216	7863	7864	7865	5703	2681	7866	5895	3824	7867
7868	2310	7869	7870	1000	7871	244	976	1818	7872	1054	2613	4850	5244	7873	5965	7874	7875	1499
2797	35	7876	7877	7878	7879	3952	7880	775	1737	7881	7882	7883	7884	7885	7886	7887	7888	2175
3493	5715	72	7889	3147	3501	7890	7891	2563	7892	6033	7893	5466	7894	3569	85	7895	7896	1524
2867	7897	3858	7898	7899	964	7900	7901	211	7902	7903	3914	830	7904	1690	7905	2833	171	7906
7907	2435	355	867	7908	7909	2213	7910	7911	2859	7912	746	7913	5650	1622	2014	7914	7915	1424
7916	972	7917	100	7918	961	7919	1684	7920	7921	1592	7922	7923	2526	7924	5934	1308	736	4912
7925	1236	7926	2093	3483	7927	3434	7928	7929	7930	2206	7931	7932	2286	7933	5807	7934	2809	7935
7936	7937	7938	7939	7940	7941	7942	3093	7943	7944	224	859	2805	3034	7945	7946	7947	7948	7949
7950	2766	2274	2142	7951	5379	7952	7953	2187	3136	7954	7955	7956	2265	3787	7957	7958	7959	5291
1487	1715	7960	2930	1839	3644	2991	6110	2706	2898	7961	5576	2480	2370	4886	804	7962	793	7963
7964	7965	7966	5967	7967	7968	7969	7970	7971	7972	7973	3597	3066	7974	7975	7976	829	532	7977
7978	5092	5856	7979	7980	488	7981	7982	4677	7983	7984	92	7985	7986	7987	76	7988	7989	1932
7990	3802	3214	7991	7992	2733	5789	5704	7993	7994	7995	4952	5198	7996	7997	7998	5477	1992	1961
5193	3852	7999	8000	1098	1208	487	1404	8001	8002	8003	2928	2155	8004	8005	8006	8007	8008	146
3044	857	8009	8010	8011	3429	1029	2536	8012	2738	8013	3329	8014	8015	8016	8017	2649	1281	8018
589	2821	8019	2918	8020	2027	352	8021	8022	8023	8024	8025	8026	8027	2883	2638	8028	8029	5459
5668	8030	8031	3306	2278	521	8032	8033	8034	2590	8035	3125	2872	1819	8036	8037	71	8038	1993
8039	5234	1394	8040	8041	3022	451	8042	2043	1044	8043	5171	8044	2933	8045	8046	1294	3196	2529
8047	8048	1427	8049	5019	1409	8050	3000	8051	8052	4661	8053	213	3591	3409	8054	8055	2776	8056
8057	8058	8059	8060	8061	8062	2889	8063	1352	652	8064	2444	8065	8066	8067	8068	8069	2737	8070
8071	2519	2086	1234	1711	8072	8073	217	222	8074	1503	2573	1700	8075	8076	5975	8077	8078	8079
1019	535	8080	3189	2281	8081	168	484	8082	1893	8083	8084	2400	8085	3933	8086	3599	8087	8088
273	1754	8089	8090	8091	1275	8092	8093	8094	8095	3324	8096	5204	655	8097	8098	3375	2087	8099
3365	8100	8101	108	5523	847	5643	8102	8103	2042	37	3261	5570	5322	5871	2599	8104	8105	3773
8106	8107	8108	1196	8109	4659	8110	8111	1525	3811	4780	8112	8113	8114	639	1985	6020	8115	8116
5461	8117	8118	801	2410	737	8119	8120	8121	8122	5814	1211	3426	2271	8123	3507	1676	8124	2856
3319	2796	8125	2136	8126	3614	2845	984	8127	3412	8128	2539	8129	864	1586	8130	8131	8132	989
8133	1482	8134	8135	8136	8137	8138	23	6080	8139	5699	8140	1206	8141	187	1650	2581	2843	8142
1126	1869	8143	4871	8144	8145	2941	534	1451	8146	8147	8148	6096	328	924	294	6072	2691	8149
8150	8151	8152	1220	8153	3517	2369	8154	8155	156	1973	2140	1765	1458	8156	8157	8158	8159	8160
5691	2024	8161	8162	2437	8163	8164	8165	1797	8166	8167	106	8168	8169	242	8170	3081	1749	3144
8171	93	99	8172	8173	8174	8175	8176	2917	8177	8178	8179	1731	8180	8181	8182	8183	8184	8185
2429	1011	719	5972	3929	8186	8187	8188	8189	8190	5542	2173	8191	8192	8193	1528	3065	8194	8195
2084	8196	8197	8198	8199	2589	1958	1168	8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210
8211	8212	962	3343	5939	898	464	8213	8214	8215	8216	572	8217	8218	8219	1160	3227	8220	3626
3328	3624	1724	8221	3258	3882	8222	1047	3600	973	8223	8224	8225	8226	3576	884	5179	5928	8227
2757	5850	8228	2476	8229	8230	8231	8232	8233	8234	8235	8236	3291	947	3528	8237	8238	8239	8240
8241	8242	8243	8244	8245	2387	3143	8246	8247	8248	8249	8250	8251	8252	8253	743	1941	8254	3283
8255	1498	646	306	119	1965	8256	425	8257	8258	8259	8260	8261	323	3083	8262	343	682	3290
3827	520	8263	2663	5248	1005	8264	2158	8265	8266	8267	3206	8268	1062	8269	1154	3505	2316	8270
2012	8271	8272	8273	1742	8274	1199	5132	1132	852	1997	2781	6087	8275	2754	3834	2489	8276	8277
2551	8278	971	5890	1343	8279	2487	8280	8281	8282	313	3592	8283	8284	2392	5757	1657	3567	2460
8285	8286	8287	8288	8289	5225	8290	8291	8292	8293	8294	8295	8296	8297	1791	575	2033	8298	8299
2950	8300	8301	5827	8302	3804	2418	745	8303	8304	2283	8305	5988	1202	669	1830	8306	8307	8308
730	8309	8310	3789	8311	2077	8312	8313	1590	2961	8314	8315	2049	8316	1734	3247	8317	1075	8318
3881	8319	8320	8321	8322	8323	8324	3160	2256	8325	3274	1589	8326	5362	8327	6097	8328	8329	8330
733	3033	8331	1190	8332	1435	8333	8334	8335	2111	8336	279	2259	4690	8337	8338	8339	8340	1421
875	1937	5583	1575	1774	8341	8342	8343	1014	960	8344	8345	239	8346	8347	1289	5426	8348	5694
8349	8350	8351	461	8352	1452	1743	8353	4945	8354	8355	3547	8356	8357	2873	3337	4665	2230	8358
2785	8359	3813	3708	8360	3286	2149	8361	8362	412	8363	8364	8365	1076	8366	8367	8368	8369	8370
3310	1894	8371	2587	8372	8373	8374	8375	4898	8376	8377	8378	1897	8379	3916	8380	8381	714	8382
8383	8384	8385	8386	3150	8387	2237	1608	1686	2912	8388	3313	1905	8389	2130	1043	8390	8391	8392
473	8393	3584	2062	3617	3400	8394	8395	2532	5559	8396	8397	8398	5627	434	8399	8400	2300	8401
8402	1284	8403	796	1086	8404	8405	1643	8406	2799	2956	2482	1725	5609	8407	8408	2528	2585	8409
1998	4604	8410	5118	2665	8411	8412	8413	396	1377	8414	8415	8416	1861	2832	5936	662	2446	8417
8418	8419	8420	2658	8421	8422	8423	31	8424	8425	8426	8427	5482	8428	8429	8430	1871	8431	4577
165	8432	1947	531	8433	2973	8434	8435	8436	3745	121	3855	8437	8438	8439	2838	73	2905	8440
8441	8442	8443	5947	2031	8444	8445	8446	1125	2058	8447	3080	8448	4932	8449	8450	892	8451	496
3512	8452	8453	8454	5229	8455	5088	712	1172	2464	8456	8457	8458	528	8459	8460	4868	8461	8462
8463	704	8464	5133	3074	2307	8465	8466	3228	2346	2207	2852	1733	8467	8468	2893	5870	8469	4576
8470	2736	8471	8472	3915	2503	280	2438	413	978	8473	133	2254	8474	201	8475	671	8476	2341
5785	3349	8477	8478	8479	5639	8480	690	8481	1984	8482	8483	8484	2650	298	8485	1238	5139	2212
1917	8486	3484	3930	2582	2471	2910	8487	5203	8488	2891	87	8489	8490	2915	8491	1326	3419	1418
5110	8492	8493	8494	1994	115	1442	5973	8495	5851	404	8496	8497	8498	8499	8500	8501	8502	3851
8503	8504	8505	1210	8506	8507	1554	2098	1108	8508	1080	1219	1545	3949	4934	5353	8509	4936	8510
8511	8512	8513	3481	3762	3515	3297	8514	741	3921	8515	3573	2479	842	8516	480	1486	8517	1203
8518	8519	8520	1460	4664	2548	8521	3678	2934	5164	560	4812	2741	2608	8522	1113	8523	8524	3402
8525	2032	2870	8526	8527	8528	2496	3164	8529	8530	8531	271	8532	8533	8534	8535	8536	4915	8537
466	6003	8538	8539	8540	8541	5917	8542	1131	8543	8544	8545	8546	1207	172	1956	8547	2458	8548

2384	5946	8549	8550	3042	8551	1188	1827	8552	8553	4958	8554	8555	872	8556	189	8557	8558	8559
8560	1982	1801	757	1491	3036	8561	2726	2924	3499	8562	2986	3179	8563	2006	1833	8564	8565	3470
8566	8567	354	8568	8569	1282	8570	5905	278	8571	777	8572	8573	8574	5527	8575	2163	8576	834
8577	8578	8579	3668	3605	8580	1318	8581	1111	5296	8582	8583	8584	8585	8586	5077	4776	8587	8588
8589	3822	8590	3849	8591	1908	948	8592	454	2533	2724	8593	8594	8595	8596	8597	8598	8599	1492
3146	3848	8600	8601	8602	8603	8604	237	2380	3128	8605	3239	8606	8607	8608	893	8609	424	2345
202	8610	8611	8612	8613	3661	2377	8614	8615	3562	8616	8617	8618	8619	2467	2577	2073	2085	3071
8620	8621	3046	8622	8623	3232	8624	8625	1205	4756	3627	623	8626	8627	152	8628	2525	8629	415
8630	8631	8632	8633	8634	5094	8635	8636	2475	8637	5816	1136	1558	1254	2425	3943	2593	2421	1832
8638	1364	8639	8640	3460	787	5481	8641	2607	8642	1656	2811	8643	5573	8644	8645	1756	8646	83
684	8647	8648	2426	8649	8650	3948	766	3025	3726	3299	8651	8652	1615	8653	8654	8655	1252	8656
8657	5124	8658	8659	1838	3604	790	3135	574	8660	8661	8662	8663	8664	2082	2510	8665	2553	1570
3904	1280	1360	1230	8666	5180	8667	8668	8669	2095	8670	8671	8672	8673	8674	8675	8676	667	1412
8677	8678	8679	8680	287	3068	8681	8682	8683	1691	977	8684	8685	8686	8687	8688	8689	8690	637
8691	1008	8692	8693	8694	8695	626	8696	5349	1969	167	8697	8698	1578	5083	2466	8699	8700	8701
8702	8703	1794	8704	160	8705	706	8706	8707	912	596	8708	8709	8710	8711	3633	1305	8712	8713
8714	8715	707	3694	2328	8716	8717	8718	8719	8720	5126	5795	8721	786	8722	8723	1073	2646	5921
8724	3463	827	8725	3770	2169	8726	8727	8728	1425	2333	8729	8730	681	8731	4680	8732	8733	889
8734	3935	5693	8735	4908	8736	8737	8738	8739	8740	18	2946	2972	4748	3455	8741	8742	8743	1494
8744	717	1296	8745	2498	8746	8747	2427	2508	8748	1923	8749	8750	8751	3779	6	8752	1373	400
8753	8754	567	3092	8755	5084	8756	8757	1540	8758	2123	8759	1924	8760	8761	8762	8763	8764	3716
8765	8766	3213	8767	5790	2243	2641	1757	8768	8769	3116	8770	8771	8772	2409	8773	3064	8774	8775
8776	2177	8777	2594	5112	1348	3351	8778	8779	8780	8781	2269	8782	8783	8784	3857	8785	8786	8787
88	3063	1166	8788	2505	316	515	8789	3322	2935	2362	2507	8790	8791	8792	5518	8793	3373	8794
8795	1716	8796	2364	2258	5261	616	2276	758	8797	8798	779	8799	8800	1239	4600	1148	3090	8801
8802	8803	2020	8804	8805	1813	1022	8806	1837	109	3594	8807	3524	2568	2034	8808	8809	3476	795
8810	8811	8812	8813	2117	1999	4736	8814	3327	3643	2203	3211	8815	8816	8817	8818	4792	8819	1859
3382	1173	282	8820	8821	8822	8823	8824	8825	2251	641	3422	8826	305	624	8827	4808	8828	8829
8830	8831	8832	8833	1279	1748	8834	8835	540	5629	3653	1635	8836	8837	4857	8838	8839	8840	8841
8842	2705	8843	4874	8844	3335	246	8845	843	2615	8846	3743	8847	8848	1401	3488	8849	8850	2822
8851	8852	2153	379	5804	8853	5505	8854	8855	5760	1899	8856	8857	8858	8859	8860	147	8861	8862
407	498	2459	3017	8863	1823	1509	8864	8865	2139	3578	8866	8867	8868	8869	8870	1045	3739	8871
5174	8872	5281	2025	8873	8874	2592	1755	8875	8876	8877	1548	4608	8878	8879	8880	8881	8882	8883
8884	8885	2780	2864	3331	1214	8886	8887	8888	422	8889	8890	8891	1467	3803	8892	8893	2413	8894
8895	1703	8896	262	8897	1764	2857	2403	4930	1930	3113	4877	8898	3502	8899	4718	8900	1500	2052
8901	8902	2875	8903	3333	8904	837	8905	2826	1169	1788	8906	556	8907	8908	569	3525	2064	1611
705	33	8909	3945	724	5105	990	8910	8911	2442	763	8912	8913	8914	8915	8916	8917	8918	3318
8919	8920	8921	5867	3458	8922	2560	8923	8924	411	8925	2238	8926	8927	8928	6046	3321	8929	5031
983	5948	8930	2211	1704	8931	2054	5010	2285	1185	8932	3386	8933	8934	748	8935	8936	8937	8938
8939	8940	8941	8942	8943	8944	8945	5932	1273	8946	1903	274	2105	182	8947	8948	8949	601	444
2129	8950	8951	8952	8953	8954	22	8955	8956	8957	8958	8959	8960	1493	8961	8962	8963	139	3284
8964	8965	2399	3925	8966	685	2366	2814	8967	3111	8968	8969	8970	2118	1565	3271	1134	358	8971
8972	3703	5079	8973	8974	8975	3257	8976	3766	365	1561	2472	1647	956	8977	8978	5502	8979	8980
2982	1027	8981	558	179	8982	8983	8984	8985	3005	8986	8987	8988	8989	8990	2137	8991	8992	8993
8994	8995	8996	6084	6088	8997	2546	8998	5330	8999	9000	9001	3123	9002	9003	9004	9005	1187	1017
2450	4973	9006	9007	9008	9009	9010	848	9011	9012	2789	9013	9014	9015	9016	2554	1606	9017	1873
9018	3485	3102	3704	9019	117	9020	9021	9022	9023	263	1870	5383	2718	4766	9024	740	2465	2959
3468	9025	9026	9027	9028	9029	1516	1948	3212	9030	9031	3120	3509	9032	3187	9033	9034	945	2138
1872	4753	3012	9035	1595	9036	742	3293	1892	394	9037	562	2394	3941	9038	523	9039	2439	9040
9041	9042	9043	9044	2966	9045	9046	2275	5021	2196	78	8	3519	9047	9048	9049	9050	9051	9052
9053	6057	9054	1915	9055	6094	427	9056	2323	9057	9058	9059	9060	9061	1229	9062	5919	9063	5897
2888	9064	4853	497	9065	9066	9067	5412	9068	9069	4804	9070	2457	9071	2358	3883	4633	9072	5560
3300	3084	9073	9074	3692	9075	1707	9076	1834	9077	9078	9079	9080	9081	9082	3466	1476	9083	635
9084	9085	5995	9086	3428	9087	9088	2423	2048	9089	3699	9090	9091	2016	1201	625	410	3077	9092
118	9093	9094	9095	3756	9096	1342	385	5613	9097	840	9098	9099	9100	9101	9102	1648	9103	2215
9104	9105	9106	9107	993	9108	2788	1381	9109	4834	9110	9111	3094	4742	9112	1591	3706	9113	1825
9114	9115	198	887	1290	9116	9117	4612	507	9118	9119	344	3439	9120	9121	9122	9123	3152	9124
5729	24	9125	1553	9126	9127	2408	2431	9128	9129	3200	9130	2374	9131	5047	9132	3085	1371	9133
9134	3564	4831	2625	700	1386	1616	5310	5645	5488	9135	9136	9137	9138	9139	9140	4964	94	9141
3698	9142	9143	9144	9145	2759	1852	9146	9147	9148	9149	9150	5263	9151	9152	3032	9153	3215	9154
9155	5185	9156	281	9157	2882	510	465	9158	9159	9160	9161	2992	3133	30	9162	9163	9164	9165
9166	2096	9167	9168	1103	5070	2858	9169	672	1138	3553	9170	1026	9171	9172	9173	935	9174	453
3370	9175	954	2455	5024	9176	9177	5388	9178	9179	9180	3109	2164	2159	9181	573	9182	9183	1137
5579	1365	650	551	9184	612	1808	110	9185	9186	9187	1455	3104	9188	3537	1949	390	21	3523
9189	9190	9191	9192	9193	9194	9195	3723	9196	4924	9197	9198	9199	4910	9200	9201	5245	2932	9202
3359	9203	9204	9205	9206	9207	9208	9209	9210	9211	9212	9213	9214	1471	9215	6106	9216	5195	1596
389	9217	46	3281	69	9218	2239	2631	3893	1232	269	5952	9219	932	2657	5976	9220	9221	9222
9223	9224	6051	9225	9226	9227	9228	9229	3741	67	1978	9230	658	3900	9231	5187	3570	9232	1875
5641	1223	638	9233	9234	1579	9235	1116	9236	56	9237	1975	9238	9239	192	5582	9240	844	9241
5143	9242	3585	1477	1708	755	9243	3361	1654	9244	9245	9246	9247	1143	9248	9249	9250	44	9251
3048	9252	5157	5040	3928	376	4747	9253	9254	666	3052	1696	2490	1594	4713	9255	9256	1967	346
9257	3256	9258	1602	9259	9260	3334	761	426	9261	9262	1526	1068	9263	9264	9265	9266	9267	5775
9268	2732	1181	9269	2110	9270	3791	3275	1800	9271	9272	3309	4914	3696	9273	9274	9275	3527	9276
2056	9277	9278	9279	9280	9281	9282	1920	9283	9284	565	9285	3895	5260	5253	1655	1762	9286	9287
998	9288	9289	9290	9291	9292	9293	1399	9294	3182	9295	208	5440	9296	2612	9297	9298	9299	4959
9300	9301	9302	2979	3810	9303	5476	9304	1110	9305	9306	3050	1989	2527	1855	9307	3467	9308	9309

9310	9311	3590	3401	663	9312	9313	1671	9314	9315	9316	9317	9318	2765	9319	9320	9321	9322	4858	
9323	131	9324	9325	9326	9327	5808	1453	1078	9328	397	9329	9330	17	9331	645	3304	9332	9333	
307	348	9334	9335	9336	9337	9338	9339	9340	854	9341	4732	2132	9342	9343	3666	9344	9345	9346	
9347	9348	9349	9350	9351	2199	1946	9352	9353	9354	3408	1184	3686	5681	9355	1669	691	9356	5046	
260	600	9357	9358	9359	20	3418	9360	3377	3690	9361	9362	9363	9364	9365	9366	9367	2474	9368	
9369	5978	914	9370	9371	1383	1324	1114	9372	9373	9374	2740	2308	9375	2571	9376	9377	9378	2621	
9379	2570	9380	1191	2456	9381	9382	3364	9383	9384	2996	807	9385	9386	2451	9387	36	9388	5460	
9389	5413	9390	1706	9391	9392	9393	9394	9395	9396	1416	367	2714	9397	9398	9399	9400	462	1582	
9401	9402	9403	1267	3889	9404	9405	381	9406	9407	3680	9408	5172	2685	5910	4809	310	9409	247	
9410	9411	1688	9412	9413	9414	696	9415	9416	9417	9418	3731	9419	9420	3797	3007	9421	9422	9423	
9424	9425	2368	9426	9427	4652	9428	9429	9430	9431	6027	9432	4784	2807	3173	2253	2493	1795	9433	
183	4882	1824	9434	1157	3749	9435	9436	1102	9437	9438	1272	9439	9440	9441	9442	2381	1814	9443	
9444	9445	2787	9446	9447	1913	2388	9448	9449	9450	2806	3416	1865	9451	204	3631	3378	2640	1786	
9452	9453	5858	3411	9454	9455	9456	205	9457	9458	9459	9460	5741	2712	9461	9462	2651	1263	9463	
9464	1474	1085	9465	2061	579	9466	9467	9468	300	1882	1319	9469	9470	9471	545	9472	2188	9	
3366	9473	1506	9474	1681	749	4738	9475	9476	9477	9478	4575	9479	986	1380	2419	9480	1040	3516	
9481	9482	5651	2654	9483	3886	9484	9485	5162	1778	2695	9486	9487	9488	9489	221	3445	9490	9491	
3106	9492	2795	5499	9493	1799	1900	9494	9495	3558	2019	9496	2468	1096	3413	9497	3684	2556	5058	
9498	9499	9500	9501	1538	9502	1153	9503	3839	9504	9505	9506	9507	9508	9509	490	2976	3674	9510	
9511	9512	2	3477	1079	3096	1182	9513	9514	693	9515	2567	3931	9516	9517	9518	2473	9519	5342	
9520	9521	9522	6031	1807	2720	5170	9523	9524	1245	812	9525	9526	1226	9527	9528	9529	9530	3808	
9531	858	1933	9532	1668	1746	5373	9533	3492	4913	9534	2386	2659	1240	4844	2360	3175	9535	9536	
5843	3374	9537	698	5654	9538	1361	2516	9539	9540	1322	5797	2561	9541	9542	4615	3041	9543	1775	
9544	4772	9545	701	9546	9547	2958	9548	1922	9549	9550	9551	9552	3273	9553	2962	4769	9554	9555	
1768	9556	9557	9558	4926	9559	9560	9561	9562	9563	9564	1235	9565	9566	9567	9568	9569	9570	47	
2919	5985	2329	2242	9571	9572	9573	2869	9574	4654	1397	5657	5750	5873	9575	9576	9577	3016	9578	
3248	3339	9579	3474	1879	5005	1804	2841	5780	6022	9580	9581	5800	9582	9583	2939	9584	91	9585	
9586	9587	9588	9589	9590	9591	3030	9592	2777	9593	9594	2829	2818	9595	5841	9596	5056	9597	2363	
4739	9598	9599	9600	9601	9602	3675	6078	9603	9604	636	3296	9605	3038	9606	9607	309	9608	9609	
9610	1446	9611	2937	2241	9612	9613	9614	3449	9615	9616	9617	3338	3538	9618	9619	9620	2903	9621	
1683	9622	9623	9624	3057	9625	9626	9627	1495	2249	1925	9628	9629	634	5637	929	9630	9631	39	
753	9632	3639	9633	9634	9635	128	4887	3088	504	9636	3395	1292	5736	2081	1052	3740	9637	1552	
9638	9639	2075	9640	9641	1968	2686	9642	9643	9644	9645	3602	9646	9647	1806	9648	9649	9650	9651	
1581	3652	2837	9652	4635	9653	9654	9655	9656	845	3362	173	3531	9657	2502	2099	1006	9658	1128	
606	9659	5288	9660	9661	4730	9662	4888	4705	1302	4609	50	2604	135	9663	9664	9665	1209	9666	
3018	9667	909	3472	1164	9668	9669	9670	9671	1556	229	9672	6105	9673	2643	9674	3072	2846	4893	
2044	9675	9676	4724	1428	9677	9678	9679	9680	1520	9681	9682	9683	1097	9684	9685	9686	4689	9687	
1550	9688	3700	9689	1544	9690	9691	3536	6090	9692	676	2114	9693	1484	5016	9694	1250	1962	9695	
817	9696	771	2949	3054	9697	3784	9698	9699	9700	9701	153	9702	2260	5428	3216	3495	9703	2854	
9704	9705	1943	2330	9706	9707	5835	1167	9708	1902	9709	9710	9711	2449	9712	9713	3937	320	9714	
366	9715	1354	3710	9716	9717	2226	3688	660	9718	9719	1829	4801	9720	9721	9722	907	2299	9723	
3701	9724	4981	9725	9726	9727	2849	9728	5828	3294	9729	9730	3277	9731	1328	3920	3075	2121	1831	
9732	9733	9734	1081	9735	9736	3424	738	586	9737	9738	2233	9739	9740	9741	1955	5632	9742	9743	
420	9744	3669	9745	9746	5075	9747	9748	3388	141	9749	9750	9751	9752	9753	4688	4829	3259	2221	
1013	3720	9754	4843	1449	9755	1222	9756	9757	9758	3435	3285	2998	3946	611	9759	1588	9760	9761	
9762	9763	9764	9765	9766	9767	3853	9768	632	9769	9770	9771	5252	502	9772	3155	9773	9774	824	
429	2684	1942	9775	2483	9776	5173	9777	375	9778	9779	9780	9781	9782	9783	2750	199	1511	3148	
9784	530	9785	9786	9787	9788	9789	9790	9791	9792	3208	2688	158	9793	1419	9794	9795	2018	9796	
2406	906	9797	9798	5622	2704	9799	9800	9801	9802	9803	1155	1637	3267	2185	9804	870	9805	9806	
9807	9808	9809	9810	9811	9812	9813	751	1880	9814	9815	1283	9816	2617	9817	9818	9819	9820	1025	
9821	9822	9823	49	3385	9824	9825	9826	9827	1180	9828	9829	2816	3352	9830	9831	2401	9832	125	
1358	508	9833	9834	9835	112	9836	9837	9838	9839	9840	642	9841	9842	9843	9844	9845	816	3642	
9846	9847	9848	9849	813	5689	9850	9851	9852	9853	5188	9854	2850	9855	3718	2572	9856	1564	9857	
9858	9859	2433	9860	9861	80	6082	9862	9863	9864	3270	9865	1304	81	1384	303	2954	940	9866	
214	5012	4935	9867	1653	1130	9868	9869	9870	9871	3255	9872	9873	9874	9875	9876	9877	317	9878	
9879	9880	9881	2495	3606	866	9882	785	9883	856	3201	2463	5427	9884	9885	3940	9886	9887	1432	
797	220	9888	9889	9890	38	442	3672	3880	3630	9891	9892	2112	9893	5068	9894	9895	9896	2200	
2689	9897	3560	2355	9898	2951	9899	3542	9900	9901	9902	5500	9903	3037	1445	2770	1488	3031	6038	
1926	9904	414	2981	3161	9905	5535	1194	3899	860	5009	1392	226	713	9906	2735	1640	3664	5874	
5352	9907	469	9908	9909	9910	9911	3817	9912	5106	9913	1785	9914	9915	9916	943	9917	9918	9919	
2404	1475	2748	9920	9921	9922	9923	9924	4820	9925	2344	9926	9927	9928	2634	3518	9929	3421	9930	
9931	249	1566	9932	9933	9934	3632	9935	9936	9937	9938	9939	9940	9941	9942	9943	1093	9944	3209	
9945	850	9946	9947	2963	9948	1987	511	1112	2967	9949	593	1574	9950	2047	9951	9952	1266	9953	
9954	9955	9956	459	1426	9957	3039	3020	9958	5	9959	9960	9961	9962	9963	9964	2227	9965	9966	
9967	286	9968	9969	608	9970	9971	64	9972	9973	920	1372	9974	9975	1021	1133	9976	9977	9978	
9979	2143	6076	9980	9981	9982	3344	715	9983	9984	5601	4625	3336	9985	1151	9986	9987	9988	3406	
9989	1662	9990	2622	1531	9991	5491	9992	1041	9993	2895	2116	2125	9994	9995	9996	9997	3648	9998	
9999	3167	2071	1313	10000	10001	3266	10002	10003	3702	5747	10004	10005	1064	5451	10006	1429	4822	3581	
4702	2596	643	10007	10008	10009	10010	3205	10011	253	10012	10013	10014	6009	10015	2844	10016	10017	825	
2186	3871	3103	2292	5803	10018	10019	5701	10020	3003	10021	5968	10022	10023	10024	10025	4903	10026	699	
522	2453	10027	2825	10028	10029	1	364	2115	10030	10031	10032	10033	10034	1557	1307	10035	10036	10037	
10038	10039	10040	2127	10041	10042	673	10043	10044	10045	3134	3464	10046	10047	10048	10049	3689	10050	10051	
2578	10052	10053	10054	10055	10056	10057	5213	1402	1024	10058	10059	6008	1601	10060	5473	3854	10061	729	
10062	10063	223	1242	1257	1303	10064	10065	10066	3110	10067	10068	3846	10069	10070	10071	10072	157	3508	
2862	2191	10073	563	10074	2745	10075	1519	3202	3737	1820	1738	10076	25	2753	10077	10078	10079	10080	
161	2144	686	10081	10082	10083	10084	10085	5821	2840	10086	4781	1228	5553	10087	2983	10088	10089	3783	

10090 3884 10091 10092 10093 828 10094 2716 4963 79 3859 1382 10095 4725 1329 936 10096 2672 2361
10097 10098 3151 1816 10099 1231 10100 10101 3500 1271 10102 3909 831 10103 2965 1009 10104 10105 10106
10107 10108 525 2955 10109 10110 335 5900 10111 5682 10112 350 1868 10113 10114 2588 3149 10115 4684
10116 10117 4607 10118 3894 10119 10120 10121 10122 10123 10124 2317 70 10125 1225 2396 1518 2921 5468
10126 10127 934 10128 10129 419 2878 437 10130 10131 10132 10133 3440 10134 5108 2397 602 10135 10136
1848 206 891 1463 5783 10137 10138 1144 10139 10140 3454 1391 1856 4599 10141 10142 10143 361 10144
10145 10146 10147 10148 10149 342 144 10150 10151 5700 10152 885 2632 2619 10153 10154 3758 3019 2952
10155 10156 5548 2405 2774 10157 10158 3055 10159 10160 3130 878 10161 10162 2220 4947 296 1049 10163
10164 10165 3504 10166 10167 10168 2518 4761 10169 1323 10170 10171 10172 10173 10174 2943 1939 10175 2195
2182 886 10176 10177 26 10178 2007 5287 1127 1521 4992 140 10179 5319 1353 212 3682 10180 3241
10181 10182 10183 10184 10185 10186 10187 132 1513 2080 2931 10188 10189 2013 10190 1728 10191 10192 10193
10194 2866 10195 10196 10197 10198 10199 10200 3838 10201 1614 10202 10203 369 10204 5805 10205 1293 1779
175 1527 2802 10206 2541 3738 1793 5721 10207 3124 2141 10208 3611 911 10209 710 10210 10211 1314
10212 849 3513 10213 3748 10214 1901 10215 2348 10216 10217 3105 2637 2974 10218 10219 10220 3101 10221
10222 10223 3131 3534 10224 10225 10226 2980 10227 10228 10229 10230 1849 5792 10231 5887 3496 1443 10232
992 10233 10234 4787 10235 5635 10236 2314 314 5958 10237 10238 2810 2901 10239 1369 2761 10240 10241
2696 5227 2855 486 10242 1481 10243 3828 2122 2340 1896 10244 3243 2574 3805 1721 2595 2747 2302
1675 10245 10246 10247 10248 5257 5125 3236 10249 5120 10250 3831 5420 10251 5424 10252 3250 996 10253
10254 883 10255 5435 1161 10256 188 10257 10258 10259 10260 10261 3583 1430 55 10262 3316 1931 10263
10264 10265 10266 10267 373 997 2167 283 10268 10269 10270 2311 2559 10271 10272 10273 10274 10275 10276
10277 10278 2925 10279 1510 10280 10281 2549 386 10282 10283 1175 10284 10285 10286 10287 3837 2218 10288
3913 2022 10289 10290 10291 10292 1101 10293 1792 10294 2057 325 1436 2017 664 10295 3091 10296 2771
10297 1092 10298 517 10299 3221 1298 10300 10301 10302 5649 3735 10303 1058 10304 10305 345 10306 1070
10307 581 1988 4923 2133 1465 2063 3154 10308 5344 441 10309 2509 10310 10311 10312 4835 10313 10314
5773 10315 10316 2874 10317 10318 2156 1507 5964 1844 10319 3127 10320 10321 3121 10322 10323 10324 5994
10325 10326 166 10327 291 10328 1038 265 3169 5026 10329 10330 10331 10332 10333 3307 1751 10334 3636
10335 10336 248 10337 10338 3230 10339 10340 10341 10342 1355 1337 10343 838 32 3086 2999 10344 10345
881 10346 10347 1237 1422 10348 10349 2828 10350 1571 823 10351 10352 2734 10353 10354 10355 1726 10356
10357 10358 10359 10360 10361 10362 10363 5784 3021 3539 10364 4641 10365 10366 2648 10367 1597 2026 5027
10368 2040 10369 10370 10371 10372 2257 10373 10374 2677 10375 809 3772 5206 10376 1593 10377 10378 10379
10380 2331 10381 10382 10383 10384 3522 3566 3874 2126 184 10385 10386 347 10387 10388 10389 3204 5177
10390 13 10391 10392 3244 1547 215 10393 10394 10395 4640 82 10396 10397 10 1514 10398 10399 10400
10401 1497 10402 10403 3873 10404 1919 10405 988 97 3002 10406 10407 10408 10409 2512 10410 10411 10412
10413 10414 10415 764 10416 1406 5097 10417 10418 2469 1512 2378 2764 10419 5846 10420 2461 65 10421
5908 10422 10423 3190 597 10424 3796 3436 5624 10425 10426 10427 1417 2209 10428 10429 4861 3663 3676
245 10430 10431 10432 3011 10433 10434 10435 10436 10437 10438 10439 10440 10441 2773 2820 10442 10443 10444
5525 1012 10445 10446 10447 10448 6011 10449 10450 10451 10452 10453 2772 10454 10455 10456 3394 10457 10458
10459 1152 2940 10460 1082 10461 3254 10462 2376 10463 3367 10464 5955 10465 5545 10466 10467 3498 10468
2755 10469 10470 2347 10471 10472 1379 2767 203 10473 4719 1227 10474 10475 3014 1016 10476 10477 10478
311 1972 3234 953 3115 10479 2194 3108 10480 200 731 10481 10482 10483 10484 599 2997 10485 10486
10487 10488 10489 10490 2642 2170 10491 10492 1732 194 10493 1002 3847 3004 2078 10494 3944 10495 10496
10497 832 267 10498 142 10499 5829 10500 1150 10501 10502 10503 3430 10504 10505 1195 10506 1056 1624
382 10507 5400 10508 2441 10509 1803 456 10510 3095 3303 272 5683 5847 3655 3775 1898 1720 10511
3891 10512 10513 4949 1604 10514 10515 10516 10517 10518 10519 10520 2282 821 10521 10522 138 10523 1723
3145 10524 10525 10526 10527 3555 2692 10528 1174 1349 1639 89 10529 5127 10530 546 3251 10531 2289
10532 10533 591 10534 10535 10536 2157 10537 10538 10539 1340 10540 1253 10541 3363 1863 5028 10542 10543
10544 3670 10545 2823 10546 10547 10548 10549 1790 10550 494 1854 2945 5209 4814 2990 10551 869 10552
10553 10554 10555 10556 10557 2879 10558 10559 3753 3023 10560 331 2922 10561 3350 10562 1341 10563 10564
10565 1739 10566 10567 862 10568 590 10569 10570 1317 288 2295 10571 2412 10572 10573 3490 114 10574
2235 5556 3860 10575 10576 10577 888 5509 360 10578 10579 3089 10580 10581 5990 10582 10583 3262 3199
1204 3059 10584 2501 10585 10586 10587 10588 10589 5661 3568 10590 10591 10592 2987 6059 3660 10593 10594
10595 10596 10597 10598 10599 1747 3314 10600 10601 5060 478 10602 1462 1388 10603 10604 1940 4928 10605
10606 814 1370 10607 10608 2713 10609 10610 10611 10612 10613 3865 10614 10615 10616 1877 1159 10617 10618
10619 952 5718 10620 10621 10622 10623 10624 10625 10626 1959 10627 822 10628 5892 10629 10630 10631 1573
10632 10633 1670 3380 4968 628 1439 4953 10634 3612 10635 1031 585 10636 2989 5216 493 10637 10638
2890 2731 10639 10640 4657 10641 10642 2151 10643 4998 1046 10644 10645 10646 4879 3533 784 10647 10648
2270 10649 10650 5907 4644 10651 4610 3245 10652 1652 5100 10653 10654 2975 10655 10656 3719 3752 10657
10658 10659 2352 10660 10661 10662 5363 230 2109 2627 10663 10664 10665 723 2357 2372 10666 10667 10668
1632 2202 10669 10670 10671 1071 10672 10673 5445 10674 6107 2618 10675 2379 10676 3730 10677 1705 800
2041 2906 10678 10679 2248 3722 10680 10681 10682 10683 10684 2356 10685 10686 2424 2697 5868 3582 10687
5516 10688 10689 10690 10691 10692 2021 1840 301 2752 10693 10694 10695 10696 10697 1077 10698 1730 10699
3777 10700 10701 3673 10702 942 10703 10704 4896 10705 10706 1853 661 2511 3843 10707 5324 10708 5957
5308 1362 10709 10710 2224 4794 10711 1857 10712 10713 3181 2165 10714 10715 10716 3799 10717 3360 3441
10718 5300 10719 10720 1094 1912 1745 10721 10722 2103 1089 778 10723 3677 485 10724 5487 5159 10725
1577 90 10726 10727 3325 2606 10728 5774 6015 644 1469 10729 2542 10730 865 10731 1697 10732 374
10733 1881 10734 393 3479 10735 4588 10736 3628 10737 1215 10738 10739 10740 2538 10741 10742 10743 10744
10745 277 2699 10746 10747 10748 399 1197 2791 10749 10750 5647 908 1664 10751 10752 10753 1698 10754
10755 2266 3457 2786 14 1646 1470 1357 10756 1549 3231 10757 10758 1862 10759 5446 1936 10760 10761
190 10762 3938 3456 10763 651 505 10764 10765 10766 1628 10767 10768 10769 3279 633 235 3183 371
5113 10770 3637 1750 10771 1393 232 559 3901 180 10772 10773 150 10774 10775 3552 10776 679 10777
10778 10779 10780 10781 3667 10782 10783 725 3746 10784 1612 1331 680 2768 10785 5969 2923 2150 1712
1320 10786 728 688 1878 529 607 4944 10787 5098 10788 10789 10790 3786 185 11 1277 10791 2325
2698 10792 10793 10794 10795 4649 1450 1938 10796 10797 5915 3323 1759 994 10798 3288 10799 10800 10801
3114 10802 10803 561 10804 10805 2881 2210 10806 2727 3724 2334 10807 10808 10809 10810 10811 3249 10812
1247 2287 10813 10814 10815 2273 5515 913 861 10816 5610 2600 10817 10818 10819 408 2135 129 10820
5940 772 10821 5182 1437 1921 10822 3289 10823 1667 10824 10825 10826 10827 3917 10828 10829 10830 3405

10831 10832 10833 10834 10835 3609 4638 10836 10837 5508 10838 10839 3832 10840 3658 10841 10842 10843 28
584 10844 10845 3404 10846 4778 10847 750 3603 4819 499 3024 10848 1118 5717 2050 5979 1489 2603
10849 10850 3478 774 3157 10851 10852 10853 284 10854 10855 10856 10857 351 10858 2703 1951 1866 4581
10859 890 10860 10861 1249 10862 10863 10864 10865 10866 10867 5483 10868 10869 10870 10871 10872 10873 4883
10874 10875 10876 10877 10878 4668 10879 10880 4731 10881 10882 10883 10884 10885 10886 10887 10888 10889 10890
4614 10891 10892 10893 10894 10895 10896 10897 5698 10898 10899 10900 10901 10902 10903 10904 10905 10906 10907
10908 10909 10910 10911 10912 10913 10914 10915 4619 4758 5652 10916 10917 5660 5290 10918 10919 10920 10921
5434 10922 10923 10924 10925 10926 10927 10928 10929 10930 5672 10931 10932 10933 10934 10935 10936 10937 10938
10939 10940 10941 10942 10943 10944 10945 10946 10947 10948 5007 4911 10949 10950 10951 10952 4630 10953 10954
10955 10956 10957 10958 10959 10960 10961 10962 10963 10964 10965 10966 10967 10968 10969 5457 10970 10971 10972
10973 10974 5334 10975 10976 10977 10978 10979 10980 10981 10982 5756 10983 10984 10985 10986 6083 10987 10988
5916 10989 10990 10991 10992 4909 10993 10994 10995 10996 10997 10998 10999 11000 11001 11002 11003 11004 11005
11006 5289 11007 11008 11009 11010 11011 5429 11012 11013 11014 11015 11016 11017 5901 11018 11019 11020 11021
11022 11023 11024 11025 11026 11027 5594 11028 11029 11030 11031 11032 11033 11034 11035 11036 11037 11038 11039
11040 4962 11041 11042 11043 11044 11045 4768 11046 5295 11047 11048 5284 11049 4704 11050 11051 11052 5140
11053 11054 5377 11055 6050 11056 11057 11058 11059 11060 11061 11062 11063 11064 11065 6005 11066 4646 11067
11068 11069 4613 11070 11071 11072 11073 11074 11075 11076 11077 11078 5671 11079 4695 11080 11081 11082 11083
11084 11085 11086 11087 11088 11089 11090 11091 11092 11093 11094 11095 11096 11097 11098 11099 11100 11101 11102
11103 11104 5343 11105 11106 11107 11108 11109 11110 11111 11112 6063 11113 11114 4716 4830 11115 11116 11117
11118 11119 11120 11121 11122 6041 11123 5962 11124 11125 11126 11127 11128 11129 11130 11131 11132 11133 11134
11135 11136 11137 11138 11139 11140 11141 11142 11143 4927 11144 5181 11145 11146 11147 11148 11149 11150 5966
4578 11151 11152 11153 11154 11155 5857 11156 11157 11158 11159 11160 11161 11162 11163 11164 11165 11166 11167
11168 11169 5927 11170 11171 11172 11173 5581 11174 11175 11176 11177 4740 11178 11179 11180 11181 11182 11183
11184 11185 11186 11187 11188 11189 11190 11191 11192 11193 4675 11194 11195 5082 11196 11197 5880 11198 11199
11200 11201 5540 11202 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11213 11214 11215 11216 11217
11218 11219 11220 11221 11222 11223 11224 11225 11226 11227 11228 11229 11230 11231 11232 11233 11234 11235 11236
11237 11238 11239 11240 11241 11242 11243 11244 11245 5971 11246 11247 11248 11249 11250 11251 11252 11253 11254
11255 11256 11257 11258 11259 11260 11261 11262 11263 11264 5894 11265 11266 11267 11268 11269 11270 11271 11272
11273 11274 4624 11275 11276 11277 11278 11279 4984 11280 11281 11282 11283 11284 11285 11286 11287 11288 11289
5676 11290 11291 11292 11293 11294 5628 11295 11296 11297 11298 11299 5018 4703 11300 11301 11302 11303 11304
11305 11306 11307 11308 11309 11310 11311 6102 11312 11313 5492 11314 11315 11316 11317 11318 5771 11319 11320
11321 11322 11323 11324 11325 11326 11327 11328 11329 11330 5275 4655 11331 11332 11333 11334 5136 5980 11335
11336 11337 11338 11339 11340 11341 11342 4734 11343 11344 11345 11346 11347 11348 5465 4860 11349 11350 11351
5199 11352 11353 11354 11355 11356 11357 5564 11358 11359 5731 11360 11361 5355 5762 11362 11363 11364 11365
11366 5183 11367 11368 11369 11370 11371 11372 11373 11374 11375 11376 11377 11378 4939 11379 11380 11381 11382
5357 11383 6032 5367 11384 11385 11386 11387 11388 11389 5588 11390 11391 11392 11393 11394 11395 11396 11397
11398 11399 11400 11401 11402 11403 11404 11405 11406 11407 11408 11409 11410 11411 11412 11413 11414 11415 11416
11417 11418 11419 11420 11421 11422 11423 5230 11424 11425 5495 11426 11427 11428 11429 11430 11431 11432 11433
11434 11435 11436 11437 11438 11439 4726 11440 11441 5003 4785 5240 11442 11443 11444 11445 11446 11447 11448
11449 11450 11451 5655 5493 11452 11453 4708 4580 11454 11455 11456 11457 11458 11459 11460 11461 5048 11462
11463 11464 11465 11466 11467 11468 11469 6061 11470 5250 11471 11472 11473 11474 11475 5149 11476 11477 11478
11479 11480 11481 11482 11483 11484 11485 11486 11487 11488 11489 11490 11491 11492 11493 11494 11495 11496 5608
11497 11498 11499 5882 5071 4986 11500 11501 11502 11503 11504 5043 11505 11506 5562 11507 11508 11509 11510
4786 11511 11512 11513 11514 11515 11516 11517 11518 11519 11520 11521 11522 11523 11524 5678 11525 11526 11527
11528 11529 11530 11531 11532 11533 11534 5702 11535 11536 11537 4938 11538 11539 11540 11541 11542 11543 5722
11544 11545 11546 11547 11548 11549 11550 11551 11552 11553 11554 11555 11556 11557 11558 11559 11560 11561 11562
11563 11564 11565 11566 11567 11568 11569 11570 11571 11572 11573 11574 11575 11576 11577 11578 11579 11580 11581
11582 11583 5237 11584 11585 11586 11587 11588 11589 11590 11591 11592 11593 11594 11595 5148 11596 11597 11598
11599 11600 11601 11602 5685 11603 4666 11604 11605 11606 11607 11608 11609 11610 11611 11612 11613 11614 11615
11616 11617 11618 11619 11620 11621 11622 11623 11624 11625 11626 11627 11628 11629 11630 11631 11632 5086 5621
11633 11634 11635 11636 11637 11638 11639 11640 5517 11641 11642 11643 11644 11645 5200 11646 11647 11648 11649
11650 11651 11652 11653 11654 11655 5744 11656 4602 11657 11658 11659 11660 11661 11662 11663 11664 5035 11665
11666 11667 11668 4586 5726 11669 4637 4603 11670 11671 11672 11673 11674 11675 11676 11677 11678 11679 11680
11681 11682 11683 11684 11685 11686 11687 11688 11689 11690 11691 4991 5674 11692 11693 11694 11695 5552 11696
5147 11697 11698 11699 11700 11701 5380 11702 11703 11704 11705 11706 11707 11708 11709 5811 11710 11711 11712
11713 11714 11715 11716 11717 11718 11719 11720 11721 11722 11723 11724 5612 11725 11726 5623 11727 5283 11728
6052 11729 11730 4869 11731 11732 11733 11734 11735 11736 11737 11738 11739 11740 11741 11742 5103 11743 11744
11745 11746 11747 11748 11749 11750 4867 11751 11752 11753 11754 11755 11756 11757 11758 11759 11760 11761 11762
11763 11764 11765 11766 11767 5834 11768 4902 11769 11770 11771 4662 4974 4693 11772 11773 5998 11774 11775
11776 11777 5276 11778 11779 11780 4779 11781 11782 11783 11784 11785 4954 5309 11786 11787 5282 11788 11789
11790 11791 11792 6099 11793 6060 5091 11794 5474 11795 11796 11797 11798 5325 11799 11800 11801 11802 11803
11804 11805 11806 11807 4611 11808 11809 5925 5212 11810 5706 11811 11812 11813 11814 11815 5806 11816 5724
11817 11818 11819 11820 11821 11822 11823 11824 11825 11826 11827 11828 11829 5557 11830 11831 11832 4626 11833
5101 11834 11835 11836 11837 11838 5266 11839 11840 11841 11842 11843 11844 11845 11846 11847 11848 11849 11850
11851 11852 11853 11854 11855 11856 11857 11858 11859 11860 11861 11862 11863 11864 11865 11866 11867 5235 11868
11869 11870 11871 11872 11873 11874 11875 5709 11876 11877 11878 4885 11879 11880 11881 11882 11883 11884 11885
11886 11887 11888 11889 11890 11891 11892 11893 11894 11895 11896 11897 11898 11899 11900 11901 11902 11903 11904
11905 5062 11906 11907 11908 11909 11910 11911 11912 11913 11914 11915 5328 11916 11917 5053 11918 11919 11920
11921 5470 11922 11923 11924 11925 11926 11927 11928 11929 11930 11931 11932 11933 5286 11934 11935 11936 11937
11938 11939 11940 11941 11942 11943 11944 11945 11946 11947 11948 11949 11950 11951 11952 11953 11954 4584 11955
11956 11957 11958 11959 11960 11961 11962 11963 11964 11965 11966 11967 11968 11969 5432 11970 11971 11972 11973
11974 5067 11975 11976 11977 11978 11979 11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
11992 11993 11994 5567 5830 11995 11996 11997 11998 11999 12000 12001 12002 12003 12004 12005 12006 12007 12008
12009 12010 12011 12012 12013 12014 12015 12016 12017 4999 12018 12019 12020 12021 5302 12022 12023 12024 12025
12026 12027 12028 12029 12030 12031 12032 12033 5255 12034 5462 12035 12036 12037 12038 12039 12040 12041 5114
12042 12043 12044 12045 12046 12047 12048 12049 12050 6010 12051 12052 12053 12054 12055 12056 12057 12058 12059

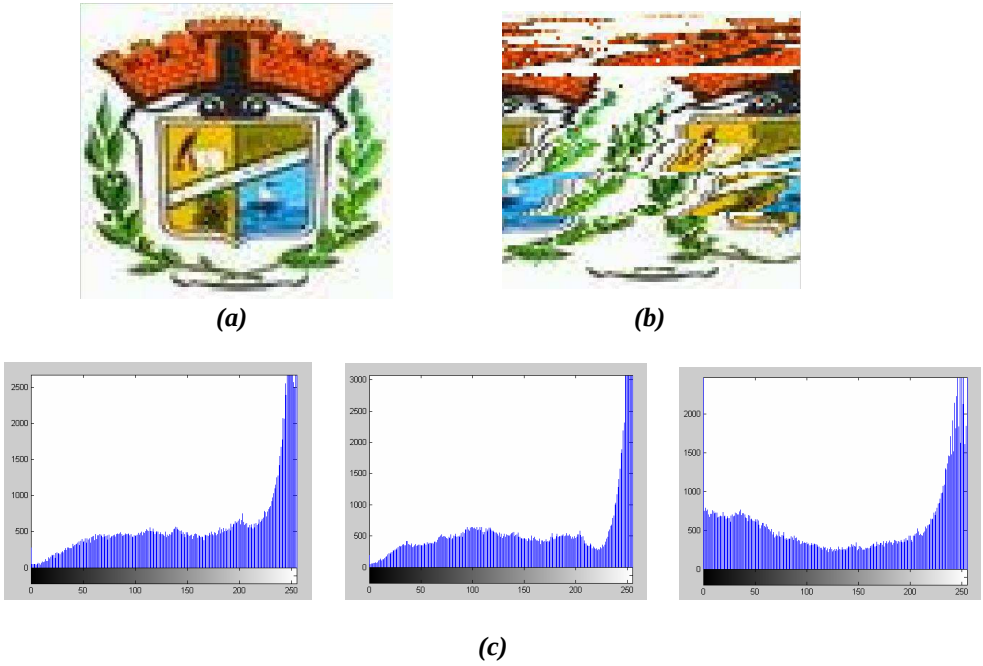
12060 12061 12062 12063 12064 5150 5299 12065 12066 12067 12068 12069 12070 12071 12072 12073 12074 12075 12076
5864 12077 12078 12079 5555 12080 12081 12082 5191 5912 12083 12084 12085 12086 12087 12088 12089 12090 12091
12092 12093 12094 12095 12096 4881 12097 12098 12099 12100 12101 12102 12103 12104 12105 12106 12107 12108 12109
4825 12110 12111 12112 12113 12114 12115 5455 12116 12117 5590 12118 12119 12120 5840 12121 12122 5145 12123
12124 12125 12126 12127 12128 12129 12130 12131 12132 12133 12134 12135 12136 12137 12138 4823 4764 12139 5087
12140 12141 12142 12143 12144 12145 12146 12147 12148 12149 12150 4686 12151 12152 12153 12154 4899 12155 12156
12157 12158 12159 12160 12161 12162 12163 12164 12165 12166 12167 12168 6025 4948 12169 12170 5577 12171 5395
6100 12172 12173 12174 12175 12176 12177 12178 12179 12180 12181 12182 12183 12184 12185 12186 12187 12188 12189
12190 12191 12192 12193 12194 12195 12196 12197 12198 12199 12200 12201 12202 12203 12204 12205 12206 12207 12208
12209 12210 12211 5327 12212 12213 12214 5630 6026 4925 12215 12216 12217 12218 12219 12220 12221 12222 12223
12224 12225 12226 12227 12228 12229 12230 12231 12232 5095 12233 5360 5845 12234 12235 6092 5135 12236 12237
12238 12239 12240 12241 12242 12243 12244 12245 12246 12247 12248 12249 12250 12251 12252 5898 12253 12254 12255
12256 12257 12258 12259 12260 12261 12262 12263 5163 12264 12265 12266 12267 12268 12269 12270 12271 12272 12273
12274 12275 12276 12277 5954 12278 12279 12280 12281 12282 12283 12284 4593 12285 12286 12287 12288 12289 12290
12291 12292 12293 12294 12295 4979 5526 5039 12296 12297 12298 12299 12300 12301 12302 12303 4674 4585 12304
12305 12306 12307 12308 5862 12309 5041 12310 12311 12312 12313 12314 12315 12316 12317 12318 12319 12320 12321
12322 12323 12324 12325 12326 12327 12328 5667 12329 5176 12330 5869 12331 12332 12333 12334 12335 5378 12336
12337 12338 12339 12340 12341 12342 5022 12343 5758 12344 12345 5123 12346 12347 5034 12348 12349 5663 12350
12351 12352 5096 5345 12353 12354 12355 12356 12357 12358 12359 5339 12360 12361 5403 5000 12362 12363 12364
12365 12366 12367 12368 12369 12370 12371 12372 12373 12374 12375 12376 12377 12378 12379 12380 12381 12382 12383
12384 12385 12386 12387 12388 12389 12390 12391 12392 12393 12394 12395 12396 12397 12398 5679 12399 12400 12401
12402 4905 12403 12404 12405 12406 12407 12408 4639 4889 12409 12410 12411 12412 12413 12414 5410 12415 12416
12417 12418 12419 12420 12421 4579 12422 12423 12424 12425 12426 12427 12428 12429 12430 12431 12432 5589 12433
12434 12435 12436 12437 12438 5616 12439 12440 12441 12442 12443 12444 12445 12446 12447 12448 12449 12450 12451
12452 12453 12454 12455 4891 12456 12457 12458 12459 12460 12461 12462 12463 12464 12465 12466 12467 5524 12468
5987 12469 12470 12471 12472 12473 12474 12475 12476 12477 12478 12479 12480 12481 12482 5316 12483 12484 12485
12486 4658 12487 12488 12489 4746 12490 12491 4745 12492 12493 12494 12495 12496 12497 12498 12499 12500 12501
12502 5002 12503 12504 12505 12506 12507 12508 12509 12510 5037 12511 12512 12513 12514 12515 12516 12517 12518
12519 4818 4791 12520 12521 12522 12523 12524 5761 12525 12526 12527 12528 12529 12530 12531 12532 12533 12534
12535 12536 5602 5072 12537 12538 5055 12539 12540 12541 12542 12543 12544 12545 12546 12547 5677 12548 12549
12550 12551 12552 12553 12554 12555 12556 12557 5178 4983 12558 12559 5093 12560 12561 12562 12563 5501 12564
12565 12566 12567 12568 12569 12570 12571 12572 12573 12574 6075 12575 12576 12577 12578 12579 12580 12581 12582
12583 12584 12585 12586 12587 12588 12589 12590 5341 12591 12592 12593 12594 12595 6036 5859 12596 12597 12598
5636 12599 12600 12601 4692 12602 12603 12604 12605 12606 12607 12608 12609 5728 12610 12611 12612 12613 12614
12615 5520 12616 12617 12618 12619 12620 12621 12622 5475 12623 12624 12625 12626 12627 12628 12629 12630 12631
12632 12633 12634 12635 5743 12636 12637 5390 12638 12639 12640 12641 12642 12643 12644 12645 12646 4741 12647
12648 12649 12650 12651 12652 12653 12654 12655 12656 12657 12658 12659 12660 12661 12662 12663 12664 4851 12665
12666 12667 12668 12669 12670 12671 5768 12672 12673 12674 12675 12676 12677 12678 12679 12680 12681 12682 12683
12684 12685 12686 12687 5640 12688 5751 12689 5575 12690 12691 12692 12693 12694 12695 12696 12697 12698 12699
12700 12701 12702 12703 12704 12705 12706 12707 12708 12709 12710 12711 5471 12712 12713 4752 5600 12714 5045
5831 12715 12716 12717 12718 12719 12720 12721 12722 5504 12723 12724 12725 12726 12727 12728 12729 12730 12731
12732 12733 5742 12734 12735 12736 12737 12738 5745 12739 12740 12741 12742 12743 12744 12745 12746 12747 12748
5085 12749 12750 12751 5298 12752 12753 12754 12755 5658 12756 12757 12758 5697 12759 5394 4798 12760 12761
12762 12763 12764 12765 12766 12767 12768 12769 12770 12771 5983 12772 12773 12774 12775 12776 12777 12778 12779
12780 12781 12782 12783 4717 12784 12785 12786 5346 5735 12787 12788 12789 12790 12791 12792 12793 12794 12795
12796 12797 12798 12799 5375 12800 12801 12802 12803 12804 5974 6055 12805 12806 12807 12808 4782 12809 12810
12811 12812 12813 12814 12815 12816 5809 12817 12818 12819 12820 12821 12822 12823 12824 4727 12825 12826 12827
12828 12829 12830 12831 12832 12833 12834 12835 12836 12837 12838 12839 12840 12841 12842 5586 12843 12844 12845
12846 12847 12848 5450 12849 12850 12851 12852 5496 12853 12854 12855 12856 12857 12858 12859 12860 12861 12862
12863 12864 12865 12866 12867 12868 12869 12870 12871 12872 12873 12874 12875 12876 12877 5443 12878 12879 12880
12881 12882 12883 5128 12884 12885 12886 12887 12888 12889 12890 5236 5076 12891 12892 12893 12894 12895 12896
12897 12898 12899 12900 12901 12902 12903 4595 12904 12905 12906 12907 12908 12909 12910 12911 12912 12913 12914
12915 12916 4849 12917 12918 12919 12920 12921 12922 12923 12924 5478 12925 12926 12927 12928 12929 12930 12931
12932 12933 12934 12935 12936 12937 12938 12939 12940 4956 12941 12942 12943 4900 12944 12945 12946 12947 12948
12949 6070 12950 12951 12952 12953 12954 12955 12956 12957 12958 5823 12959 12960 12961 12962 12963 12964 12965
12966 12967 12968 5249 12969 12970 12971 12972 12973 12974 12975 5786 12976 12977 12978 12979 12980 12981 12982
12983 12984 12985 12986 12987 12988 12989 12990 5023 12991 5819 12992 12993 6004 12994 12995 12996 5951 12997
12998 12999 13000 13001 13002 13003 13004 13005 5318 13006 13007 5961 13008 13009 13010 13011 13012 13013 13014
13015 13016 13017 13018 13019 13020 13021 13022 13023 5311 4628 13024 13025 13026 13027 13028 13029 13030 5913
4865 13031 13032 13033 13034 13035 5008 13036 5644 13037 4985 13038 13039 13040 6028 5479 13041 13042 13043
13044 13045 13046 13047 13048 13049 13050 13051 13052 13053 13054 13055 13056 13057 13058 13059 13060 13061 13062
13063 13064 13065 13066 13067 13068 13069 13070 13071 13072 13073 13074 13075 13076 13077 13078 13079 13080 5194
13081 13082 13083 13084 13085 13086 5444 13087 13088 13089 5226 13090 13091 13092 13093 13094 13095 13096 13097
13098 13099 4651 13100 13101 13102 13103 13104 5141 4982 13105 13106 13107 13108 13109 6012 13110 13111 13112
13113 13114 13115 13116 13117 13118 13119 13120 13121 13122 13123 13124 13125 13126 5374 13127 13128 13129 13130
13131 13132 13133 4989 13134 13135 13136 13137 13138 5825 13139 13140 13141 13142 13143 13144 13145 5712 13146
5625 13147 13148 13149 13150 13151 13152 13153 13154 13155 5812 13156 13157 13158 13159 13160 13161 5247 13162
13163 13164 13165 13166 13167 13168 13169 13170 13171 13172 13173 13174 13175 13176 13177 4777 13178 13179 13180
13181 13182 13183 5215 5937 13184 13185 13186 13187 13188 13189 13190 13191 13192 6108 13193 13194 4707 13195
13196 13197 13198 13199 5943 13200 13201 13202 13203 4894 13204 13205 13206 13207 13208 13209 13210 13211 13212
13213 13214 13215 13216 13217 13218 13219 13220 13221 13222 13223 13224 13225 13226 13227 13228 13229 4729 13230
5899 13231 13232 13233 13234 13235 13236 13237 13238 13239 13240 5218 13241 13242 13243 13244 13245 13246 13247
5453 13248 13249 13250 13251 13252 4813 13253 13254 13255 13256 13257 13258 13259 13260 13261 13262 13263 13264
13265 13266 13267 13268 13269 4837 13270 13271 13272 13273 13274 13275 13276 13277 13278 13279 13280 13281 5036
13282 13283 13284 13285 13286 13287 13288 13289 13290 5441 13291 13292 13293 13294 4590 13295 13296 13297 13298
13299 13300 13301 13302 5169 13303 13304 13305 13306 13307 13308 13309 13310 13311 13312 13313 13314 13315 13316

13317 13318 13319 13320 13321 13322 13323 13324 13325 13326 13327 13328 13329 13330 13331 13332 5402 13333 13334
13335 13336 13337 13338 13339 13340 13341 13342 5347 13343 13344 13345 13346 13347 13348 13349 13350 13351 13352
13353 13354 13355 5922 13356 13357 13358 13359 13360 4783 13361 5369 13362 13363 13364 13365 13366 13367 13368
13369 13370 13371 13372 13373 13374 5323 13375 13376 13377 13378 13379 13380 13381 5592 13382 13383 13384 13385
13386 13387 5563 13388 13389 13390 13391 13392 13393 13394 13395 13396 13397 13398 13399 13400 13401 13402 13403
13404 13405 13406 13407 5268 13408 6035 13409 5111 13410 13411 4653 13412 4906 13413 13414 13415 13416 13417
4636 4797 13418 13419 13420 13421 13422 13423 13424 13425 13426 13427 13428 13429 13430 13431 13432 13433 13434
13435 13436 13437 13438 13439 13440 13441 13442 13443 13444 13445 13446 13447 13448 13449 13450 5820 13451 13452
13453 13454 13455 13456 13457 13458 13459 13460 13461 13462 13463 13464 13465 13466 13467 13468 13469 13470 13471
13472 13473 13474 13475 13476 13477 5558 13478 13479 5764 13480 13481 13482 13483 13484 13485 13486 13487 4681
13488 13489 13490 13491 13492 13493 13494 13495 13496 5854 13497 13498 13499 13500 4884 13501 4800 13502 13503
13504 13505 13506 13507 13508 13509 13510 13511 13512 13513 13514 13515 13516 13517 13518 13519 13520 13521 13522
13523 13524 13525 13526 13527 13528 5886 13529 5670 13530 13531 13532 13533 5791 13534 13535 13536 13537 13538
13539 13540 13541 13542 13543 13544 13545 13546 13547 13548 13549 13550 13551 13552 13553 13554 13555 13556 13557
13558 13559 13560 13561 5883 13562 13563 13564 13565 13566 13567 13568 13569 13570 13571 13572 13573 13574 13575
13576 13577 13578 13579 13580 5001 13581 13582 13583 13584 4821 13585 13586 13587 13588 13589 13590 13591 13592
13593 13594 13595 13596 13597 13598 13599 13600 13601 13602 13603 13604 13605 13606 5425 13607 13608 13609 13610
13611 13612 13613 13614 13615 13616 13617 13618 13619 13620 13621 6034 13622 13623 13624 13625 13626 13627 13628
13629 13630 13631 4765 4642 13632 4827 13633 13634 6039 4836 13635 13636 5117 13637 13638 5837 13639 13640
13641 13642 13643 13644 13645 13646 13647 13648 13649 13650 13651 13652 13653 13654 13655 13656 13657 5551 13658
13659 13660 13661 5267 13662 13663 13664 5904 13665 13666 13667 13668 13669 13670 13671 13672 13673 13674 13675
13676 13677 13678 13679 5385 13680 13681 13682 5595 13683 13684 13685 13686 13687 13688 13689 13690 13691 13692
13693 13694 13695 5920 13696 13697 13698 13699 13700 13701 13702 13703 13704 13705 13706 13707 13708 4605 13709
13710 13711 13712 13713 13714 13715 13716 13717 5963 13718 13719 13720 13721 13722 5387 13723 13724 13725 13726
13727 13728 13729 13730 13731 13732 13733 5903 13734 13735 13736 13737 13738 5817 13739 5408 5269 13740 13741
13742 13743 13744 13745 13746 13747 13748 13749 13750 13751 13752 13753 13754 13755 13756 13757 13758 13759 13760
13761 13762 5646 13763 13764 13765 13766 13767 13768 13769 13770 13771 13772 13773 13774 13775 13776 13777 13778
13779 5597 13780 13781 13782 13783 4673 5956 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 4696
13794 13795 13796 5497 13797 13798 4712 5799 13799 13800 13801 13802 13803 13804 13805 13806 13807 13808 13809
13810 13811 13812 13813 13814 13815 13816 13817 13818 13819 13820 13821 13822 13823 13824 13825 13826 13827 13828
13829 5442 4701 13830 4859 13831 6064 13832 13833 13834 13835 13836 13837 13838 13839 13840 13841 13842 13843
13844 13845 13846 13847 13848 13849 13850 13851 13852 13853 13854 13855 13856 13857 5186 13858 13859 13860 13861
13862 13863 13864 5431 13865 13866 13867 13868 5449 13869 4995 13870 13871 13872 13873 13874 13875 13876 5604
13877 13878 13879 5406 13880 13881 13882 13883 13884 13885 13886 13887 13888 13889 13890 13891 4863 13892 13893
5695 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903 13904 13905 5815 13906 13907 13908 13909 13910
13911 13912 13913 13914 5571 13915 13916 13917 5208 13918 13919 13920 5109 13921 13922 13923 13924 13925 13926
13927 13928 13929 13930 13931 13932 13933 13934 13935 13936 13937 13938 13939 13940 5224 13941 13942 13943 13944
13945 13946 13947 5407 13948 13949 13950 13951 5189 4620 13952 4582 13953 13954 13955 13956 13957 13958 13959
13960 13961 13962 13963 13964 13965 13966 13967 13968 13969 13970 13971 5536 13972 13973 13974 13975 13976 13977
13978 13979 13980 13981 13982 5533 13983 13984 13985 5042 13986 13987 13988 13989 13990 13991 5776 13992 13993
13994 13995 4803 13996 13997 5167 13998 4743 13999 14000 14001 14002 14003 14004 14005 14006 14007 14008 14009
14010 14011 14012 14013 14014 14015 14016 14017 5274 14018 14019 14020 14021 14022 14023 14024 14025 14026 5011
14027 14028 14029 14030 5168 14031 14032 5550 14033 14034 14035 5251 14036 14037 14038 14039 14040 14041 14042
5014 14043 14044 14045 14046 14047 14048 14049 4990 14050 14051 14052 5933 14053 14054 14055 14056 14057 14058
5813 14059 14060 14061 14062 14063 5015 14064 14065 14066 5531 14067 14068 14069 14070 14071 14072 14073 14074
14075 14076 14077 14078 14079 14080 14081 14082 14083 14084 14085 14086 14087 14088 14089 5716 14090 14091 5924
14092 14093 14094 14095 14096 5749 14097 14098 14099 14100 14101 14102 14103 14104 14105 14106 14107 14108 14109
14110 14111 14112 4918 5619 14113 14114 14115 5593 14116 14117 5166 14118 14119 14120 14121 14122 14123 14124
14125 14126 14127 14128 14129 14130 14131 14132 14133 14134 14135 14136 14137 14138 14139 14140 14141 14142 14143
14144 14145 14146 14147 14148 14149 14150 14151 14152 14153 14154 14155 14156 14157 14158 14159 14160 14161 14162
14163 14164 14165 14166 5081 14167 14168 14169 14170 14171 14172 4852 5256 14173 14174 14175 14176 14177 14178
14179 4872 14180 14181 4996 14182 14183 14184 14185 14186 14187 14188 5580 14189 14190 14191 14192 14193 14194
14195 14196 14197 14198 14199 5472 5231 14200 14201 14202 14203 14204 14205 14206 14207 14208 14209 14210 5232
4750 14211 14212 14213 14214 5538 14215 14216 14217 14218 14219 14220 5788 14221 14222 14223 14224 4676 14225
14226 14227 14228 14229 14230 14231 14232 14233 14234 14235 4720 14236 14237 14238 4987 14239 14240 14241 14242
4955 14243 14244 14245 14246 14247 14248 14249 14250 14251 14252 14253 14254 14255 5004 14256 14257 14258 14259
14260 14261 14262 5918 14263 14264 14265 14266 14267 14268 14269 14270 5197 14271 14272 5866 14273 14274 14275
14276 14277 4789 14278 14279 14280 14281 14282 14283 14284 14285 14286 14287 14288 14289 14290 14291 14292 14293
14294 14295 14296 14297 14298 14299 14300 14301 14302 5241 14303 14304 14305 14306 14307 14308 14309 14310 14311
14312 14313 14314 6058 14315 14316 14317 5642 14318 14319 14320 14321 14322 14323 14324 14325 14326 14327 6040
14328 14329 5569 14330 14331 14332 14333 14334 14335 14336 14337 14338 4631 5315 14339 14340 14341 14342 14343
14344 14345 14346 14347 14348 14349 14350 14351 14352 14353 14354 14355 14356 14357 14358 14359 14360 14361 14362
14363 14364 5534 14365 14366 4775 14367 5993 14368 14369 14370 14371 14372 14373 6085 14374 14375 14376 5337
14377 5708 14378 14379 14380 14381 14382 14383 14384 14385 6069 14386 14387 14388 14389 14390 14391 14392 14393
14394 14395 14396 14397 14398 14399 14400 14401 14402 14403 14404 14405 14406 14407 14408 14409 5529 14410 14411
14412 14413 14414 14415 14416 14417 14418 14419 14420 14421 14422 14423 14424 14425 14426 14427 14428 5755 5732
14429 5356 14430 14431 14432 5192 14433 14434 14435 5733 14436 5351 14437 14438 14439 14440 14441 14442 14443
14444 14445 14446 14447 5598 4816 14448 14449 14450 14451 5603 14452 14453 14454 5448 14455 14456 14457 14458
14459 14460 14461 14462 14463 14464 14465 14466 14467 14468 14469 14470 5765 5370 4965 4715 14471 5416 14472
14473 14474 14475 14476 4805 14477 14478 14479 14480 14481 14482 14483 14484 14485 14486 14487 14488 14489 5503
14490 14491 14492 5982 14493 4678 14494 14495 14496 14497 14498 14499 14500 14501 14502 14503 14504 14505 14506
14507 14508 5822 14509 14510 14511 14512 14513 14514 14515 14516 5591 5361 14517 14518 14519 14520 14521 14522
14523 14524 14525 14526 14527 14528 14529 14530 14531 14532 14533 14534 14535 14536 14537 14538 14539 14540 5278
14541 14542 14543 14544 14545 14546 14547 14548 14549 14550 14551 14552 14553 14554 14555 14556 14557 14558 14559
14560 14561 14562 14563 4921 14564 14565 14566 14567 14568 14569 14570 14571 14572 14573 14574 14575 4988 14576
14577 14578 14579 14580 14581 14582 14583 14584 14585 5513 14586 14587 14588 14589 14590 14591 5875 5050 14592

14593 14594 14595 14596 14597 14598 14599 14600 14601 14602 14603 14604 5572 14605 14606 14607 14608 14609 14610
5364 14611 14612 4645 14613 14614 14615 14616 14617 14618 14619 14620 14621 14622 14623 14624 14625 14626 14627
14628 14629 14630 14631 14632 14633 14634 14635 14636 14637 14638 14639 14640 14641 14642 14643 14644 14645 14646
14647 14648 14649 14650 14651 14652 14653 14654 14655 14656 14657 14658 14659 14660 14661 14662 14663 5512 14664
14665 14666 14667 14668 5836 14669 14670 14671 14672 5737 14673 14674 14675 14676 4842 14677 14678 14679 14680
14681 14682 4737 14683 14684 4833 14685 14686 14687 14688 5359 14689 14690 14691 14692 14693 14694 14695 14696
4650 14697 5293 14698 14699 14700 14701 14702 14703 5063 14704 5941 14705 14706 14707 14708 5938 14709 14710
14711 14712 5710 14713 14714 14715 14716 14717 14718 6086 14719 14720 14721 14722 14723 14724 14725 14726 14727
14728 14729 14730 14731 14732 14733 14734 14735 14736 14737 14738 14739 14740 14741 14742 14743 14744 14745 14746
14747 14748 14749 14750 14751 14752 14753 14754 14755 14756 14757 14758 14759 5165 14760 14761 14762 14763 14764
14765 14766 6048 14767 14768 14769 6071 14770 14771 14772 14773 14774 14775 14776 14777 14778 14779 14780 14781
14782 6073 14783 14784 14785 14786 14787 14788 6101 14789 14790 4854 14791 14792 14793 14794 14795 14796 14797
14798 4617 14799 5753 14800 14801 14802 14803 14804 14805 14806 14807 14808 14809 14810 14811 14812 14813 14814
14815 14816 14817 14818 14819 14820 14821 14822 14823 14824 14825 4598 14826 14827 5233 14828 14829 14830 14831
14832 14833 14834 14835 14836 14837 14838 14839 14840 5017 14841 14842 14843 14844 14845 14846 14847 14848 14849
5107 14850 14851 14852 14853 14854 14855 14856 14857 14858 14859 14860 14861 14862 14863 14864 14865 14866 14867
5752 5511 5522 14868 14869 14870 14871 14872 14873 14874 14875 5313 14876 14877 14878 14879 14880 14881 14882
4796 14883 14884 14885 14886 14887 4840 14888 14889 14890 14891 14892 14893 14894 14895 14896 14897 14898 14899
14900 14901 14902 14903 14904 14905 14906 14907 14908 14909 14910 14911 14912 14913 5152 14914 14915 14916 4687
14917 14918 14919 14920 4722 14921 14922 14923 14924 5537 14925 14926 14927 14928 14929 14930 14931 14932 14933
14934 14935 14936 14937 14938 14939 14940 14941 14942 14943 14944 14945 14946 14947 14948 14949 14950 14951 14952
14953 14954 14955 14956 14957 14958 14959 4951 4759 14960 14961 14962 14963 14964 5144 5464 14965 14966 14967
14968 14969 14970 14971 14972 14973 14974 14975 6042 14976 14977 14978 14979 14980 14981 14982 14983 14984 14985
14986 14987 14988 14989 6066 14990 14991 14992 14993 14994 14995 14996 14997 14998 14999 15000 15001 15002 15003
15004 15005 15006 15007 15008 15009 15010 15011 15012 15013 15014 15015 15016 15017 5044 15018 15019 15020 15021
15022 15023 5826 15024 15025 15026 15027 15028 15029 15030 15031 5544 15032 5219 15033 15034 15035 15036 5396
15037 15038 15039 15040 15041 15042 15043 15044 15045 15046 15047 4714 15048 15049 15050 15051 15052 15053 15054
15055 4922 15056 15057 15058 15059 15060 15061 15062 15063 15064 15065 15066 15067 5222 4855 15068 15069 15070
15071 4937 15072 15073 4618 15074 15075 15076 5587 15077 6089 15078 15079 15080 15081 5656 15082 15083 15084
15085 15086 15087 15088 15089 15090 5578 15091 15092 15093 15094 15095 15096 15097 4643 5605 15098 15099 15100
15101 15102 15103 15104 15105 15106 15107 15108 15109 15110 15111 15112 15113 6067 15114 15115 15116 15117 15118
15119 15120 15121 15122 15123 15124 15125 15126 5860 15127 15128 15129 15130 15131 15132 15133 15134 15135 15136
15137 15138 5953 15139 15140 15141 15142 15143 15144 15145 15146 5030 15147 15148 15149 15150 15151 5006 5329
5121 15152 15153 15154 15155 4762 15156 15157 15158 5730 15159 5013 15160 15161 15162 4917 15163 15164 15165
15166 15167 15168 15169 15170 15171 15172 15173 15174 15175 15176 15177 15178 15179 15180 6054 15181 15182 15183
15184 15185 15186 15187 15188 5611 15189 15190 15191 15192 5824 15193 15194 15195 5419 15196 15197 15198 15199
15200 15201 15202 15203 15204 15205 15206 15207 15208 15209 15210 4685 15211 15212 15213 15214 15215 15216 5909
15217 15218 15219 15220 15221 15222 15223 15224 15225 15226 15227 15228 5787 15229 15230 15231 15232 15233 15234
15235 15236 15237 5389 15238 15239 15240 15241 15242 15243 15244 15245 15246 15247 4683 4832 15248 15249 15250
5494 15251 5687 15252 15253 15254 5723 15255 15256 15257 15258 15259 15260 15261 15262 15263 4757 15264 4754
15265 15266 15267 15268 15269 15270 15271 15272 15273 15274 15275 15276 15277 15278 15279 15280 15281 15282 15283
15284 4591 15285 15286 15287 15288 15289 15290 15291 15292 15293 6077 15294 15295 15296 15297 4606 15298 15299
15300 5404 15301 15302 15303 15304 15305 15306 15307 4670 15308 6098 15309 15310 4589 15311 15312 15313 15314
6074 15315 15316 15317 15318 15319 15320 15321 15322 15323 15324 15325 15326 15327 15328 15329 15330 15331 4583
15332 15333 15334 15335 15336 5888 15337 15338 15339 15340 15341 15342 15343 15344 15345 15346 5970 15347 15348
15349 15350 15351 15352 15353 5688 15354 15355 15356 15357 15358 15359 15360 15361 15362 15363 15364 15365 15366
15367 15368 15369 15370 15371 15372 15373 15374 15375 15376 15377 5720 15378 15379 15380 15381 15382 15383 15384
15385 15386 15387 5423 15388 15389 15390 15391 15392 15393 15394 15395 15396 15397 15398 4817 5793 4698 15399
15400 15401 15402 15403 5210 15404 5574 15405 15406 5740 4941 15407 15408 15409 15410 6109 15411 15412 15413
15414 15415 15416 15417 15418 15419 15420 15421 15422 15423 15424 15425 15426 5202 15427 15428 15429 15430 15431
15432 15433 15434 15435 15436 15437 15438 5211 15439 15440 15441 5073 15442 15443 5779 15444 5696 15445 15446
15447 4629 5398 15448 15449 15450 15451 15452 15453 15454 15455 15456 15457 15458 15459 15460 15461 15462 15463
15464 15465 15466 15467 15468 15469 15470 15471 15472 15473 15474 15475 15476 4845 15477 15478 15479 15480 5772
15481 5596 15482 15483 15484 15485 15486 15487 15488 5949 15489 15490 15491 15492 15493 15494 15495 6021 15496
15497 15498 15499 15500 15501 15502 15503 15504 15505 5818 15506 15507 15508 15509 5326 15510 15511 15512 15513
15514 15515 4994 15516 15517 15518 15519 15520 15521 15522 5154 15523 4663 15524 15525 15526 5734 15527 15528
15529 15530 15531 15532 5059 15533 15534 4943 15535 15536 15537 15538 15539 15540 15541 15542 15543 15544 15545
5273 15546 15547 15548 15549 15550 15551 15552 4760 15553 5399 15554 6056 15555 15556 15557 5303 15558 15559
15560 15561 15562 15563 15564 5705 15565 15566 15567 15568 5852 5842 15569 15570 15571 15572 5838 15573 15574
15575 15576 15577 15578 5981 15579 15580 15581 15582 15583 15584 15585 15586 15587 15588 15589 15590 15591 15592
15593 15594 5350 15595 15596 15597 15598 5122 15599 15600 15601 4774 15602 15603 15604 15605 15606 15607 15608
15609 15610 15611 15612 15613 15614 15615 15616 15617 15618 15619 15620 15621 5280 15622 5748 15623 15624 15625
15626 15627 15628 15629 5996 15630 5863 15631 15632 15633 15634 15635 15636 15637 15638 15639 15640 15641 15642
15643 15644 5653 15645 15646 15647 15648 4940 15649 15650 15651 15652 15653 15654 15655 15656 15657 15658 15659
15660 15661 15662 15663 15664 5876 15665 15666 15667 15668 15669 15670 15671 15672 15673 5984 15674 15675 15676
15677 15678 15679 15680 5725 15681 15682 15683 15684 15685 15686 15687 15688 15689 15690 15691 15692 15693 15694
15695 15696 15697 5659 15698 15699 15700 15701 15702 15703 15704 15705 15706 15707 15708 15709 15710 5214 15711
15712 15713 15714 15715 15716 4592 15717 15718 5777 15719 15720 15721 15722 5271 15723 15724 15725 15726 15727
15728 15729 15730 4870 15731 15732 15733 15734 15735 15736 15737 15738 15739 15740 15741 15742 15743 15744 15745
15746 15747 15748 15749 15750 15751 15752 15753 15754 15755 15756 15757 15758 15759 15760 15761 15762 15763 15764
15765 15766 15767 15768 15769 15770 15771 15772 15773 15774 4647 15775 15776 15777 4931 15778 15779 15780 15781
15782 15783 4672 15784 15785 15786 15787 15788 15789 15790 15791 15792 15793 15794 15795 15796 15797 15798 15799
15800 15801 15802 15803 15804 15805 15806 15807 15808 15809 15810 4697 15811 15812 15813 15814 15815 15816 15817
15818 15819 15820 5223 15821 15822 15823 15824 15825 15826 4841 15827 15828 15829 5711 15830 4878 15831 15832
15833 15834 15835 15836 15837 15838 15839 4875 15840 15841 15842 15843 15844 15845 15846 15847 15848 15849 15850
15851 15852 15853 15854 15855 15856 15857 15858 15859 15860 15861 15862 15863 4721 4807 15864 15865 15866 15867

15868 15869 15870 5099 5421 15871 15872 15873 15874 15875 15876 15877 15878 15879 15880 15881 15882 15883 15884
15885 4749 15886 15887 15888 15889 15890 15891 15892 15893 15894 15895 4682 15896 15897 15898 15899 15900 15901
15902 5853 15903 15904 5521 15905 15906 5770 15907 15908 15909 15910 15911 15912 15913 15914 15915 15916 15917
5565 15918 15919 15920 15921 15922 15923 15924 15925 15926 15927 15928 15929 15930 15931 5991 15932 15933 15934
15935 15936 15937 15938 15939 15940 5879 15941 15942 15943 15944 15945 15946 15947 15948 15949 15950 15951 15952
15953 15954 4621 15955 15956 15957 15958 4744 15959 15960 15961 5584 15962 15963 15964 15965 15966 15967 4788
15968 15969 5220 15970 15971 15972 15973 15974 15975 15976 15977 15978 15979 5415 5798 15980 15981 15982 15983
15984 15985 15986 15987 15988 15989 15990 15991 15992 15993 15994 15995 15996 15997 15998 5254 15999 16000 16001
16002 16003 16004 16005 16006 16007 16008 16009 16010 16011 16012 5029 16013 16014 16015 16016 16017 16018 4880
16019 5510 16020 16021 16022 16023 5914 16024 16025 5332 16026 5485 16027 16028 5456 16029 16030 5707 16031
16032 16033 16034 16035 16036 16037 16038 16039 16040 5317 16041 16042 16043 4966 16044 6068 16045 16046 16047
16048 16049 16050 16051 16052 16053 5102 16054 16055 16056 16057 16058 16059 16060 16061 16062 16063 16064 16065
16066 16067 16068 16069 16070 16071 16072 16073 16074 16075 16076 16077 5376 16078 16079 16080 16081 16082 16083
16084 16085 16086 16087 16088 5032 5305 16089 16090 16091 16092 16093 16094 16095 16096 16097 16098 16099 16100
5463 16101 16102 16103 16104 5228 4895 16105 16106 16107 16108 16109 16110 16111 16112 16113 16114 16115 16116
16117 16118 16119 16120 4997 16121 16122 4864 16123 16124 16125 16126 16127 16128 16129 5618 5467 6065 16130
16131 16132 16133 16134 16135 16136 16137 16138 16139 16140 16141 16142 16143 16144 16145 16146 16147 16148 16149
16150 16151 16152 16153 16154 16155 16156 16157 16158 16159 16160 16161 16162 16163 5480 16164 16165 16166 16167
16168 16169 16170 16171 16172 16173 16174 16175 5078 16176 16177 16178 16179 16180 16181 16182 16183 16184 16185
16186 5782 16187 6049 16188 16189 16190 16191 16192 5243 16193 16194 16195 16196 16197 4839 16198 16199 16200
16201 16202 16203 16204 16205 16206 16207 16208 16209 16210 16211 4596 16212 16213 16214 16215 16216 16217 16218
16219 16220 16221 16222 16223 5489 16224 16225 16226 5391 16227 16228 16229 16230 16231 16232 16233 16234 16235
16236 16237 16238 16239 16240 16241 16242 4574 5160 5340 16243 16244 16245 16246 16247 5458 16248 16249 5692
16250 16251 16252 16253 16254 16255 16256 16257 16258 16259 5543 16260 16261 5333 16262 16263 16264 16265 16266
16267 16268 16269 16270 5810 16271 16272 16273 16274 16275 16276 16277 16278 16279 16280 16281 16282 16283 16284
16285 16286 16287 16288 16289 16290 16291 16292 16293 16294 16295 16296 16297 16298 16299 4916 5433 5872 16300
16301 16302 16303 16304 16305 16306 16307 16308 16309 16310 16311 16312 16313 6053 16314 16315 6016 16316 16317
16318 16319 16320 16321 16322 5365 16323 16324 4978 16325 16326 16327 16328 16329 16330 16331 16332 5246 4790
16333 16334 5033 16335 16336 16337 16338 16339 16340 16341 5270 16342 16343 16344 16345 16346 16347 16348 16349
16350 16351 16352 5069 16353 5401 16354 16355 16356 5615 16357 16358 16359 16360 4977 16361 16362 16363 16364
16365 16366 16367 16368 16369 16370 16371 16372 16373 16374 16375 16376 16377 16378 16379 16380 16381 16382 16383
16384 5320 16385 16386 4733 16387 16388 5417 16389 16390 16391 16392 16393 16394 16395 16396 16397 16398 16399
16400 16401 16402 16403 16404 16405 16406 16407 4826 16408 16409 4616 16410 16411 16412 16413 16414 16415 16416
16417 16418 16419 16420 5294 16421 5977 16422 5796 16423 16424 16425 16426 16427 16428 16429 16430 16431 5469
16432 16433 16434 16435 16436 16437 16438 16439 16440 16441 16442 16443 16444 16445 16446 16447 16448 16449 16450
5832 16451 16452 16453 16454 16455 16456 16457 16458 16459 16460 16461 16462 16463 16464 16465 5669 16466 16467
16468 16469 16470 16471 16472 16473 6017 16474 16475 16476 16477 16478 5959 16479 16480 16481 16482 16483 16484
16485 16486 16487 16488 16489 5153 16490 16491 16492 16493 16494 16495 16496 4706 16497 16498 16499 16500 16501
5498 16502 16503 16504 16505 5238 16506 16507 16508 16509 16510 16511 16512 16513 16514 16515 16516 16517 16518
16519 16520 16521 16522 16523 16524 16525 16526 16527 16528 16529 16530 16531 16532 16533 6062 16534 16535 16536
16537 16538 5634 16539 16540 16541 16542 5314 16543 16544 16545 16546 16547 16548 5409 16549 16550 5944 4767
16551 16552 16553 16554 16555 5547 16556 16557 16558 16559 16560 4627 16561 16562 16563 5393 16564 16565 16566
16567 16568 16569 16570 16571 16572 16573 16574 16575 16576 16577 16578 16579 16580 16581 16582 16583 5766 5585
16584 16585 5384 5207 16586 16587 16588 16589 16590 16591 16592 16593 5090 16594 16595 16596 16597 5104 16598
16599 16600 16601 16602 16603 16604 16605 16606 16607 16608 16609 16610 16611 16612 16613 16614 16615 6018 16616
16617 16618 16619 16620 4848 16621 16622 16623 16624 16625 16626 16627 16628 16629 16630 16631 16632 16633 16634
16635 16636 16637 16638 16639 16640 16641 16642 16643 16644 16645 16646 16647 16648 5662 16649 16650 16651 16652
16653 16654 16655 16656 16657 16658 16659 16660 16661 16662 16663 16664 16665 16666 16667 16668 16669 4601 16670
16671 16672 16673 16674 16675 16676 16677 16678 16679 16680 16681 16682 16683 16684 16685 16686 16687 16688 16689
16690 16691 16692 16693 16694 5484 16695 5877 16696 16697 16698 16699 16700 16701 16702 16703 16704 16705 16706
16707 16708 16709 16710 16711 16712 16713 16714 16715 16716 16717 16718 5381 16719 16720 16721 16722 16723 16724
16725 16726 16727 16728 16729 16730 16731 16732 16733 6002 16734 16735 16736 16737 16738 16739 5568 16740 16741
16742 5221 16743 16744 16745 16746 16747 16748 16749 4660 16750 16751 16752 16753 16754 16755 16756 16757 16758
5997 16759 16760 16761 16762 16763 5633 16764 16765 6019 16766 16767 16768 5052 16769 16770 16771 16772 16773
16774 16775 16776 16777 16778 16779 16780 16781 16782 16783 16784 16785 16786 16787 16788 16789 16790 16791 16792
16793 16794 16795 16796 16797 16798 16799 16800 16801 16802 16803 16804 16805 16806 16807 16808 16809 16810 16811
16812 16813 16814 16815 16816 16817 16818 16819 16820 16821 16822 16823 5960 16824 16825 16826 16827 16828 16829
16830 6043 16831 16832 5855 16833 16834 16835 16836 16837 5134 16838 16839 16840 16841 16842 16843 16844 16845
4728 16846 16847 16848 16849 16850 16851 16852 16853 16854 16855 6103 16856 16857 16858 5138 16859 16860 5064
16861 16862 16863 16864 16865 16866 16867 16868 16869 16870 16871 4773 16872 16873 16874 16875 16876 16877 16878
6093 16879 16880 16881 16882 16883 16884 16885 16886 4811 16887 16888 16889 16890 16891 16892 16893 16894 16895
16896 16897 16898 16899 16900 16901 16902 16903 16904 16905 5321 6095 16906 16907 16908 16909 16910 16911 16912
16913 16914 16915 5366 16916 4799 16917 16918 16919 16920 16921 16922 16923 16924 16925 16926 5926 16927 16928
5986 16929 16930 16931 16932 6081 16933 16934 16935 16936 16937 16938 16939 16940 16941 16942 16943 16944 16945
16946 16947 16948 5382 16949 16950 16951 16952 16953 16954 16955 16956 16957 16958 16959 16960 16961 16962 16963
16964 16965 16966 16967 16968 16969 16970 16971 16972 16973 16974 16975 16976 16977 16978 16979 16980 16981 16982
16983 16984 16985 5930 16986 6030 16987 16988 16989 16990 16991 16992 16993 16994 5115 16995 16996 16997 16998
16999 17000 17001 17002 17003 17004 4838 17005 17006 17007 17008 17009 17010 17011 17012 17013 17014 17015 17016
17017 17018 17019 17020 17021 5614 17022 17023 17024 17025 17026 17027 17028 17029 17030 17031 17032 17033 17034
17035 17036 17037 17038 17039 17040 17041 5923 17042 17043 17044 17045 17046 17047 17048 17049 5205 17050 5418
17051 17052 5259 17053 17054 17055 17056 17057 17058 17059 17060 17061 17062 17063 17064 17065 17066 17067 17068
17069 17070 17071 17072 17073 17074 17075 17076 17077 17078 17079 17080 17081 17082 17083 17084 17085 17086 17087
17088 17089 17090 17091 17092 17093 17094 17095 17096 17097 17098 17099 4679 17100 17101 17102 17103 17104 17105
17106 17107 17108 17109 17110 17111 17112 17113 17114 17115 17116 17117 17118 17119 5354 5430 17120 17121 17122
17123 17124 17125 17126 17127 4711 17128 17129 17130 17131 17132 17133 17134 17135 17136 17137 17138 17139 17140

17141 17142 5272 17143 17144 17145 17146 17147 17148 17149 17150 17151 17152 17153 17154 17155 17156 17157 17158 17159 6013 17160 17161



(a) ***(b)***

(c)

Figure III.12. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,* ***(c)*** *Histogrammes de l'image chiffrée.*

III.4.1.2. Description de PosESecL2

Comme la clé générée par notre première alternative codant le premier niveau de sécurisation, est de taille variable d'une donnée à une autre et qui est égale à la taille de la donnée en terme d'éléments (caractères ou pixels), telle que : taille de la donnée = n éléments => taille de la clé générée = n nombres ; alors la résistibilité à l'attaque exhaustive est strictement dépendante de la taille de la donnée. C'est pourquoi, nous proposons, ici, une variante de PosESecL1 et que nous l'appelons PosESecL2 (Position based Evolutionary Secure Level 2).

L'évolution vers la solution optimale (donnée cryptée) par PosESecL2 est assurée par application des opérateurs de reproduction (croisement, mutation) et de sélection des meilleurs individus. Dans ce qui suit nous ne présentons que les phases du processus évolutionnaire faisant la différence entre les deux algorithmes PosESecL1 et PosESecL2. Il s'agit, principalement, des phases de codage et de reproduction. Toutefois, le même mécanisme est adopté pour créer la population initiale en permutant aléatoirement le chromosome codant l'image originale.

a. Codage

Dans le but d'augmenter le niveau de sécurité du premier algorithme décrit dans la section précédente, nous présentons, ici, le nouveau codage proposé donnant lieu à un nouvel algorithme de chiffrement : PosESecL2. Ce dernier opère sur le codage des différents éléments de la donnée à chiffrer : dans le cas d'une donnée texte et de même que pour PosESecL1, le codage utilisé est l'hexadécimal, toutefois, chaque quatre bits des 16 bits codant un élément sont considérés comme un gène. Pour une donnée image, chacune des

composantes R, V et B est considérée comme étant un gène. Ainsi, la taille de la nouvelle clé sera plus grande que celle générée par le premier algorithme. Elle est d'une taille quatre fois plus grande dans le cas de manipulation des données texte et de trois fois plus grande dans le cas de manipulation d'une donnée image. Le codage proposé est résumé à travers la figureIII.13.

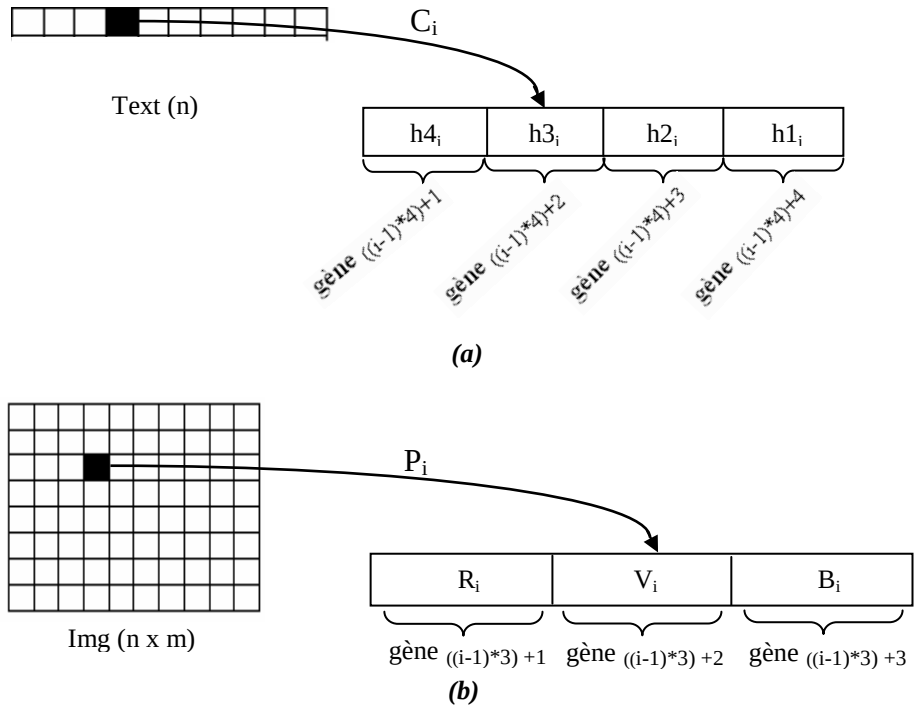


Figure III.13. **(*a*)** *Représentation chromosomale sous PosESecL2 d'un caractère C_i du texte Text(n),* **(*b*)** *Représentation chromosomale sous PosESecL2 d'un pixel P_i de l'image Img(n*m).*

Avec : ($h4_i$ $h3_i$ $h2_i$ $h1_i$) représente le codage Unicode hexadécimal du caractère C_i.

b. Reproduction

Contrairement à l'algorithme PosESecL1, l'algorithme PosESecL2 considère le codage des éléments d'une donnée indépendamment. Ainsi, l'application des opérateurs de croisement et de mutation ne nécessite pas la prise en compte des contraintes à vérifier comme dans le cas de PosESecL1, mais il enrichit d'avantage la population d'individus par de nouvelles caractéristiques.

De même que pour le PosESecL1, l'opérateur de croisement utilisé est le OX « Order Cross-over » appliqué avec un taux fixé expérimentalement et compris entre 60% et 100%. Toutefois, l'opérateur de mutation utilisé est une permutation multipoints pour éviter une convergence trop lente vu la grande différence des tailles des chromosomes dans PosESecL1 et PosESecL2 (le deuxième est d'une taille huit ou trois fois plus grande que le premier). Cet opérateur est appliqué avec un taux, aussi fixé expérimentalement et compris entre 0.1% et 5%.

c. Evaluation des individus

Le choix des individus qui survivront d'une génération à une autre s'effectue selon la fonction objective F, donnée ci-dessous, représentative de l'efficacité des solutions générées sur le problème posé.

$$F(I_i) = \sum_{j=1}^{n \times l} \left| G_{ji} - G_{j0} \right| \quad \text{(III.12)}$$

Avec : G_{ji} est la valeur du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.
l est la taille du codage d'un seul élément de la donnée à chiffrer.

Ainsi, la fonction d'évaluation prendra les deux instanciations suivantes dans le cas de chiffrement de données texte et celui de chiffrement de données images respectivement :

$$F(I_i) = \sum_{j=1}^{n \times 4} \left| G_{ji} - G_{j0} \right| \quad \text{(III.13)}$$

$$F(I_i) = \sum_{j=1}^{n \times 3} \left| G_{ji} - G_{j0} \right| \quad \text{(III.14)}$$

d. Critère d'arrêt

La condition d'arrêt assurant la convergence de l'algorithme PosESecL2 est la même que celle de PosESecL1 puisque le codage des individus dans les deux algorithmes ne diffère que sur le plan contenu de gènes mais les mêmes informations sont présentes dans les deux codages (codage Unicode des caractères du texte à chiffrer ou représentation RVB des pixels de l'image à chiffrer). Autrement dit, c'est la manipulation du codage des éléments de la donnée à chiffrer qui fait la différence entre les deux codages utilisés par les deux algorithmes proposés. Donc, la condition d'arrêt adoptée par ESecL2 dans le cas de chiffrement de données texte et celui de chiffrement de données images, respectivement, est la suivante :

$$0 \le F(I_i) \le F \times 4 \times n \quad / (F)_{16} = (15)_{10} \quad \text{(III.15)}$$

$$0 \le F(I_i) \le 255 \times 3 \times n \quad \text{(III.16)}$$

De même que pour PosESecL1, cette unique condition n'assure pas dans tous les cas la convergence de PosESecL2. C'est pourquoi un nombre maximal de générations sera fixé pour résoudre le problème.

e. Déchiffrement

De même que pour PosESecL1, ce deuxième algorithme utilise le même mécanisme de calcule de clé de chiffrement ; elle change d'un chiffrement à un autre, donc, elle dépend de la donnée à chiffrer et de sa taille. Ainsi, même la taille de la clé de session générée varie d'une donnée à une autre.

f. Réglage des paramètres et résultats

Dans cette section nous présentons les résultats de l'application du deuxième algorithme développé opérant suivant le paramétrage reporté à travers le tableau III.3. À partir des mêmes données originales précédemment utilisées pour tester PosESecL1, nous avons appliqué notre deuxième algorithme PosESecL2 afin d'obtenir les données chiffrées correspondantes. Ces dernières sont celles faisant l'objet des figures III.18.a, III.19.a, III.20.a et III.21.a.

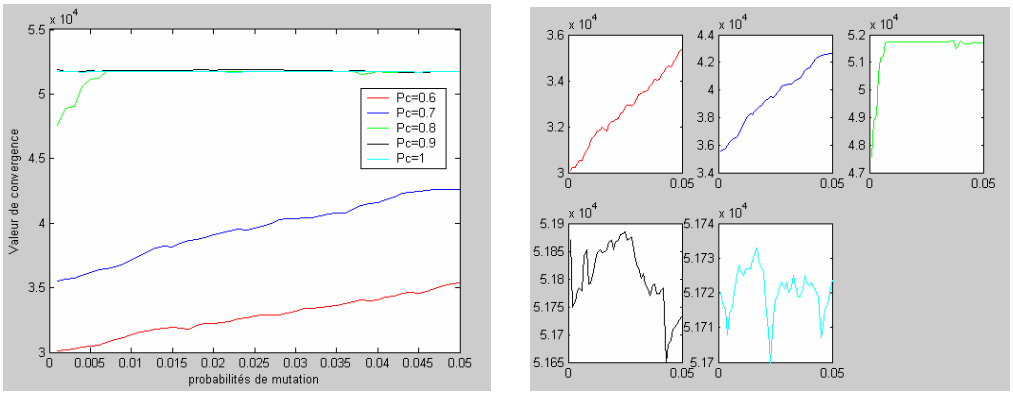


Figure III.14. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

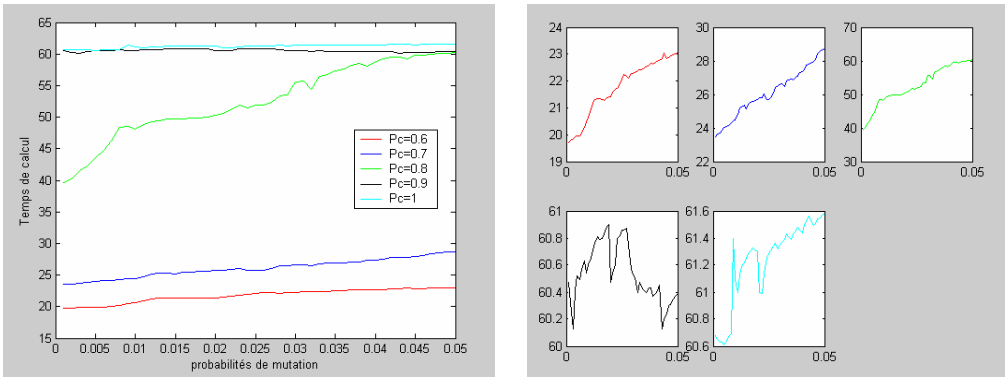


Figure III.15. *Influence des paramètres Pc et Pm sur le temps de calcul.*

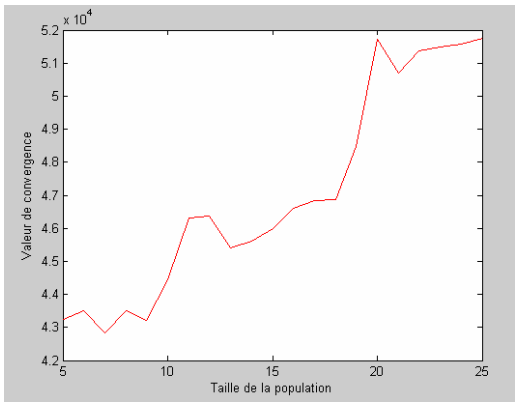


Figure III.16. *Evolution des valeurs de convergence en fonction de la taille de population.*

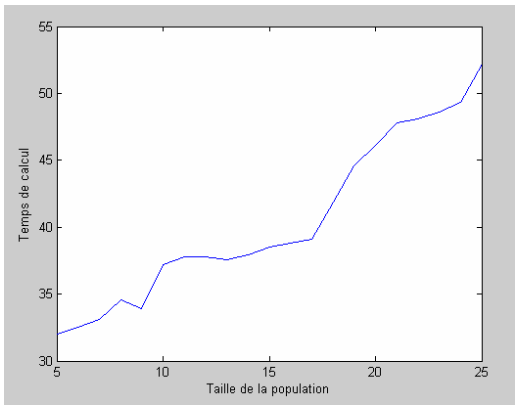


Figure III.17. *Evolution du temps d'exécution en fonction de la taille de population.*

	Valeurs des paramètres de PosESecL2
Taille de la population	20
Nombre de générations	100
P_c	0.8
P_m	0.007

Tableau III.3. *Valeurs adoptées pour les paramètres de PosESecL2.*

L'application de ce deuxième algorithme suivant le paramétrage reporté dans le tableau III.3 sur les mêmes données tests précédentes (Texte 1, Texte 2, Lena et Logo) a permis d'obtenir les résultats résumés à travers le tableau III.4.

D'après ces résultats, il est clair que la taille de la clé augmente proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cette dernière influence la taille des chromosomes résultants du codage utilisé. Donc, c'est la taille des chromosomes augmentant en fonction de la taille des données à chiffrer qui conduit à une augmentation de la taille des clés générées et du temps de calcul correspondant.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	56368	30729	28.74
	Texte 2	1151	59852	35100	31.5
Données images	Lena	131 X 131	823728	51720	46.15
	Logo	420 X 395	9456300	368712	73.06

Tableau III.4. *Résultats obtenus par PosESecL2.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

´1à1њњ╬‹à+1وî¨à+#cЖéZ0¨£1#§£î¨îوغàî+6+¨1وcњ♀њ{à0é6¨à+ض¨éغ+‹0++qو61ëc¨00c¨î{وî8à‹9¨+њZtt01{î1£#à˚0î♀њ1à+╬0î£01↔
↔¨àâîéë¨+#↔↔¨1´éà0ààà+§8¨£§¨¨0´£è#§é1ë+♀++£11¨{é؟↔1£ëàאאַ¨++§+à+¨++9§t´£09£و╬+ët؟é#éà0غњ↔6àא18وא##ë£¨+0
+¨00‹؟+و+9}1¨˚+8؟c8££↔à+11++§#‹îخوZ0؟#ووëé{§1‹+9é86£009وZغ¨c؟}18+î¨cëîé+§א0‹0+غو§£§îњ1˚0+1c+אوàوà+Z#q£18£אخغ♀و06#
î1¨غs‹1غ´î#é0#њZ╬§£é↔è+1+غ8و£+ëà9{њњњ¨éو¨#و£àt¨é+++00é˚#î1+î#6à‹0#0##àé11+؟‹é#אЖZ0#+غ££+↔Ж+à1خén£¨¨à#8}وé↔+¨î
╬90+و+Жë¨î‹§1++↔ëéא؟+î£àé#ëZ11א§#א♀à#î£+1£é+1ët↔c¨خ+ؤëאcوîé#îé؟#؟î1#˚î¨+8#+1+و£éوé╬¨؟1£وZZc8њغ#é10خà#+↔08¨╬1#+
£é0té£+88+#^وא9‹î¨î+א¨{˚£§£+´6§6╬╬↔+¨њњњњà{6+##و↔↔ë؟+îà+16îњ1¨11£1£0#9+غà§+é£+ë+#11؟£ëî‹î+£‹1+´£00£+غ¨+وé#£19
♀♀c§î+c01àî¨8#8و+╬à§¨{î¨+60¨§6♀0Z8é#1#à{+1£‹à9#1؟à6+¨1א£˚6à+1î1ë+؟t9£+îî+80£6é#¨+£¨ZZ1c¨њњ{ëc0♀t0#وtو}+0è↔{ècc8î#
£¨£#وàà£#+1£+њЖњ╬à˚116↔1؟+éîZà£c{´+#

(b)

Figure III.18. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session (Taille$_{\text{Clé}}$ = 6,8808 K octets) :

1	450	496	406	440	458	465	500	437	477	491	298	487	444	468	497	441	471	493
494	454	102	225	455	453	484	473	330	429	432	428	447	469	480	436	488	475	74
446	478	490	443	430	486	483	481	466	499	239	449	489	456	120	464	435	498	451
442	479	474	412	434	467	336	485	5	457	101	213	448	470	431	460	463	438	307
476	445	461	427	459	462	290	495	482	472	433	157	439	426	452	740	1330	545	1331
508	1332	975	1333	564	1334	1335	961	818	1336	886	812	1337	908	674	875	1054	1338	808

725	1339	1340	1341	1342	1343	1344	786	732	1226	1297	228	635	1345	1346	863	696	758	1347
689	1348	1349	1350	710	126	855	548	756	667	522	892	604	1351	1352	602	1353	146	920
770	691	678	729	901	1354	876	1025	1355	4	3	6816	609	993	1356	1357	1229	1358	309
1359	1360	502	1361	706	1362	1363	1364	1365	880	911	644	784	1366	773	359	768	1367	811
299	940	578	198	563	1368	149	957	742	668	687	833	738	1369	623	878	852	636	1370
634	253	1371	873	586	1151	836	620	1372	948	1004	928	1373	799	882	830	270	684	847
964	550	759	1374	870	937	1375	765	905	514	1376	528	739	583	959	337	1377	693	752
612	796	540	1378	552	680	1379	510	946	216	637	630	992	127	978	1309	643	1380	516
1381	559	889	1382	797	567	1383	645	1384	820	949	844	747	1385	866	1386	1387	1160	848
652	384	1388	560	976	985	603	1003	942	501	1389	633	841	974	1093	1390	657	39	912
1391	982	1392	1393	1394	699	750	894	794	1395	1396	932	864	1397	987	716	807	509	676
708	505	1398	838	1399	231	1400	315	339	523	556	326	673	1020	737	599	577	791	751
1329	613	898	1401	1402	419	1001	1403	1404	900	1405	1406	492	1407	536	766	1408	1409	702
787	1410	187	924	1411	845	793	1412	1138	804	1413	1414	1415	546	731	1416	18	1417	66
1418	650	511	662	1288	971	715	817	962	694	1419	968	631	966	537	654	364	520	1104
850	1193	150	981	782	1420	543	915	827	1421	995	1422	843	542	934	1423	1424	775	517
642	709	9	1425	607	772	529	80	994	842	819	712	639	656	2	989	1426	938	610
851	735	666	832	651	541	600	227	241	884	781	895	175	515	417	1237	525	825	734
965	278	890	138	1427	1428	1429	596	1430	553	1293	923	589	730	1431	555	917	1261	663
1432	1433	1000	58	130	1434	790	672	1435	534	1436	1437	1169	44	805	641	800	1438	958
1291	954	1439	1440	1441	1442	835	1443	640	595	1444	933	717	700	888	1445	538	601	616
1446	675	617	701	972	1447	757	909	1448	918	931	1171	704	660	573	998	1449	871	771
1450	763	906	1451	185	1116	557	1452	1453	711	403	916	983	1454	769	1455	221	979	713
1456	846	903	813	839	743	558	910	1295	585	1457	597	1458	1459	1460	967	1461	1462	310
362	313	697	1463	1464	935	719	16	105	856	135	921	952	726	896	1240	1465	705	927
930	1466	397	1467	720	647	980	1468	829	575	950	1469	580	883	664	527	746	323	780
561	614	877	611	828	776	1470	626	1471	570	681	688	569	622	618	879	1472	810	973
572	1473	314	1474	615	683	1475	1476	108	627	1264	665	343	714	382	52	801	1287	152
1477	521	579	28	1478	1479	721	532	1480	1481	549	951	874	46	91	744	592	284	1210
254	1482	206	1483	606	646	945	872	767	320	774	55	348	1484	593	698	535	1485	659
1486	1155	1487	1488	1489	670	1490	990	1491	240	518	815	1492	653	1493	754	899	857	755
798	1494	887	861	1495	143	897	544	996	503	1496	1497	1243	1498	1499	824	1500	1501	1502
533	551	840	977	1503	837	590	1504	988	1505	1506	1507	565	963	788	727	777	1508	1509
809	1510	109	1511	174	1512	869	547	1234	986	904	991	638	1513	999	598	1514	566	722
748	1515	513	802	524	96	1516	1517	1518	919	245	679	507	255	690	1519	733	506	939
922	685	860	1520	853	865	628	504	1521	1522	1523	941	1037	1524	686	1525	355	1526	677
1527	1528	295	914	1189	834	749	661	718	581	1529	764	1530	1531	785	280	823	854	1532
1533	244	859	658	519	648	649	1534	831	984	792	608	554	122	1535	1536	1537	955	723
1538	947	1539	997	1540	568	1541	582	1031	1542	1543	692	1207	588	821	956	671	418	1544
1545	760	1546	943	1547	1548	902	1549	703	1550	1289	695	574	728	745	41	594	970	587
1551	1552	233	1553	1554	655	1555	1556	421	1557	1558	893	562	803	7	814	1559	891	1560
1142	512	1561	629	1562	311	936	73	530	1563	1564	1565	783	1566	1567	1568	137	881	669
624	926	1569	741	415	736	753	953	302	1073	318	1570	779	1571	162	1572	1573	867	266
1574	605	1064	1575	334	90	12	1200	576	1576	1577	969	762	907	1038	619	862	1272	1578
1579	571	822	526	1580	45	49	913	795	1315	761	960	849	944	95	625	1581	1582	201
1583	1005	929	1584	1585	632	531	925	591	1586	789	868	300	682	1587	826	584	885	707
539	1002	806	621	778	724	858	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
1600	1159	1276	1601	1103	1602	1603	1604	1605	1606	1607	128	1608	1609	1610	1611	1612	1613	1614
34	1615	1328	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
1632	1317	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649
1062	1650	1651	1652	1653	1654	1655	1656	1657	1246	1658	1659	1660	1661	1662	1663	1664	1132	1665
1666	1667	349	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683
1684	199	1685	1083	1686	1687	1688	1689	1690	1691	1692	1211	166	1693	1694	1695	1696	1697	291
1247	1153	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1084	1709	1710	1711	140	1712
1713	1714	196	1715	1716	141	1717	1718	1719	1720	1721	1722	1723	1724	1327	287	1725	1726	1009
1727	1728	1729	1730	235	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744
1745	1746	1747	1748	1749	1750	1167	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762
1763	1764	1765	383	57	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779
1780	1144	1781	1185	1782	1783	1784	1294	1785	1786	1324	1787	1156	1788	1789	1790	1791	1792	1793
1195	1281	1794	1795	1796	1797	1798	1799	1800	1014	1801	1802	246	1803	1804	1805	1806	1145	1807
1808	1809	1810	1811	79	1812	1813	1814	1050	1274	286	1815	1816	1817	1818	1819	1820	1821	1822
1823	1824	1825	374	1186	1826	1827	395	1828	1112	1829	1830	112	1831	1832	82	1833	1834	1835
1836	1837	1838	1839	1840	1091	1841	1842	1843	1844	1845	1846	1847	1848	269	1849	1850	17	1275
1851	1177	204	1852	1853	1854	1855	1856	1857	1858	1859	1860	1216	1861	1862	1863	1864	1865	142
1866	1867	1090	1868	1869	1870	56	27	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1039
1881	1882	1883	1884	1075	1885	1886	154	1887	1888	1115	1889	1063	1890	1891	1892	208	1893	1894
1895	1896	1897	1898	1899	1900	1901	1902	1903	1107	1904	1905	1906	1907	1908	1909	1910	1911	1912
1913	1914	1915	1916	125	1917	1918	169	1919	1920	1921	1922	1923	1924	1925	1926	1927	398	1928
1929	1930	1931	1267	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	352	1943	1051	1944
1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	8	1957	1958	1959	1960	1961	1962
293	1963	1964	1965	1966	1967	1968	53	1969	1970	1296	1971	1015	1972	1973	1974	1975	1976	1977
1978	1979	1074	1980	1279	1981	312	1072	1982	1983	1984	1985	1986	264	1987	1988	1989	265	1990
1991	1992	1993	1994	1995	1996	1997	1998	1999	1017	2000	2001	2002	252	2003	2004	2005	342	2006
2007	2008	2009	2010	2011	1130	2012	2013	2014	2015	2016	2017	2018	2019	2020	1041	2021	2022	2023
1058	133	2024	2025	2026	2027	2028	2029	2030	2031	2032	1184	2033	2034	2035	2036	2037	1187	2038
2039	1136	2040	381	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055

1304	305	410	2056	2057	2058	2059	2060	10	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070
2071	2072	2073	2074	35	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088
2089	2090	2091	283	2092	2093	2094	2095	2096	2097	2098	1069	2099	2100	2101	170	2102	2103	2104
2105	2106	2107	2108	2109	2110	1111	2111	297	226	2112	2113	64	2114	2115	267	2116	33	2117
2118	2119	2120	2121	2122	2123	2124	288	2125	2126	2127	2128	2129	2130	2131	176	2132	2133	2134
2135	2136	2137	214	2138	2139	2140	2141	2142	2143	2144	1179	356	2145	2146	2147	2148	2149	2150
2151	2152	2153	2154	2155	2156	2157	2158	2159	99	2160	2161	2162	2163	2164	2165	139	2166	2167
2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	89	2182	2183	2184	153
2185	2186	2187	54	2188	2189	2190	2191	2192	2193	2194	186	2195	1125	2196	2197	2198	2199	2200
1057	2201	2202	2203	2204	2205	261	2206	1158	2207	2208	2209	2210	2211	1087	2212	2213	2214	2215
1273	2216	2217	2218	2219	2220	2221	2222	2223	1230	2224	2225	2226	2227	2228	2229	229	2230	2231
2232	2233	2234	2235	2236	2237	2238	2239	2240	86	2241	2242	1266	2243	2244	103	2245	2246	2247
2248	2249	2250	2251	1235	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	1231	1137	2262	2263
2264	2265	2266	2267	2268	219	2269	2270	376	2271	2272	2273	2274	2275	2276	2277	1113	2278	2279
2280	2281	304	2282	2283	2284	2285	2286	72	2287	2288	2289	2290	2291	394	2292	2293	2294	24
2295	2296	2297	2298	1258	165	2299	1259	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
107	2311	2312	2313	2314	2315	1110	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327
2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	136
2346	1008	1024	2347	2348	2349	2350	2351	2352	2353	404	2354	1232	2355	2356	2357	2358	2359	2360
2361	2362	1102	2363	2364	2365	2366	371	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377
2378	2379	2380	2381	2382	2383	1217	1046	2384	2385	2386	2387	2388	2389	2390	2391	2392	123	2393
2394	2395	379	2396	2397	2398	2399	2400	1203	2401	274	1223	1071	2402	388	2403	2404	2405	1081
2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	1019	2416	2417	2418	2419	2420	2421	2422	2423
2424	2425	2426	2427	2428	2429	2430	2431	2432	40	2433	2434	2435	2436	2437	2438	2439	2440	1162
2441	2442	2443	1109	2444	306	2445	1325	2446	2447	2448	2449	2450	2451	2452	2453	2454	134	2455
2456	2457	76	2458	2459	2460	43	2461	2462	2463	2464	2465	129	2466	2467	2468	2469	2470	2471
399	191	2472	2473	2474	2475	2476	2477	405	2478	2479	2480	2481	2482	2483	2484	2485	2486	2487
2488	294	2489	2490	2491	2492	2493	2494	2495	2496	177	2497	2498	276	247	2499	2500	256	2501
2502	2503	2504	2505	2506	2507	389	2508	2509	2510	2511	2512	2513	2514	2515	2516	1012	346	2517
2518	350	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535
2536	2537	2538	2539	2540	2541	2542	2543	1222	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553
1206	2554	2555	2556	369	2557	2558	2559	292	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569
2570	2571	2572	2573	2574	1028	2575	1205	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586
104	2587	2588	2589	2590	2591	2592	2593	2594	328	2595	2596	1120	2597	2598	2599	2600	2601	2602
2603	2604	2605	2606	2607	316	236	2608	2609	1286	2610	2611	2612	2613	2614	2615	2616	2617	2618
411	2619	2620	2621	2622	2623	2624	2625	2626	1016	2627	365	2628	1214	2629	2630	2631	2632	2633
2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652
2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	19	2664	2665	2666	2667	2668	2669	2670
2671	1176	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688
2689	366	2690	2691	2692	2693	2694	2695	2696	2697	2698	271	2699	172	2700	1114	2701	2702	2703
2704	2705	193	2706	2707	2708	2709	205	2710	2711	1122	2712	2713	2714	2715	2716	1248	2717	2718
1150	2719	2720	2721	2722	2723	2724	189	373	2725	2726	2727	2728	2729	1183	2730	2731	2732	2733
2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	1182	2748	2749	2750	1199
2751	2752	1036	2753	420	2754	2755	145	1148	2756	2757	2758	2759	2760	1194	2761	2762	2763	1106
2764	2765	2766	115	2767	2768	368	2769	2770	163	2771	2772	2773	2774	2775	390	2776	2777	2778
2779	111	113	2780	248	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794
2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	173
2813	2814	2815	168	2816	2817	2818	2819	1061	2820	2821	1249	2822	1094	2823	159	2824	2825	2826
2827	2828	2829	2830	2831	2832	2833	2834	2835	1146	2836	1022	2837	2838	2839	1027	2840	2841	2842
48	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	360	2857	2858	2859
2860	2861	2862	2863	2864	1209	22	178	2865	2866	1181	2867	2868	2869	2870	1085	32	14	2871
1174	2872	2873	2874	2875	2876	1307	1134	1096	2877	1218	2878	2879	2880	2881	2882	2883	2884	2885
2886	329	2887	2888	144	2889	2890	2891	2892	2893	2894	2895	2896	1066	2897	2898	2899	2900	2901
2902	2903	2904	1244	409	2905	400	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917
2918	2919	1170	2920	2921	2922	132	42	2923	2924	2925	2926	2927	2928	2929	2930	77	2931	2932
211	1143	1121	242	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943	1192	2944	2945	422
2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	87	2961	2962	2963
2964	2965	2966	2967	1040	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	322	2978	2979	2980
2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999
3000	3001	1013	3002	3003	37	3004	3005	3006	3007	3008	1300	3009	3010	1250	3011	3012	3013	257
3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029	1166	3030	3031
3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	1262	3043	3044	1313	1149	372	1078	3045
3046	3047	3048	3049	3050	3051	3052	1215	3053	3054	3055	3056	3057	3058	3059	3060	3061	1265	3062
3063	3064	181	3065	3066	3067	268	3068	3069	3070	3071	182	3072	3073	3074	3075	3076	3077	3078
401	3079	3080	3081	3082	161	3083	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	1178
3095	1268	3096	3097	3098	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	1055	3110	3111
3112	3113	3114	1256	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	1204	3125	3126	3127	3128
3129	3130	3131	3132	3133	3134	3135	3136	3137	3138	50	3139	3140	1131	3141	3142	3143	3144	3145
3146	3147	3148	3149	151	3150	3151	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	1320
3163	317	1070	327	3164	3165	1242	3166	3167	1299	402	3168	3169	367	3170	3171	3172	3173	3174
3175	3176	3177	3178	131	3179	94	3180	3181	3182	1154	3183	1047	3184	3185	3186	1168	3187	3188
363	3189	3190	3191	3192	3193	3194	258	3195	3196	121	3197	3198	93	3199	319	3200	1099	3201
3202	3203	202	3204	3205	354	416	3206	3207	3208	3209	3210	3211	1316	3212	3213	3214	3215	3216
3217	3218	3219	3220	3221	303	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	243	3233
1269	3234	1233	3235	3236	3237	85	3238	3239	3240	3241	3242	3243	1319	1213	3244	3245	3246	3247
3248	218	1225	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	1048	3259	3260	119	3261	3262

3263 3264 3265 3266 3267 3268 3269 238 3270 3271 3272 3273 3274 3275 3276 223 3277 1032 3278
3279 1101 3280 3281 3282 3283 1076 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 1260 3294
3295 3296 3297 3298 3299 3300 279 3301 3302 3303 3304 3305 118 3306 3307 3308 1129 3309 3310
3311 97 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328
3329 1135 3330 3331 3332 3333 184 3334 370 3335 3336 224 3337 3338 3339 1191 3340 3341 3342
3343 110 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 1117
1254 3360 3361 3362 3363 3364 3365 188 3366 3367 3368 1157 1147 3369 155 3370 3371 3372 1282
21 3373 3374 3375 3376 1241 3377 3378 425 3379 3380 3381 3382 3383 1190 3384 3385 1033 3386
3387 3388 3389 3390 3391 413 3392 3393 1065 3394 3395 285 1228 3396 3397 3398 3399 3400 3401
3402 3403 3404 194 3405 3406 3407 321 230 1098 3408 3409 3410 3411 3412 3413 197 3414 3415
3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 1251 3428 3429 3430 3431 3432 3433
1152 3434 3435 353 1030 3436 1292 3437 3438 215 1180 3439 3440 3441 3442 3443 3444 3445 3446
3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 164 3457 3458 3459 361 3460 3461 3462 3463
3464 3465 3466 3467 25 3468 3469 331 3470 3471 3472 3473 3474 3475 1270 3476 195 3477 3478
3479 3480 3481 3482 3483 3484 3485 3486 20 340 3487 3488 3489 3490 84 3491 3492 3493 3494
3495 3496 1290 209 1108 1006 3497 3498 3499 3500 3501 190 3502 3503 3504 3505 3506 1018 3507
3508 3509 3510 1224 1053 1124 207 3511 212 3512 3513 3514 3515 1285 114 3516 3517 3518 3519
324 3520 3521 3522 3523 3524 3525 3526 1221 3527 3528 158 3529 3530 3531 3532 3533 1236 3534
335 3535 3536 1172 3537 3538 3539 3540 3541 3542 1035 3543 1318 1119 1220 1011 3544 3545 3546
3547 3548 3549 3550 3551 3552 3553 3554 424 3555 3556 387 3557 3558 3559 3560 3561 3562 3563
3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 156 3574 3575 3576 3577 3578 3579 3580 3581
3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 1118 3592 1303 3593 3594 3595 3596 31 263
3597 3598 3599 3600 3601 3602 3603 69 3604 3605 3606 3607 3608 3609 1133 3610 3611 3612 3613
1196 3614 3615 3616 1219 1173 3617 396 1188 3618 3619 3620 3621 282 3622 3623 3624 3625 3626
3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 1239 1198 3637 3638 3639 408 3640 3641 3642
1049 1089 3643 3644 1306 3645 3646 3647 3648 3649 296 3650 3651 3652 1271 3653 3654 3655 160
3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674
3675 3676 3677 3678 3679 3680 1305 3681 3682 3683 1284 3684 3685 3686 3687 3688 3689 3690 3691
3692 3693 3694 3695 3696 3697 3698 15 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 217
3709 3710 1127 3711 1298 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725
1140 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 1067 3738 3739 3740 3741 180
3742 3743 3744 1023 3745 3746 3747 3748 3749 3750 3751 3752 1161 3753 332 3754 3755 3756 3757
1252 3758 3759 3760 3761 3762 171 3763 3764 3765 3766 3767 3768 100 3769 3770 3771 3772 3773
3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 192 3787 3788 3789 3790 3791
3792 3793 3794 3795 3796 3797 262 1082 333 3798 3799 124 3800 3801 3802 1312 1052 3803 3804
3805 3806 391 3807 3808 1278 3809 1042 83 3810 347 3811 3812 3813 70 3814 3815 3816 3817
3818 3819 3820 3821 3822 3823 3824 3825 1126 3826 3827 3828 3829 3830 3831 3832 3833 3834 98
3835 51 81 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 1255 3846 3847 67 3848 3849
3850 3851 3852 3853 3854 3855 1208 3856 3857 30 3858 3859 3860 3861 65 301 47 1021 3862
3863 3864 3865 3866 3867 3868 29 3869 3870 3871 3872 3873 3874 3875 1253 3876 3877 3878 3879
1321 3880 3881 3882 3883 63 3884 71 3885 3886 3887 3888 275 1175 3889 3890 3891 3892 3893
3894 3895 1045 3896 3897 3898 3899 423 260 3900 3901 3902 308 3903 3904 3905 3906 26 59
3907 3908 3909 1322 3910 3911 3912 3913 3914 3915 1197 1308 3916 3917 3918 3919 3920 3921 3922
3923 3924 1086 3925 3926 3927 3928 1263 68 1128 272 3929 3930 3931 3932 3933 167 3934 3935
3936 3937 3938 3939 3940 3941 3942 147 3943 1034 3944 3945 3946 3947 3948 3949 3950 3951 3952
3953 3954 3955 3956 3957 3958 3959 3960 357 3961 3962 3963 3964 3965 3966 3967 1165 1277 3968
38 3969 3970 232 3971 3972 3973 3974 3975 277 3976 3977 3978 3979 3980 3981 3982 3983 3984
3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003
13 4004 4005 4006 1202 4007 4008 4009 4010 4011 4012 4013 4014 4015 1314 4016 4017 4018 1326
4019 4020 4021 1141 4022 250 4023 4024 4025 4026 4027 4028 4029 4030 1201 4031 1227 4032 386
4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 1095 179 4043 4044 4045 1080 4046 4047 4048
4049 75 1029 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 237 4060 4061 4062 4063 4064
4065 4066 203 4067 4068 1301 414 249 4069 4070 380 4071 4072 1283 4073 4074 4075 78 4076
378 4077 61 4078 4079 4080 4081 4082 4083 4084 88 4085 4086 325 4087 1044 4088 4089 4090
200 4091 377 4092 4093 1164 4094 4095 4096 4097 222 4098 4099 4100 4101 4102 1059 4103 4104
4105 4106 4107 1026 4108 4109 1100 4110 358 4111 4112 1238 4113 1139 4114 4115 4116 4117 4118
4119 4120 1060 1302 4121 4122 4123 4124 4125 4126 4127 344 148 4128 4129 4130 4131 1092 4132
4133 4134 4135 4136 4137 4138 4139 4140 4141 1310 273 4142 4143 4144 4145 4146 4147 4148 4149
4150 4151 1163 116 4152 4153 4154 4155 259 251 4156 4157 4158 4159 4160 4161 4162 407 4163
4164 375 60 4165 4166 4167 1043 4168 4169 4170 4171 36 4172 4173 4174 1105 385 4175 4176
4177 4178 4179 4180 4181 4182 4183 220 4184 1077 4185 4186 4187 1007 4188 4189 4190 106 4191
11 4192 4193 1280 4194 4195 4196 4197 4198 4199 4200 289 92 4201 4202 4203 4204 4205 4206
1311 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223 4224
4225 4226 1097 1123 4227 4228 4229 4230 62 4231 351 4232 4233 4234 4235 4236 4237 4238 4239
4240 1257 4241 4242 4243 4244 4245 4246 4247 4248 1056 4249 183 4250 1245 4251 4252 4253 4254
4255 4256 4257 338 4258 4259 4260 4261 117 341 4262 4263 4264 1079 4265 4266 4267 4268 4269
4270 4271 4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282 4283 4284 4285 4286 4287 4288
4289 4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303 4304 4305 4306 4307
4308 345 4309 4310 234 4311 4312 281 4313 4314 1212 1088 4315 4316 1323 4317 4318 4319 392
210 4320 4321 4322 4323 4324 393 4325 4326 4327 4328 4329 4330 1010 23 4331 4332 4333 4334
1068 4335 4336

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملا أساسيا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردى بعدما كان الأمر مقصوراً فى الماضى على المراسلات الرسمية أو السرية بطبيعة الحال.

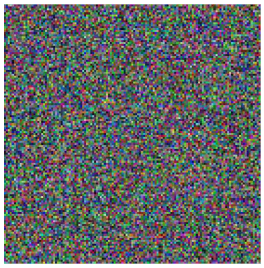
(a)

8ʊ8▒8}¢j^∂ز╔╔ש╔╔j8♂♂®±ĀʊĀ¦88¦(▒}÷╔♂}j¢∂àא`j0ıoز®שʊʊâ♠â&∑▒÷(8♠┤÷┤j∂
(&g8}¦`¦8j(ë÷Ж÷Жjv`♂8±┤¢Æ8^){&ıoıo88j&שÓÖ¦ʊ¢א3ز÷`!ë}ıo¦(ز∑┤¦÷÷3j8®Ā&#j8∂ë8â}╔÷8#8ʊʊʊıoâ:}jj8(&שıoıo8┤j╔8®ĀĀ3¢±
88ë(¦àjא∂}:8¢¢jj¦ç83÷8`ز&∂0&ز®&j÷8▒â┤}Ö&}¢}!j┬jâ3¦¦#`8┤jÖ&Ā8¢â¦±¢j3♠8}ʊ♠Ā3¢`¦ë3±▒▒▒ت&÷j¢ے 8!8┤
∂Æ3÷}jj}¦¢¦&¦8♠♠♠j3ت¦ز¦ıo¦¢8∂8jت÷ʊâ÷^â¢{ز:8÷¢ز∂8¢±¢ıoıoıo╔╔8!¢¦8¢â♂∑∂∑+_¦&زÆ ے {÷`&
8jj(&▒Æ╔╔╔â8┤א8Ā&א÷^±3∑8}{¢ت Ж®¢8¦&♠&¦ت▒0א8±ש∑∑Ā&(&}3¦â3¦®8&â&(÷àj¦(¦ے
ëâ_8ززjj&88&88ت&àj(¦±╔+!}÷ززت÷} Ж&&}ʊجë:8÷╔Ö8!¢ ש}8±8Öے ت╔8{8&8*!ت: { ʊز±&
ے®3ëâ3زشززjʊא0}∂¢j÷&gıoıo¢3┤♠}3♠j8÷âj88زg0¢¢∂ıo∂¦â?ت÷┤▒8÷&j♠+Æ8¢^±ت(`jıo^▒א ▒ز (àت Ā3: ıo&8±ש¢`3ژ88┤ ے ıo3Ж8ے
8:8}±}8زj`88#┤¦&j¦88: Ö3ز8¢¦}∑8&▒زj&8ЖК (¢&ز®
(j88ژ¦ت÷ف}ë÷&8÷&♂¢88ززjh&Æ&÷8¦3&¦88¢8!{¢)¢&8÷ز±±}^!âj«!â¦à8&&!8ز
♠┤&▒8{j8j±¦ש¦j8à÷:8®8ز±∑3j&3ששש8±┤}j}j÷÷+i#∂j¦(3¦ز3┤àĀЖ}3! ש±÷â83&â0¦â3`0&ت▒ے ز ë83شàא∂ıoא∂∂3&8┤Æے
Æ8^â♠^3}&â¦┤┤ز^Öj¦â888ے
â}÷&}&88÷ز¢ز8ز*:zj3¢}j¢زא8&jزıoj}8¢¢ز♠♠8â&┤+3¢8¢ت:âjژ8{+┤`âà÷#8¦`}ز÷âЖ&j:`jà÷83∑Æj÷÷j`â÷}÷*Öj30₪8÷8زà*8¢┤÷ز±÷
3÷¢8«8±┤&j8¦8â÷8╔8¢j88±8ز♠8∂ز3j&♠¦¢ë╔┤÷زâ388àzj¦¦0ت®®÷jâ┤}÷«ɜ3â8}₪8▒&8â3*¦₪₪¢â¦¦â÷¦88¦*8┤ʊ±¦à÷^¦שЖ88∑8ë¦jâĀĀĀ&8
*08#«8ЖÆ8j#¢ʊʊâ3÷!زÖ&┤{{*ÖÖÖשששש∂88ژتj┤÷¦àà: ╔▒▒ש8ز÷¦0Æ♠}j&}∂}jزʊא₪ЖК╔Жâ¦÷ت₪ı┤¦¢ששز}3¢8ت₪±jâ∑!÷38j}¦÷♠

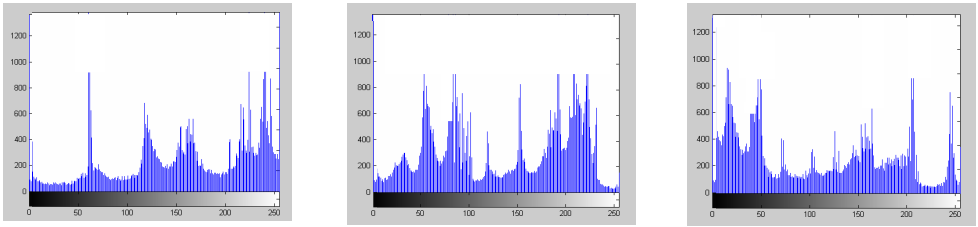
(b)

Figure III.19. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

(a)

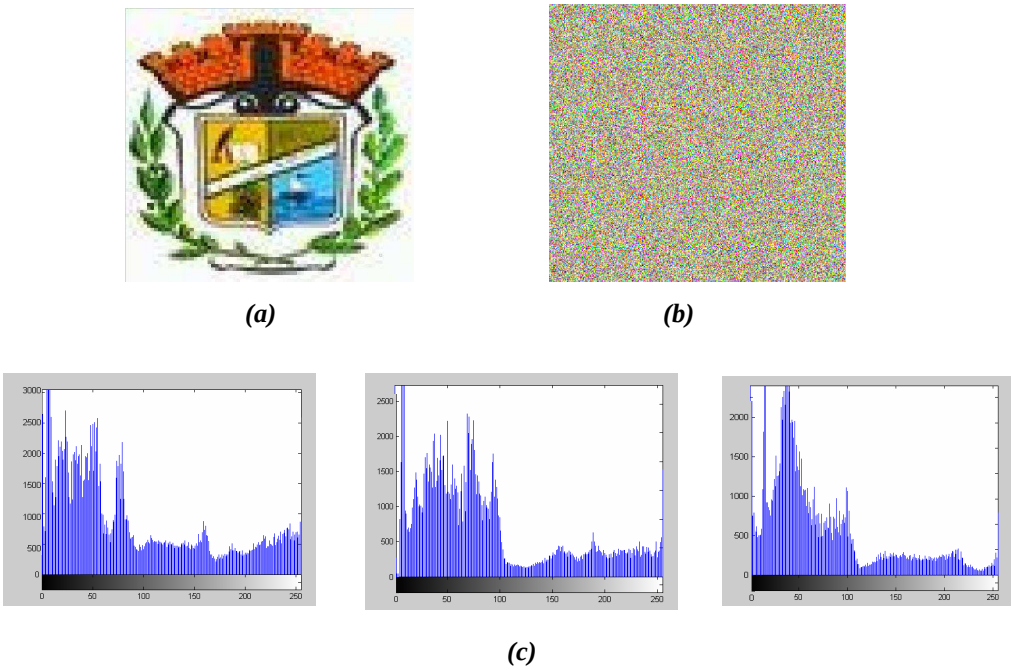


(b)



(c)

Figure III.20. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*



(a) ***(b)***

(c)

Figure III.21. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

III.4.2. Chiffrement à base d'occurrences

Nous avons vu dans la section précédente que le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, dans la présente section nous allons présenter un nouveau codage totalement différent de ceux vu dans les sections précédentes qui ont été bâti autour de la notion de positions de la représentation des éléments constituant la donnée à chiffrer.

Pour appliquer l'approche évolutionnaire à notre problème, nous présenterons les choix utilisés en terme de : représentation chromosomale des solutions du problème, méthode de création de la population initiale des solutions, fonction d'évaluation, opérateurs génétiques et procédure de sélection. Pour mieux approcher le problème, nous présentons une formalisation de ce dernier.

III.4.2.1. Formalisation du problème

Soit D une donnée à chiffrer (texte ou image) qui est une suite de n éléments e_i, et que l'on peut formaliser comme suit :

$$D = \{e_i\}, i \in [1,n] \qquad \text{(III.17)}$$

Dans le cas où D soit une donnée texte, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{C(e_i)\}, i \in [1,n] \qquad \text{(III.18)}$$

où : $C(e_i)$ représente le codage Unicode du caractère e_i.

Dans le cas où *D* soit une donnée image, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{p_i\}, i \in [1,n] \quad \text{(III.19)}$$

Cette représentation (structure) distinguant les différents pixels formant cette image, facilite la conversion en une autre structure en choisissant le RGB (Red Green Blue) comme espace de représentation d'images. Il suffit de remplacer les n pixels par leurs représentations correspondantes en RGB. Nous obtenons, ainsi, la formalisation suivante :

$$D = \{R_i G_i B_i\}, i \in [1,n] \quad \text{(III.20)}$$

Ce mode de représentation (structure) permet de distinguer les différents codages d'éléments de la donnée à chiffrer (les codages des caractères d'une donnée texte, ou les différentes composantes codant les pixels d'une donnée image), et par conséquent de déterminer le nombre d'occurrences de chacun d'eux. C'est ce mécanisme qui sera d'ailleurs exploité pour bâtir notre suivant algorithme de chiffrement proposé.

III.4.2.2. Description d'OEEA (*Occurrences based Evolutionary Encryption Algorithm*)

Comme l'opération de chiffrement consiste à perturber la donnée originale de telle manière à avoir le maximum de désordre dans la donnée chiffrée, nous avons choisi, cette fois-ci, de jouer sur les nombres d'occurrences des éléments (caractères d'un texte ou valeurs des composantes R, G et B codants les pixels d'image), pour avoir la plus grande différence entre les nombres d'occurrences des valeurs similaires d'éléments (nombres d'occurrences des caractères d'un texte original et leurs nombres d'occurrences dans le texte chiffré correspondant ; ou nombres d'occurrences des valeurs des composantes R, G et B codant les pixels d'une image originale et leur nombre d'occurrence dans l'image chiffrée). Ce choix est dû au fait que cette unique donnée (nombres d'occurrences) ne représente pas une information pertinente pour les cryptanalystes.

Dans ce qui suit, nous décrierons les différentes étapes de notre nouvel algorithme de chiffrement proposé.

a. Codage

Les structures (III.18) et (III.20) vues précédemment représentent les données de base à partir desquelles nous sommes arrivés à définir notre codage d'individus en calculant le nombre d'occurrences des valeurs possibles d'éléments dans une certaine donnée. Ainsi, le codage proposé sera le suivant :

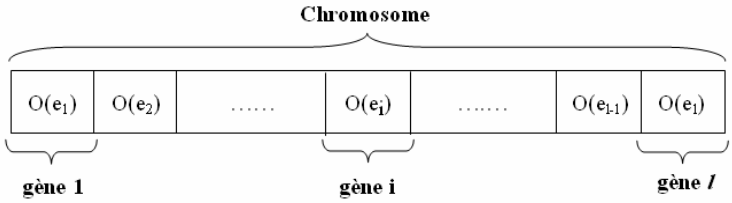


Figure III.22. *Codage des individus sous OEEA.*

Où : O (e_i) est le nombre d'occurrences de l'élément e_i dans la donnée,
l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Remarque :

1) $\sum_{i=1}^{l} O_i = n$ Où n ∈ N représente la taille de la donnée en termes d'éléments (caractères ou pixels).

2) Les éléments qui ne figurent pas dans la donnée auront une occurrence nulle.

Dans le cas de manipulation d'une donnée texte, l'opération de codage consiste à transformer le message en code particulier en calculant le nombre d'occurrence dans le message des 1393 caractères admissibles et l'attribuer à la case qui correspond à son rang dans une table, désignée par TCAR, regroupant les 1393 affichables en Unicode.

Le codage adopté dans ce cas, sera une instanciation du codage général présenté par la figure III.22. C'est celui illustré par le synoptique de la figure III.23.

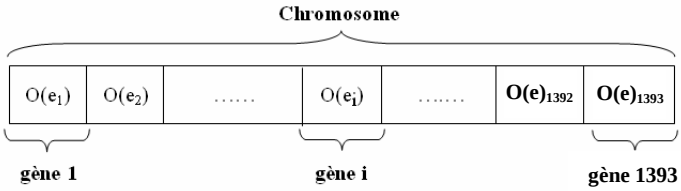


Figure III.23. *Codage des données texte sous OEEA.*

Où $i = \overline{1-1393}$ est le rang des caractères dans la table TCAR et $O(e_i)$ le nombre d'occurrences dans le massage du caractère ayant comme rang i.

Dans le cas de manipulation d'une donnée image et vu que les composantes R_i, G_i et B_i / $i = \overline{1,n}$ prennent leurs valeurs dans l'intervalle [0, 255], notre codage consiste à compter parmi les valeurs de chacune des composantes R, G et B, les nombres d'occurrences relatifs aux valeurs de l'intervalle [0, 255].

Il s'agit de compter à partir des valeurs des n éléments de R, les nombres d'occurrences des valeurs de l'intervalle [0, 255], et de même pour les valeurs des éléments de G et de B. La structure résultante dans ce cas, représente le codage adopté par notre algorithme développé. C'est celle résumée par la représentation chromosomale présentée à travers le synoptique de la figure III.24.

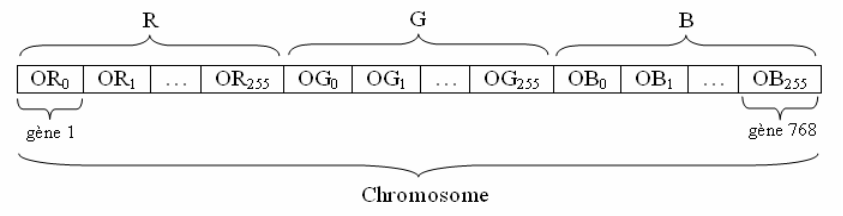


Figure III.24. *Codage des données images sous OEEA.*

Où : OR_i est le nombre d'occurrence des valeurs de la matrice composante R qui égale à *i*,
OG_i est le nombre d'occurrence des valeurs de la matrice composante G qui égale à *i*,
OB_i est le nombre d'occurrence des valeurs de la matrice composante B qui égale à *i*.

Si le codage binaire semble tout à fait adapté pour certains problèmes d'optimisation, cette représentation semble peu appropriée dans notre cas car elle est non intuitive. La représentation qui nous semble la plus naturelle est le codage entier, puisque toutes les opérations de l'algorithme vont manipuler des nombres d'occurrences qui sont des nombres entiers.
Dans ce qui suit, on désigne par individu la donnée à chiffrer ou toute autre donnée, et par chromosome tout individu ayant subi une telle opération de codage.

b. Création de la population initiale

Les m individus (I_1, I_2,…, I_m) formant notre population initiale sont obtenus par application de m perturbations aléatoires sur le chromosome initial I_0 représentant le codage de la donnée originale.

c. Reproduction

Les opérateurs génétiques servent à diversifier la population gérée par l'AE afin d'explorer le plus efficacement possible l'espace de recherche. Nous avons vu dans le chapitre 2 que les AEs disposent généralement de deux opérateurs : l'opérateur de croisement et l'opérateur de mutation.

- **Le croisement :**

Le croisement de solutions peut être toute opération permettant d'obtenir une nouvelle solution à partir de plusieurs individus existants.

Dans le but de compliquer la tâche des cryptanalystes tout en maintenant raisonnable le temps de calcul global de l'algorithme et en évitant toute convergence prématurée, nous avons utilisé un croisement à deux points. C'est le OX « Order Cross-over » proposé par Davis [Davi, 1985]. Les deux points de croisement sont choisis aléatoirement et l'opérateur est appliqué avec un taux qui varie entre 60% et 100% [Gren, 1986].

- **La mutation :**

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

Les individus résultants de ces deux opérations (croisement et mutation) seront rajoutés à la population des parents pour les acheminer vers l'étape suivante qui est celle d'évaluation.

d. Evaluation des individus

La fonction d'évaluation que nous avons définie pour associer des valeurs d'adaptation à chaque individu I_i d'une population Pop, est la suivante :

$$F(I_i) = \sum_{j=1}^{l} \left| O(e_j)_i - O(e_j)_0 \right| \qquad \text{(III.21)}$$

Avec : $I_i = [O(e_1), O(e_2),\ldots, O(e_l)]$, $Pop = \{I_1, I_2,\ldots, I_m\}$ où m est un paramètre de réglage et $O(e_j)_i$ représente le nombre d'occurrences du $j^{ème}$ élément dans le $i^{ème}$ individu.

Dans le cas de manipulation d'une donnée texte, F peut être instancié comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O(e_j)_i - O(e_j)_0 \right| \quad \text{(III.22)}$$

Ou tout simplement :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.23)}$$

Où O_{j_i} est le j^{ème} gène du i^{ème} individu.

Dans le cas de manipulation d'une donnée image, F sera instancié comme suit :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{j_i} - R_{j_0} \right| + \sum_{j=1}^{256} \left| G_{j_i} - G_{j_0} \right| + \sum_{j=1}^{256} \left| B_{j_i} - B_{j_0} \right| \quad \text{(III.24)}$$

Cette fonction est équivalente à celle donnée ci-dessous.

$$F(I_i) = \sum_{j=1}^{768} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.25)}$$

e. Sélection des individus les plus adaptés

La stratégie de sélection de l'algorithme développé va décider de la survie d'une donnée dans la population. Nous avons utilisé la méthode de la sélection par roulette.

f. Critère d'arrêt

Les étapes de reproduction, d'évaluation et de sélection sont répétées jusqu'à ce que l'optimum recherché soit trouvé. Ce dernier représente l'individu ayant la plus grande valeur de convergence durant tout le processus évolutionnaire. Donc, c'est la donnée la plus différente de la donnée originale.

Dans le cas d'une donnée texte, la condition d'arrêt assurant la convergence de notre algorithme évolutionnaire est exprimée à l'aide de la fonction F comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.26)}$$

Preuve :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} (O_{j_i} + O_{j_0}) \quad \text{(III.27)}$$

$$\leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.28)}$$

Et comme :

$$\forall I_i, I_k \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_k} \quad \text{(III.29)}$$

Et :

$$\forall I_i \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.30)}$$

Donc :

$$(III.27) \Rightarrow \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.31)}$$

$$\leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.32)}$$

Dans le cas d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right|$$

Et comme : $((0 \leq R_{ji} \leq n) \,\&\, (0 \leq R_{j0} \leq n)) \Rightarrow (0 \leq \left| R_{ji} - R_{j0} \right| \leq n)$ (III.33)

$((0 \leq V_{ji} \leq n) \,\&\, (0 \leq V_{j0} \leq n)) \Rightarrow (0 \leq \left| V_{ji} - V_{j0} \right| \leq n)$ (III.34)

$((0 \leq B_{ji} \leq n) \,\&\, (0 \leq B_{j0} \leq n)) \Rightarrow (0 \leq \left| B_{ji} - B_{j0} \right| \leq n)$ (III.35)

Alors, d'après (III.33), (III.34) et (III.35) nous aurons :

$$0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq (256 \times n) + (256 \times n) + (256 \times n) \quad \text{(III.36)}$$

$$\text{(III.36)} \Rightarrow 0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n \quad \text{(III.37)}$$

$$\Leftrightarrow 0 \leq F(I_i) \leq 768 \times n \quad \text{(III.38)}$$

Donc, les inéquations (III.26) et (III.38) représentent les conditions d'arrêt mettant fin à notre processus crypto-évolutionnaire dans le cas de manipulation de données texte ou images, respectivement. Nous constatons que $F(I_i)$ est une fonction bornée.

Après quelques tests d'exécution, nous avons constaté que le fait de prendre cette condition uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition d'arrêt ((III.26) ou (III.38)). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous avons arrivé à déterminer ce nombre :

MaxGen = 50

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes dans le cas de chiffrement de texte :

1) $F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i}$

2) MaxGen = 50

Toutefois, elle englobe les deux conditions suivantes dans le cas de chiffrement d'images :

1) $F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n$

2) MaxGen = 50

g. Déchiffrement

Dans cette phase représentant l'opération inverse du chiffrement, une clé de session est générée suivant le même mécanisme utilisé pour nos deux précédents algorithmes. Toutefois, elle est de taille fixe vu que toutes les données texte ont une représentation chromosomale fixe englobant 1393 gènes et toutes les données images ont une représentation chromosomale fixe regroupant 768 gènes.

h. Réglage des paramètres et résultats

Afin de construire un jugement objectif à propos de l'algorithme proposé et de pouvoir noter le maximum de remarques et de comprendre les détails, une large panoplie d'images tests de différentes dimensions a été utilisée. Nous présentons ci-dessous et à titre d'exemple les résultats obtenus pour les mêmes données test précédemment utilisées dans la présentation de nos deux algorithmes développés et précédemment présentés PosESecL1 et PosESecL2 : "Texte 1" de taille 1084 caractères, "Texte 2" de taille 1151 caractères, "Lena" de taille 131X131 pixels et "Logo" de taille 420X395 pixels.

- **Réglage de paramètres**

L'algorithme décrit ci-dessus possède différents paramètres, dont les valeurs influent fortement sur sa qualité. Cette section présente une étude empirique du réglage de ces paramètres afin d'obtenir de bonnes performances, ainsi que le comportement général de l'algorithme qui en résulte.

Les figures III.25 et III.26 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation.

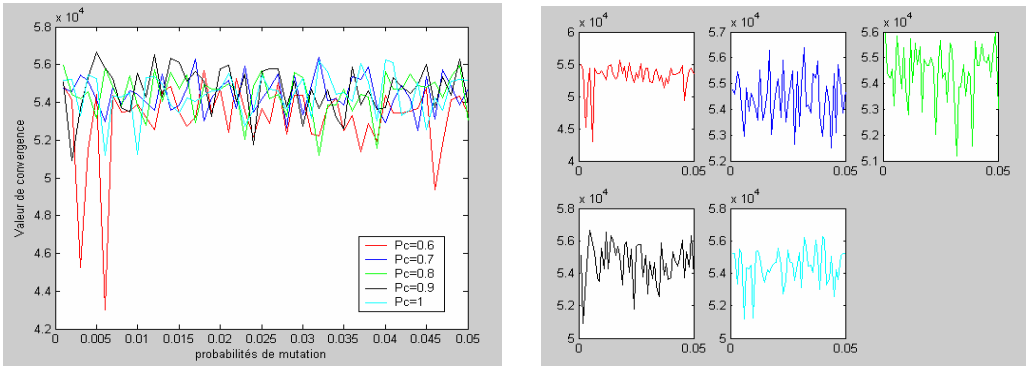


Figure III.25. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

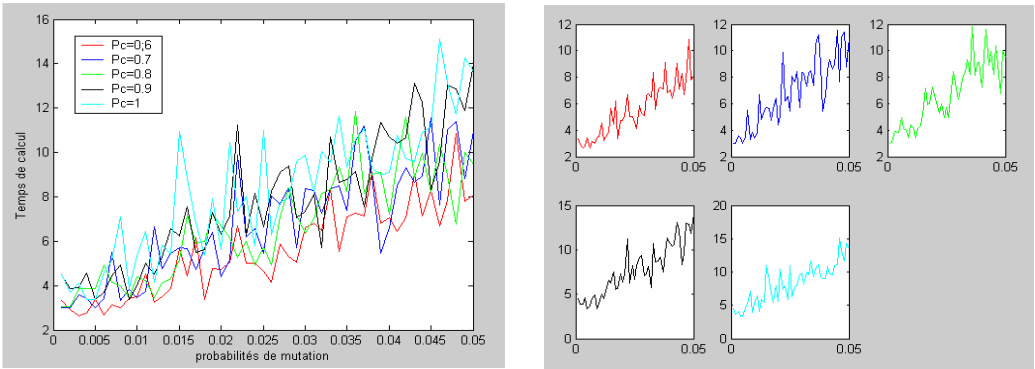


Figure III.26. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Pour choisir les valeurs des paramètres de la probabilité de croisement et celui de la probabilité de mutation, le niveau de confusion n'est pas la seule exigence à satisfaire ; le temps de calcul est aussi important. Ainsi, les valeurs qui nous semblent les plus adaptées sont : *Pc=0.9* et *Pm=0.005* assurant un haut niveau de confusion (VC=56674) en un temps de calcul très raisonnable (T=3.37s).

Après avoir fixé les paramètres de la probabilité de croisement et de mutation, la taille de la population est un autre paramètre à régler expérimentalement aussi. Les valeurs testées sont résumées à travers les figures III.27 et III.28 en indiquant pour chacune d'elles les résultats de chiffrement obtenus en terme de degré de confusion (représenté par la valeur de convergence) et de temps de réponse.

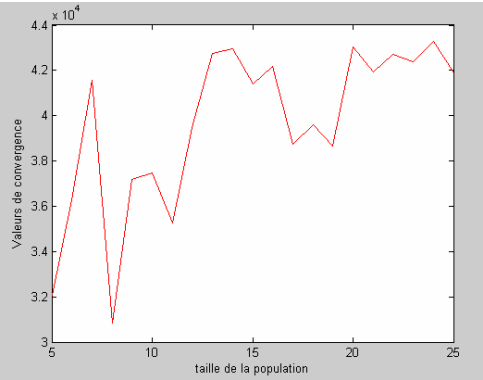


Figure III.27. *Evolution des valeurs de convergence en fonction de la taille de population.*

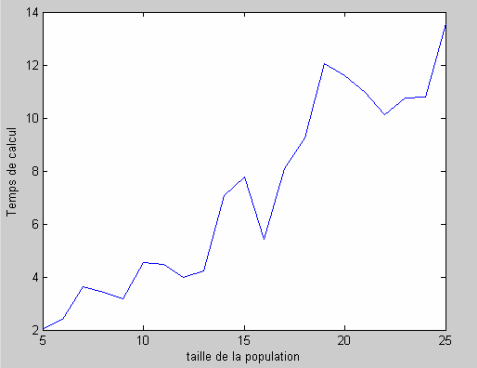


Figure III.28. *Evolution du temps de réponse en fonction de la taille de population.*

D'après la variation de l'influence de la taille de population sur la valeur de convergence et le temps de calcul résumé à travers les deux figures III.27 et III.28, nous optons pour régler le paramètre relatif à la taille de population en lui attribuant la valeur 13 (VC = 42739, T = 4.24s).

Le même principe a été utilisé pour fixer le paramètre relatif au nombre de générations. Donc, et suite à une série de tests sur un ensemble de données soumises au chiffrement et en utilisant la fonction fitness proposée, nous avons retenu certaines valeurs de paramètres qui permettent d'avoir un résultat exploitable en un temps de calcul très raisonnable (de l'ordre de 5 secondes). Ces valeurs sont données par le tableau III.5.

	Valeurs des paramètres de OEEA
Taille de la population	13
Nombre de générations	50
P_c	0.9
P_m	0.005

Tableau III.5. *Valeurs adoptées pour les paramètres d'OEEA.*

- **Résultats**

Après avoir décrit en détail notre méthode nous rapportons, dans cette section, les résultats de l'application de notre troisième algorithme proposé, OEEA, sur les données test citées précédemment (Texte 1, Texte 2, Lena et Logo) en adoptant le paramétrage fixé ci-dessus.

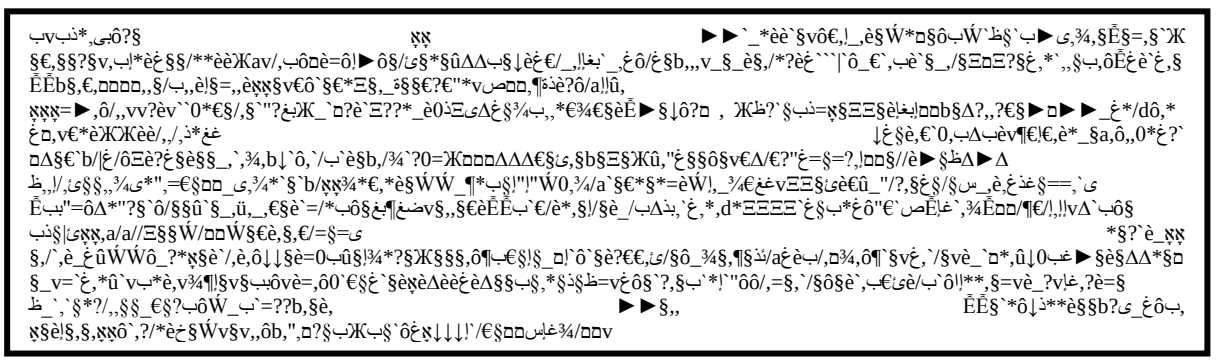
Nous constatons que la taille des clés est indépendante de la taille des données à chiffrer et elle est fixe pour des données de même nature (15323 bits pour les données texte et 7680 pour les données images).

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	15323	2173	6.54
	Texte 2	1151	15323	2284	6.73
Données images	Lena	131 X 131	7680	46034	3.37
	Logo	420 X 395	7680	431340	3.85

Tableau III.6. *Résultats obtenus par OEEA.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)

Figure III.29. ***(a)*** *Donnée originale Texte1,* ***(b)*** *Donnée chiffrée correspondante.*

Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

78 72 75 95 104 96 110 91 108 163 135 87 160 144 129 116 126 83 142
123 102 157 114 82 156 101 164 36 141 103 81 137 151 107 7 111 9 86
100 140 145 153 84 128 133 162 1053 1054 1055 1056 1057 136 120 1089 1090 1091 1092
1093 124 105 80 12 94 134 159 147 127 139 150 155 98 161 90 119 106 154
125 85 152 92 143 148 130 99 132 1281 1282 1283 1284 97 149 131 109 113 93
354 570 347 922 437 923 243 662 465 3 849 213 206 910 924 832 823 456 583
480 925 758 340 568 926 551 369 927 498 791 777 594 638 704 8 394 505 5
789 432 840 690 739 350 813 449 928 796 911 357 525 282 782 604 541 929 810
487 59 698 366 812 462 453 731 403 596 799 930 931 932 225 281 569 933 648
934 820 1386 1387 935 500 476 682 683 921 497 936 261 768 649 904 937 639 447
622 177 938 769 727 496 900 412 200 264 410 914 563 746 901 783 318 642 625
436 778 284 451 459 328 833 43 185 827 186 319 939 592 441 847 502 220 517
455 218 839 603 556 609 286 772 51 850 940 906 664 316 368 209 786 629 589
352 941 878 23 657 273 942 515 869 183 331 790 346 279 861 247 593 943 445
400 558 30 418 344 856 256 595 795 676 624 187 471 944 807 170 757 892 566
64 945 265 779 879 27 269 637 315 168 946 655 607 644 853 742 857 666 564
417 477 947 272 781 305 872 495 806 522 19 580 587 514 640 948 949 876 735
228 920 528 588 950 240 211 1231 1232 1233 915 458 33 287 312 951 545 863 860
571 667 490 952 953 954 888 916 913 299 391 905 671 378 442 669 575 591 726
811 865 202 398 358 399 955 956 314 454 172 341 234 621 546 214 643 737 230
957 223 788 891 259 577 736 958 416 181 263 755 486 50 289 618 959 292 846
864 174 960 519 433 395 434 20 902 961 907 962 1389 1390 1391 1392 1393 275 39
359 964 965 173 966 687 653 337 473 756 700 842 463 329 895 345 818 539 236
787 601 221 793 599 333 919 773 899 504 69 300 701 802 581 58 483 297 967
464 201 721 635 309 255 4 1 800 808 590 605 334 968 548 815 324 969 970
37 203 508 195 835 443 402 870 831 882 65 190 971 250 392 972 307 406 22
401 973 801 387 974 501 237 540 610 222 822 450 798 843 470 765 561 301 439
975 608 482 976 260 351 467 660 885 555 977 552 320 409 267 348 868 373 189
251 205 734 559 14 978 229 431 24 375 659 628 837 612 466 825 979 227 452
619 841 257 178 509 805 980 311 656 871 204 981 623 407 193 1060 1061 1062 1063
754 364 985 385 986 761 520 530 198 288 29 41 485 280 987 1230 1231 1232 1233
1234 290 764 988 989 469 1337 1338 1339 1340 1341 524 507 646 239 627 991 884 526
858 852 681 661 317 1197 1198 1199 1200 1201 826 992 322 699 353 1127 1128 1129 1130
475 993 47 994 887 1140 1141 1142 1143 877 672 70 430 384 684 192 995 527 996
303 997 674 383 828 1049 1050 1051 1052 342 2 326 60 678 491 866 283 26 404
792 1293 1294 1295 650 422 31 886 338 549 68 48 277 484 531 207 585 536 49
1000 705 258 600 651 521 1001 1002 523 438 363 460 15 293 513 381 747 444 550
753 310 1213 1214 1215 1216 1217 1003 898 335 248 245 1004 912 1005 386 274 529 634
630 1006 216 785 503 1257 1258 1259 584 356 620 1007 547 745 271 743 304 617 729
217 285 855 397 330 1008 1009 1010 557 516 875 750 771 355 512 728 296 472 492
1011 361 1012 880 582 809 166 427 446 803 420 494 1013 1371 1372 1373 1378 1379 1380
1381 167 576 784 1014 474 688 733 759 567 468 829 343 1271 1272 670 636 543 626
867 370 598 327 308 606 380 28 658 1015 774 295 844 665 61 560 188 268 732
821 1035 1036 1037 1038 1039 1040 874 371 360 673 663 196 1016 692 184 597 298 1017
614 415 16 1374 1375 723 1376 1377 55 752 715 244 685 215 1018 199 253 848 362
419 212 52 171 616 613 1019 917 425 390 534 1020 918 889 738 897 565 232 270
481 1021 252 488 535 641 396 838 893 797 578 896 766 1022 249 424 1023 1376 1377
1378 1379 388 834 1024 372 224 633 242 794 276 1025 854 631 1026 336 393 775 1356
1357 1358 1359 1360 691 532 675 197 235 499 194 572 730 1027 776 54 414 654 883
862 573 1028 506 349 909 679 586 56 780 873 493 332 428 231 602 377 632 542
824 423 816 53 749 1029 562 554 717 763 11 696 408 894 210 1030 179 702 645
751 804 680 1031 323 718 448 819 457 175 489 325 851 238 1032 306 814 741 374
233 611 461 1033 313 510 668 376 1034 57 748 724 44 1041 1042 1043 1044 1045 695
1046 1277 1278 1279 421 246 294 740 1058 1059 710 1064 1065 1066 1067 1068 719 1069 1070
1071 1072 1073 1074 1075 1076 1077 1078 1079 706 1080 77 73 76 74 79 117 146 1081
1082 1083 1084 63 1085 1086 1087 1088 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104
1105 1106 1107 1108 712 1109 1110 1111 1112 1113 25 169 881 413 382 1114 1115 708 1116
1117 1118 1119 1120 1121 1122 1123 714 1124 1125 1126 615 770 1131 720 1132 1133 1134 1135
1136 1137 1138 1139 18 1144 1145 1146 365 208 440 429 1147 1148 1149 1150 1151 1152 1153
1154 1155 35 1156 1157 1158 1159 1160 1161 903 537 1162 1163 1164 1165 1166 1167 1168 1169
1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 709 1180 1181 982 983 405 984 845 226
1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 66 1202 1203 1204
1205 713 1206 1207 1208 1209 1210 1211 1212 115 138 118 112 122 1218 1219 1220 1221 1222
1223 17 1224 21 1225 1226 1227 1228 1229 998 999 6 836 817 180 1235 1236 1237 1238
1239 1240 1241 1242 711 703 1243 1244 707 722 1245 1246 1247 1248 1047 1048 367 1249 1250
10 62 697 1251 1252 1253 1254 1255 1256 254 389 176 339 191 1260 716 1261 1262 1263
1264 1265 13 1266 1267 1268 1269 34 182 574 1270 1273 1274 693 1275 1276 1280 1285 1286
1287 725 1288 1289 1290 1291 1292 1296 1297 744 278 219 1298 1299 1300 1301 1302 1303 1304
1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 694 1320 32 1321
1322 158 88 89 121 40 1323 1324 1325 1326 1327 1328 1329 45 1330 1331 1332 1333 1334
1335 1336 1342 1343 67 1344 1345 1346 1347 1348 1349 1350 1351 46 1352 1353 42 1354 1355
544 426 291 647 762 262 963 411 1361 1362 1363 1364 1365 990 302 38 579 479 435

1366 1367 1368 1369 1370 538 859 686 241 908 760 533 165 830 511 266 553 1382 1383
1384 1385 518 677 890 1388 689 652 478

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسيًا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال

(a)

gé¥¥§&◄¥c:¥~x┼Ч‌x;>ش╪♫♫⇑¦;>}&◄◄¦¥"<;g§§q5⇑ ¥¥;}¬;§¦<<<î{xcx;<¥¥§xקקt;:g¥¥;ى";<xx¥:§¥§}§;¦;¥¥x;¥<
³x۞۞⇑۞rgt¦¥xrxg◄t¬g¦x4và¦x¥x>§_⇑¦ƒîx¦;<ƒv*§r¥ىx§{¥x¥xqx;⇑ ↔↔îsxî§⇑;¥§<{Ч<j³:&r'קק¥╪xr:
ى۞q¬xƒ>♫¦¥<:q¸↔~®¥xé§g§۞۞;ش;g&x؟♫§xq;
۞¬{î<¥l§;c§¥xî¥§§x{xî۞0>¦◄хњхњхîàњ¦{¥x_:q§Ч§&r¥x¥étsЧЧ{&>¥x;;¦x╪ى;¥x&4§¥:&³³<"îx§ЧЧЧé§xq{¬<ى&{□¥à&x¦¥⇑⇑&>;<
¥î§§¥§ît¥¥rr§;î&;;>g{x¥x¥¥q;<x§;+ ┼┼┼>g§§{¥x;qs;îrxgxƒ¥⇑&x§¥{_î<³¦cx¥ƒ{x>;x۞x5¥⇑xrg;§x&¥¬xx: ◄x§ƒ³{xj¥¥q{î¥;¥j◄5&{g;;
њњq¥¥x&é;g;/>:;l¶‌x³xg¥î;¦;îxx۞۞q<¥:&&♫;¦p¦:/v¥xxxקp¥&§jx;x¥q◄◄¥>;{↔۞î~_::{;x_§xњ&{ﺔ¬¥gƒ¥éî>¥¥§§tl۞;¦x;x,x¶‌¥¦l:g{
qsq¥¥x<¦¶‌Ч;ºx¦>¦x;&q¥:§; ◄xàtgƒ{¦§¥⇑cq¥ggî³¦xî♫x<t{§>;{¥t¬q>¬╪g<§q:>®&_§;<ىr↔xx>¥¦¥Ч‌g<¥àtt~¥éxà5¥ƒ;_<<¥
♫¬xrîx¥§¥¥g~¥¦j¥{x¥↔↔¥¥g{¥§~§¥§¥¥⇑¥;;¥gƒ!lx;:¬;{gњњњx§╪╪xx&¦g↔↔{¥q۞x¥¥xx¥r{t{<®ggî;□۞!gx_ﺔ{_:x{t¦¶‌Чt>◄◄◄
4g¥x<<¥¥q;à¥⇑qg³x¥;¥ƒs<_và¥r§♫xàx;³ §>xƒƒ◄◄x۞tq¥{>îxt⇑x¥¥éxxs;_xىs;{>
קà'xq{ƒ<<קקx¶‌xrgsrv<xњњ4¥;x<x³sg;§ج¦x>¦t♫§§&g{<{;;¥qקx³gњњxƒ¥;ىxéj¥s۞њsss¥¥¥¥קxq_x;¥>¦<§î¥qstc5;sj³³¥s§>;:_ﺔé~g¥¥<x:4<
۞۞{¥ts§<{s§xxrq¥<¥{;:¥¥>&¥;x¦g♫:s;:®¥gsx;¥<v¦s/¬}>ى¥¸x⇑⇑s~sx;w{◄x:ىº¦ig§jx_¶‌Ч&;g_؟>s{&◄۞¥t¦5;s®{{♫⇑¥ش§¥¥>!x
<۞¥x<x;0>¦<>xt;¥î♫{x¥⇑îjs<¦;↔x¦¥><;&xrקקקs¥¬g&<x⇑:x{î³³x:îЧЧ_'{g§◄r§¥;<x/¥<⇑⇑{;¥{{{:□~¥x۞

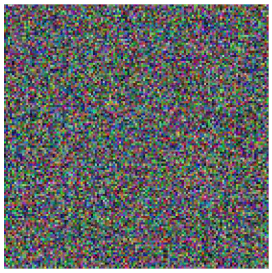
(b)

Figure III.30. **(*a*)** *Donnée originale Texte2,* **(*b*)** *Donnée chiffrée correspondante.*

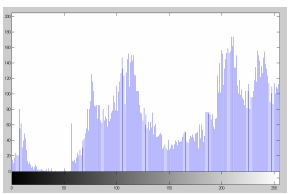
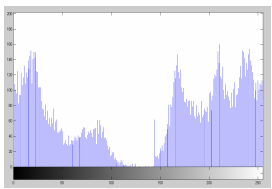
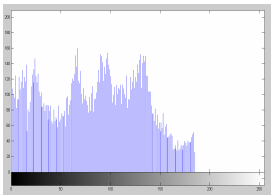
Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

59 1091 1 96 52 8 21 1200 47 54 71 16 43 14 1366 1369 25 64 26
1271 30 32 89 15 65 68 53 28 1303 1305 42 84 81 78 1090 34 24 62
20 58 74 56 39 1117 4 85 99 1324 18 98 1306 29 72 55 66 5 45
49 17 40 23 73 80 97 50 608 102 848 575 286 1220 1223 167 980 869 803
925 1201 134 310 1278 622 866 190 472 727 1118 535 818 1367 114 377 326 835 553
1368 166 488 100 1270 874 206 981 307 232 873 880 676 538 712 578 982 305 860
1325 899 933 842 332 865 315 713 239 445 619 203 939 983 785 192 1302 163 749
872 1327 297 605 513 722 270 940 269 814 198 943 201 570 195 422 746 897 802
402 904 945 250 794 560 691 1276 217 693 514 1304 852 935 804 666 518 1222 133
467 915 156 910 407 1221 170 257 738 542 692 912 984 508 443 245 815 671 372
63 61 67 985 1277 174 128 477 930 639 986 800 987 667 258 988 456 309 989
817 856 688 607 559 520 459 577 188 439 369 325 643 314 172 363 796 592 312
105 193 124 157 148 582 669 822 566 390 779 806 397 1187 1188 573 849 901 917
813 149 317 285 696 180 337 725 824 634 1057 223 990 829 554 953 466 732 242
733 905 647 308 168 101 487 958 1326 793 572 812 454 517 449 991 831 366 540
704 868 356 212 913 847 321 408 1059 280 623 909 484 271 137 662 220 433 324
881 1375 841 1333 896 837 624 992 891 1334 649 724 581 161 502 1275 545 685 924
843 1058 473 585 1374 828 833 275 993 382 450 176 107 1285 375 427 291 929 795
299 362 599 1283 863 178 736 515 531 937 331 292 994 742 888 1332 313 596 126
381 648 359 411 489 431 376 234 120 934 1284 175 714 522 955 534 247 614 598
652 1378 995 143 410 601 996 997 1273 453 883 158 208 719 298 430 689 446 462
357 104 840 928 184 413 303 657 364 399 870 256 396 636 424 159 391 1384 1385
115 1319 702 277 279 419 241 329 1000 734 698 627 469 871 521 690 602 262 140
342 687 1376 1001 464 887 350 116 268 1272 1274 895 720 465 267 197 1002 588 541
468 361 808 132 204 141 1377 287 444 547 438 150 761 300 221 425 921 1320 348
403 737 611 496 304 862 171 586 709 792 820 834 127 406 274 229 145 194 1003
886 546 392 922 288 673 731 448 205 276 561 263 1064 956 625 1004 186 227 492
1005 442 254 593 1006 1007 395 584 260 902 1065 530 968 590 334 389 1008 558 708
877 393 412 475 678 646 526 1203 1204 273 533 832 470 243 1009 579 884 474 451
786 850 908 723 1066 405 501 1010 1011 117 677 1012 498 658 1382 699 765 819 1013
214 1014 1015 233 423 135 1016 650 164 504 615 721 222 457 900 892 911 455 954
1017 642 1018 351 151 838 138 500 1381 505 1019 437 458 226 436 385 580 946 1020
179 131 400 653 173 1101 306 481 213 218 240 686 707 106 626 380 1021 511 358
333 146 1063 620 674 610 111 811 567 1022 185 507 603 878 338 706 1023 283 482
1100 266 379 323 799 926 432 1098 906 893 951 845 1024 388 668 568 718 855 764
110 701 949 224 112 864 756 551 125 920 680 1206 209 142 409 1025 207 612 844
670 633 587 367 645 302 130 591 1312 805 383 621 1026 169 679 103 349 1027 660
655 781 571 544 861 641 730 480 1028 322 1029 628 705 255 682 493 1030 1031 162

890 211 919 1099 478 927 1032 661 355 1205 768 932 1033 853 1067 236 1034 340 681
556 527 801 537 486 182 327 613 189 606 604 703 265 384 1035 617 231 1193 246
165 942 1036 728 640 821 875 200 597 659 261 717 441 415 854 1037 108 1192 569
823 374 281 894 539 923 807 452 202 154 1359 483 440 177 882 485 248 562 319
495 160 387 695 259 697 122 876 401 1190 651 576 414 879 1038 230 966 635 426
519 916 600 950 210 352 118 751 237 918 563 386 963 282 630 656 754 365 797
1182 1183 609 716 295 740 339 476 370 1039 330 378 675 1055 228 885 557 851 113
715 272 594 741 747 826 238 858 672 574 109 497 1191 447 152 353 121 663 494
760 532 499 129 898 404 1040 583 1184 524 565 638 344 294 416 253 644 629 119
938 284 429 1056 320 360 479 555 798 346 825 183 710 947 1041 936 1080 318 595
249 632 827 199 743 1180 336 637 683 435 421 191 729 684 944 810 460 512 509
503 857 1042 418 1094 739 654 1043 1044 525 529 839 1093 809 510 1181 664 771 244
394 311 434 859 1095 931 368 589 316 123 941 957 948 181 139 354 1045 1046 776
784 1047 1048 1049 1050 830 398 264 335 1051 1052 775 1053 1054 420 543 75 76 1060
979 1061 1062 1068 1069 1070 1071 1072 762 1073 1074 196 506 1075 1076 1077 969 1078 1079
11 9 1123 1124 1081 1082 293 373 1083 1084 973 965 1085 1086 296 564 1087 1088 1089
83 36 90 77 6 92 3 91 10 95 33 1092 371 219 1096 1097 726 251 1102
252 903 1103 1104 1105 1106 1107 147 889 552 1108 1109 774 1110 1111 1112 790 1113 773
1114 735 289 1115 1116 37 69 1119 1120 1121 1122 1125 788 975 1126 1127 461 780 1128
960 1129 1130 341 215 1131 1132 1133 971 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143
1144 1145 1146 976 1147 1148 1149 1150 769 1392 1393 1151 1152 1153 962 1154 1155 1156 225
694 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 216 914 153 1347
301 1171 1172 1173 1174 1175 752 1176 1177 1311 290 836 1178 1179 1185 1186 490 471 550
1189 1353 1194 766 1195 782 1196 1313 1197 1198 1199 665 428 1202 907 616 763 1207 977
1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 767 379 321 1224 758 1225 1226
1227 1228 1229 523 278 1230 1234 1235 770 1236 1237 1238 1239 748 1240 1241 1242 22 31
44 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1310 1254 1255 967 1256 1257 1258
1259 1260 753 964 1261 1262 816 1265 1266 618 516 1267 1268 1269 789 1279 1280 13 12
48 41 1281 1282 783 463 1286 1287 1288 1289 1290 959 1291 1292 1293 1294 70 7 1295
1296 1297 548 187 1298 1299 1300 1301 136 536 1307 1308 1309 970 744 1314 1315 757 777
1316 1317 1318 791 1321 1322 1323 27 35 1328 1329 144 345 952 1330 1331 961 1335 1336
1337 51 79 1338 1339 1340 711 235 1341 1342 1343 974 1344 745 1345 1263 1346 94 86
1348 1349 1350 867 155 1351 1352 528 750 1354 998 328 999 1355 1356 60 88 1357 82
46 38 87 57 1358 772 1360 1361 1362 343 549 491 417 1363 1364 1365 1370 1371 767
1372 1373 347 700 1379 1380 778 631 1383 759 846 1264 1386 1387 93 2 1388 1389 755
1390 978 787 972 1391

(a) ***(b)***

(c)

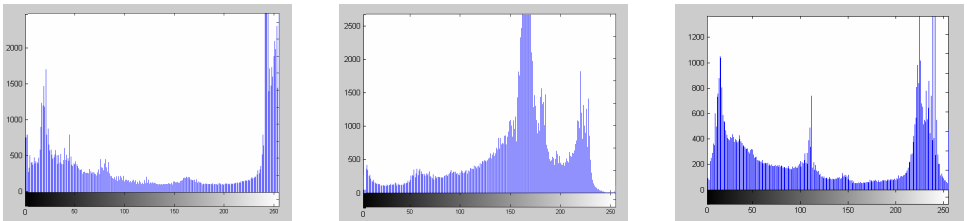
Figure III.31. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 0,9375 K octets) :

369 423 529 171 397 412 474 15 689 64 112 390 267 21 118 304 6 28 574
465 356 338 717 638 339 110 243 458 508 450 361 608 635 115 593 749 151 742
575 716 428 109 486 584 294 722 521 141 87 695 373 225 452 602 679 13 208
85 468 634 709 298 70 297 514 601 439 359 320 251 222 292 319 765 456 614
142 645 625 288 383 217 125 476 718 274 236 729 483 567 114 487 492 303 542

426	628	675	713	296	505	370	730	409	395	764	641	396	669	460	337	264	230	204
623	358	293	418	150	520	134	328	469	762	440	533	651	551	535	286	438	462	103
355	347	63	158	16	561	205	410	643	703	432	519	693	247	620	346	40	736	366
129	652	164	327	371	506	363	154	241	447	434	624	585	163	748	376	380	144	407
59	143	667	252	10	536	548	766	360	335	455	311	497	200	466	249	751	35	119
165	341	36	741	398	448	656	18	65	233	734	17	531	512	595	660	104	586	81
435	299	101	92	309	454	436	710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579	352	739	698	306	253	577	248
735	515	342	285	61	219	138	568	705	658	427	511	532	210	357	307	153	633	711
98	433	619	691	60	20	137	733	485	324	659	419	719	684	269	392	754	68	259
284	467	295	48	300	224	541	600	715	630	613	172	66	425	558	604	38	692	34
557	263	258	384	51	317	375	111	325	146	701	472	388	607	289	94	644	377	745
580	500	598	496	240	443	594	364	554	74	637	187	712	655	747	576	145	47	501
417	490	622	162	491	626	578	699	750	14	128	493	282	226	459	351	589	209	582
545	740	596	147	275	121	362	470	53	193	291	333	290	179	523	527	42	30	402
738	525	113	374	664	646	507	609	685	756	189	700	148	260	654	464	191	350	378
176	257	538	599	603	11	702	372	571	237	12	96	170	192	653	88	416	540	25
180	522	642	166	72	246	194	707	680	33	639	159	215	75	728	227	516	681	344
90	99	559	127	513	265	7	23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555	69	708	345	494	271	238	726
393	534	132	385	587	686	647	368	203	3	481	482	332	199	167	687	560	539	495
743	44	583	91	610	690	336	31	261	403	379	255	725	498	323	386	198	572	220
484	73	322	677	606	340	272	343	58	503	381	26	421	517	761	79	676	334	348
760	49	663	235	597	566	313	627	39	414	618	518	479	55	752	186	694	563	649
175	223	133	549	420	277	510	727	305	391	228	632	404	196	206	84	52	184	755
212	581	763	429	107	697	724	590	77	737	445	71	682	502	509	120	254	413	758
640	149	616	43	195	242	239	211	611	453	757	218	86	19	310	387	2	105	117
424	244	504	116	449	591	8	229	605	399	629	182	57	461	314	588	177	234	190
102	106	714	122	617	612	216	673	312	431	326	621	80	382	721	93	45	156	665
553	767	475	174	329	394	415	547	188	331	543	365	308	46	67	670	330	157	354
302	389	480	367	232	768	27	207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631	41	471	131	201	245	54	202
444	556	544	720	270	32	688	759	442	279	478	136	197	123	273	671	437	262	457
672	268	349	95	50	488	706	546	183	221	178	160	528	696	62	9	214	615	573
489	430	97	744	22	753	524	446	213	746	731	181	231	283	661	570	401	276	78
400	4	562	526	683	301	89	477											

(a) ***(b)***



(c)

Figure III.32. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session (Taille$_{Clé}$ = 0,9375 K octets) :

503	463	515	233	412	71	181	502	153	365	302	449	225	362	500	33	523	198	519
471	404	521	497	145	496	504	494	205	138	505	6	522	214	149	451	422	498	146
520	63	453	320	516	510	350	287	495	39	387	385	20	115	80	361	472	69	513
213	204	105	425	68	311	507	506	208	508	24	41	499	511	237	54	415	509	37
518	517	52	458	384	493	390	8	191	78	501	40	163	514	229	512	99	574	372
579	351	396	36	389	17	370	312	626	436	5	1	284	118	629	334	594	625	174
704	648	28	534	469	657	342	411	479	401	45	533	234	675	379	23	235	231	223
77	324	729	634	548	217	555	142	673	159	346	568	551	85	195	343	333	408	158
10	563	532	373	264	73	242	359	179	280	473	709	605	128	327	82	756	308	638
588	152	31	766	272	285	166	667	714	51	613	157	439	265	561	271	314	277	491
286	32	125	2	332	194	547	552	241	156	443	661	143	455	431	120	215	484	578
434	526	647	344	257	371	744	168	199	584	260	707	486	763	369	244	524	298	540
689	405	764	196	595	538	345	478	636	464	95	216	29	567	46	735	57	331	397
752	426	377	288	133	114	134	62	424	668	720	399	200	336	696	459	141	553	173
702	169	148	171	460	622	210	35	409	201	570	209	444	14	624	558	299	483	66
366	184	224	600	100	671	738	293	677	549	192	193	291	554	230	76	481	639	67
124	545	620	226	93	529	303	136	164	470	603	150	687	419	304	683	414	581	74
565	127	245	144	577	337	760	475	557	705	301	49	418	137	44	110	7	79	306
207	13	635	380	276	758	420	278	248	238	292	630	457	47	437	583	445	375	723
354	492	684	255	679	81	263	119	221	61	682	751	669	652	104	681	659	290	92
121	454	188	172	564	355	754	708	489	251	394	220	12	60	97	126	566	604	403
601	15	395	429	330	715	645	546	490	759	108	154	423	34	441	87	741	575	643
666	101	19	693	686	718	111	631	326	542	617	674	706	84	178	543	381	50	167
733	402	716	654	462	614	239	300	254	725	262	619	232	428	305	593	335	161	427
83	250	341	180	374	632	465	406	719	183	378	623	382	243	147	740	765	296	649
658	736	582	252	165	712	745	89	413	569	160	106	633	329	177	608	340	461	589
537	261	456	38	103	607	640	627	98	4	30	597	416	281	596	599	536	253	734
26	742	477	642	266	438	135	386	573	295	339	721	711	190	487	753	591	539	162
737	610	726	313	140	197	70	16	525	621	637	86	749	182	572	11	474	139	612
730	347	598	43	556	541	319	48	761	187	664	748	275	646	672	258	151	219	710
259	256	236	348	450	609	109	325	186	571	580	678	703	435	430	694	606	376	274
240	328	206	393	175	466	615	315	587	75	383	602	616	527	25	273	762	448	697
113	452	132	155	724	364	739	294	485	728	202	750	476	650	544	116	480	688	42
338	530	352	123	743	247	641	676	368	211	176	107	467	698	322	55	560	56	731
685	270	170	767	227	655	690	279	310	746	644	203	732	680	663	662	417	64	528
651	433	691	96	59	246	585	757	112	318	283	618	628	22	297	665	592	535	94
65	58	222	102	586	699	3	695	358	228	717	212	392	576	531	289	130	88	356
388	316	122	768	660	421	446	185	611	268	670	410	398	468	249	321	755	72	9
18	692	700	117	323	722	447	307	353	360	131	90	407	129	363	562	432	488	590
27	367	400	559	349	317	391	440	656	653	727	218	442	482	282	713	53	747	91
269	309	189	267	21	701	550	357											

III.5. Discussion et évaluation des résultats

Un bon crypto-système doit satisfaire les six points clés de Kerckhoffs [Kerc, 1883] et ceux de Shannon [Shan, 1949]. Plus explicitement, la qualité d'un crypto-système se mesure principalement par sa résistibilité face aux différents types d'attaques. La technique de cryptage proposée à travers nos trois algorithmes proposés PosESecL1, PosESecL2 et OEEA et qui consiste en une perturbation évolutionnaire de la donnée originale, assure une bonne sécurisation contre les attaques les plus destructives. Les critères utilisés pour évaluer leurs sécurités sont les suivants : la vitesse de l'algorithme, l'attaque statistique, l'attaque différentielle, l'attaque exhaustive et l'analyse de l'espace des clefs.

III.5.1 Vitesse des algorithmes

En plus du paramétrage, la vitesse d'un algorithme évolutionnaire dépend étroitement du codage utilisé. Un mauvais codage affecte non seulement la qualité de l'algorithme en termes d'optimalité de solutions mais aussi en termes de temps de convergence. Dans le cas de nos deux premiers algorithmes proposés, PosESecL1 et PosESecL2, le codage dépend de la taille de la donnée à chiffrer ce qui affectera la vitesse des algorithmes puisque le temps de traitement des données de grandes tailles sera beaucoup plus grand que celui de traitement des données de petites tailles, chose due à la taille des chromosomes manipulés durant les générations de l'évolution. Ceci se voit clairement à travers les résultats obtenus par ces deux algorithmes où le temps de convergence de PosESecL1 est meilleurs que celui de PosESecL2

du fait que la taille des chromosomes codant une donnée pour le PosESecL2 est trois fois plus grande que celle des chromosomes codant la même donnée pour le PosESecL1. Toutefois, le codage proposé à travers OEEA unifie la taille des chromosomes codant des données de même nature (textes ou images) et de différentes tailles. Ainsi, le temps de traitement est indépendant de la taille des données traitées. En plus et d'après les résultats expérimentaux présentés précédemment, nous constatons que le temps de convergence de ce dernier algorithme est très raisonnable (de l'ordre de 4 secondes).

Comparons, maintenant, la vitesse de nos trois algorithmes développés avec celle des principaux crypto-systèmes (AES du coté des crypto-systèmes symétrique, RSA du coté des crypto-systèmes asymétrique et SEC qui est un algorithme de chiffrement exploitant les algorithmes génétiques). Le tableau 7 et la figure 33 présentent un résumé des temps de chiffrement et de déchiffrement de chacun des algorithmes cités.

	PosESecL1	**PosESecL2**	**OEEA**	**AES**	**RSA**	**SEC**
Temps de chiffrement (s)	24.5	46.15	3.37	0.5	42.8	1.8
Temps de déchiffrement (s)	0.26	0.75	0.12	0.08	42.5	0.06

Tableau III.7. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

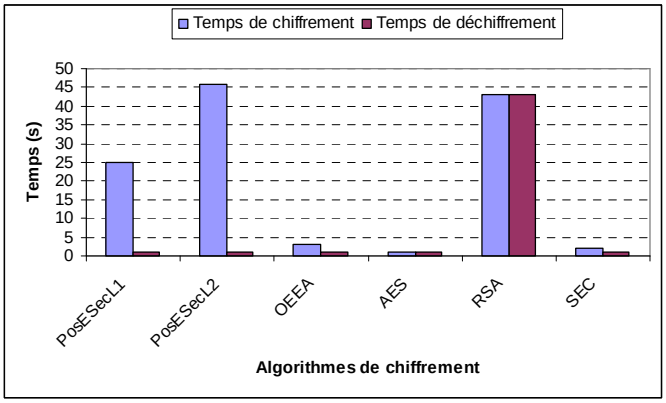


Figure III.33. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

D'après le tableau et la figure, ci-dessus, présentant le temps de chiffrement et celui de déchiffrement de nos algorithmes en comparaison des algorithmes SEC [Omar, 2007], RSA [Mene, 1996] et AES [www4], nous remarquons que :

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement beaucoup plus petit que celui de RSA. De son coté PosESecL1 possède aussi un temps de chiffrement/déchiffrement meilleur que celui de RSA. Toutefois, PosESecL2 a un temps de chiffrement de l'ordre de celui de RSA et un temps de déchiffrement meilleur que celui de RSA.
- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin de celui d'AES. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes PosESecL1 et PosESecL2 sont plus grands que celui d'AES.

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin que celui de SEC. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes sont plus grands que celui de SEC.

De même et d'après les résultats expérimentaux obtenus et présentés précédemment, nous constatons que le temps de calcul de OEEA est indépendant de la taille de la donnée à chiffrer du fait de l'unicité de taille de codage de toutes les données de même nature soumises au chiffrement. Toutefois, ce n'est pas le cas pour PosESecL1 et PosESecL2 où le temps de calcul des deux algorithmes est strictement dépendant de la taille de la donnée à chiffrer. Ainsi, OEEA est l'algorithme ayant un temps moyen de calcul le plus adapté.

III.5.2 Niveau de confusion

La confusion est l'une des notions essentielles énoncées par *Shannon* sur lesquelles se base un algorithme de cryptage. Elle sert à cacher la relation entre la donnée en clair (originale) et la donnée chiffrée correspondante. Donc, elle vise à rendre la donnée aussi peu lisible que possible.

Dans notre cas, les deux modes de chiffrement développés, à savoir le chiffrement à base de positions et le chiffrement à base d'occurrences, sont de niveaux de confusion différents.

Pour le premier, les deux algorithmes proposés dans ce mode sont aussi de niveaux de confusion différents. En effet, le niveau de confusion de PosESecL2 est meilleur que celui de PosESecL1 du fait que PosESecL2 exploite les positions des éléments codant les différents composants de la donnée, considéré chacun comme unité séparée (gène) du codage final de la donnée (chromosome). Ainsi, la donnée chiffrée aura de grande chance de contenir de nouveaux éléments ou composants ne figurant pas dans la donnée originale. Cette chose complique considérablement, voire même, pénalise toute cryptanalyse possible comme nous allons le démontrer dans les sections qui suivent. Toutefois, PosESecL1 est de niveau de confusion strictement dépendant de la taille de la donnée à chiffrer puisqu'il exploite les positions des éléments de la donnée. Donc dans le cas de données de petites ou de moyennes tailles, une attaque exhaustive finira de trouver la bonne combinaison initiale de positions et, ainsi de casser l'algorithme.

Pour le deuxième mode de chiffrement développé, il offre un très bon niveau de confusion grâce à l'utilisation d'un codage bâti autour de la notion de nombres d'occurrences. Ce mode ne présente ni information utile pouvant être exploitée par un cryptanalyse ni possibilité de reproduction de la donnée en se basant uniquement sur cette donnée surtout que le codage adopté assure que les composants de la donnée (caractères ou pixels) changent dans la donnée chiffrée par rapport à la donnée originale. En plus, le niveau de confusion de OEEA, l'algorithme développé dans ce mode, est meilleur que celui de PosESecL2 et ainsi de celui de PosESecL1 du fait que l'espace des nouveaux caractères qui peuvent apparaître dans une donnée chiffrée par rapport à la donnée originale correspondante est beaucoup plus grand dans OEEA que dans PosESecL2.

III.5.3 Attaque statistique

Ce type d'attaque considère le crypto-système comme une boite noire, il analyse statistiquement les entrées et les sorties de ce système. Pour ce faire, nous avons utilisé les mesures suivantes dans le but d'évaluer ou de quantifier la différence entre l'image originale et l'image cryptée correspondante :

Le facteur *NPCR* (*Number of Pixels Change Rate*) donné par l'expression (III.39), l'erreur absolue moyenne (*MAE : Mean Absolute Error*) donnée par l'expression (III.41) et l'erreur

quadratique moyenne (*MSE :Mean Square Error*) donnée par l'expression (III.42). Le tableau 8 résume les valeurs des différentes mesures obtenues après les tests qui ont été effectués sur l'image Lena et l'image chiffrée correspondante obtenue dans le cas de nos différents algorithmes proposés.

	NPCR	**MAE**	**MSE**
PosESecL1	0.6314	0.91	0.58
PosESecL2	0.9073	0.97	0.61
OEEA	0.9955	1.03	0.75

Tableau III.8. *Niveaux de confusion.*

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \tag{III.39}$$

$$D(i,j) = \begin{cases} 0 Si\ \mathrm{Im}_O(i,j) = \mathrm{Im}_C(i,j) \\ 1 Si\ \mathrm{Im}_O(i,j) \neq \mathrm{Im}_C(i,j) \end{cases} \tag{III.40}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \tag{III.41}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \tag{III.42}$$

D'après les résultats d'application des mesures quantifiant la différence entre l'image originale et ses versions chiffrées calculées par nos algorithmes, il est clair que la majorité des pixels formant l'image originale ont changé soit de positions (dans le cas d'application de PosESecL1) ou de contenu (dans le cas d'application de PosESecL2 ou de OEEA) pour, ainsi, calculer l'image chiffrée. En effet cela se voit surtout à travers les valeurs de NPCR qui calcul le niveau de différence entre les deux images où les résultats obtenus sont proches de la valeur optimale qui vaut un (01). Toutefois, ces valeurs ne sont pas les mêmes pour les trois algorithmes, chose qui est tout à fait normale vu que les algorithmes sont de niveau de confusion différent (voir section précédente). Et comme nos algorithmes présentent l'avantage d'être non déterministes, donc chaque nouvelle application d'un même algorithme donnera lieu à des résultats différents ce qui entraîne que l'application des mesures NPCR, MAE et MSE donnera lieu, de son tour, à de nouvelles valeurs. Ainsi, aucune propriété statistique ne peut être déduite d'une donnée chiffrée et l'attaque statistique ne possédera aucune chance de casser nos algorithmes.

III.5.4 Attaque différentielle

Cette attaque essaye de tirer des conclusions sur le fonctionnement d'un algorithme de chiffrement en comparant des versions chiffrées de plusieurs données originales et dans un meilleur cas des versions chiffrées de plusieurs blocs formant une même donnée. Ainsi, dans le cas des algorithmes de chiffrement par blocs (DES [Stin, 1996], [Biha, 1991], [Mats, 1994], 3DES [Stin, 1996], [Gant, 2001], AES [Lepr, 2000], etc), quand la donnée (surtout pour le cas d'images) contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement, ce qui fait que la donnée cryptée contiendra des zones texturées ; mais vu que nos algorithmes traitent la donnée en une seule passe, c'est-à-dire ils opèrent sur la donnée totale et non pas des blocs de la donnée ; alors la cryptanalyse

différentielle sera mise à l'écart. De ce fait aussi, nos algorithmes présenteront une certaine robustesse au bruit par opposition aux algorithmes de chiffrement par blocs, et qui se traduit par le fait qu'une erreur sur un bit chiffré ne va pas propager des erreurs importantes dans tout le bloc courant et par la suite dans toute la donnée lors de la recombinaison des blocs.

Même si le cryptanalyste pense à essayer de tirer des observations à partir des résultats de cryptage de plusieurs données pour essayer d'imiter le fonctionnement des algorithmes, l'approche évolutionnaire pseudo-aléatoire complique considérablement la tâche de ce type d'attaque. Une sensibilité à la donnée originale est aussi un point avantageux des algorithmes proposés vu que la donnée cryptée correspondante à une certaine donnée originale change d'une instanciation du problème à une autre. Autrement dit, nos algorithmes développés sont des algorithmes non déterministes à l'inverse du RSA, par exemple, qui a nécessité au préalable de randomiser les données soumises au chiffrement pour éviter les attaques exploitant les modèles connus des données en clairs [Ster, 2004].

III.5.5 Attaque exhaustive

L'attaque exhaustive ou encore dite attaque à force brute est une attaque qui demeure fatale pour les crypto-systèmes utilisant des clés de petites tailles. À ce stade, nos trois algorithmes proposés présentent des niveaux de sécurité différents. La robustesse de PosESecL1 est strictement dépendante de la taille de la donnée originale. PosESecL2 présente un niveau de sécurité plus élevé que celui de PosESecL1 du fait que sa clé soit de taille quatre fois plus élevée que celle de PosESecL1 dans le cas de manipulation de données textes et de trois fois plus grande dans le cas de manipulation des données images. Ainsi, et vu qu'à l'heure actuelle la taille de la clé assurant une résistibilité face à une attaque exhaustive est de 512 bits, alors le cryptage par PosESecL1 de toute donnée possédant une taille inférieure à 512/le nombre de bits nécessaires pour coder la taille de la donnée (nombre de caractères ou de pixels), sera exposée au risque d'une attaque réussite par force brute. C'est le cas aussi pour les données cryptées par PosESecL2 et qui sont de taille inférieure à : 4*(512/le nombre de bits nécessaires pour coder la taille de la donnée texte) ou de 3*(512/le nombre de bits nécessaires pour coder la taille de la donnée image). Toutefois, OEEA ne peut en aucun cas être cassé par une telle attaque vue que la taille de la clé utilisée est largement sécuritaire. Ainsi, OEEA est l'algorithme le plus sûr parmi nos trois algorithmes développés.

Considérons le texte à chiffrer « **there is only one God and Mohamed his prophet** », nous reportons dans le tableau suivant (tableau 9) les tailles de clés de chiffrement de nos trois algorithmes développés en comparaison de DES, 3DES, AES et SEC, ainsi que le nombre d'opérations nécessaires pour générer toutes les combinaisons de bits formant les clés pour chaque algorithme. Ceci donnera une idée sur la complexité de l'attaque exhaustive appliquée contre chacun des algorithmes où, il se voit clairement que l'attaque exhaustive menée contre nos algorithmes est la plus complexe.

	Taille de clé (bits)	**Nombre d'opérations**
DES	56	2^{56}
3DES	128	2^{128}
AES	256	2^{256}
SEC	240	2^{240}
PosESecL1	352	2^{352}
PosESecL2	1408	2^{1408}
OEEA	1393	2^{1393}

Tableau III.9. *Complexité de l'attaque exhaustive.*

III.5.6 Analyse de l'espace des clés

Ici, nous testons la sensibilité du processus cryptographique proposé aux clés utilisées. Dans notre cas, la clé générée est une clé calculée à partir de la donnée originale et de la donnée chiffrée correspondante, donc elle change d'une instanciation du problème à une autre. Ainsi, le cryptage successif d'une même donnée donne lieu à un ensemble de données chiffrées différentes ce qui entraînera, à chaque fois, la génération d'une clé de session différente pour le chiffrement d'une même donnée originale. Cela dotera notre méthode de cryptage d'une très grande sensibilité aux clés puisque toute clé interceptée d'une manière illégale ne servira que pour le déchiffrement d'une seule version chiffrée d'une même donnée donc elle ne sera plus utile par la suite.

Le problème rencontré dans le cas des crypto-systèmes symétriques, et par conséquent dans le cas de nos algorithmes développés qui sont des algorithmes symétriques, est la communication de la clé secrète au destinataire en vue de l'utiliser dans l'extraction de l'information originale (le déchiffrement). Pour ce faire, certains concepteurs ont proposé de combiner les fonctionnalités de la cryptographie symétrique et asymétrique. C'est le cas du PGP par exemple. Il crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ; puis, il crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire. Ainsi, le même principe peut être adopté pour sécuriser le transfert de nos clés en leurs chiffrant par un crypto-système asymétrique. Dans ce cas un schéma hybride de transmission sécurisée peut être envisagé comme le montre la figure suivante :

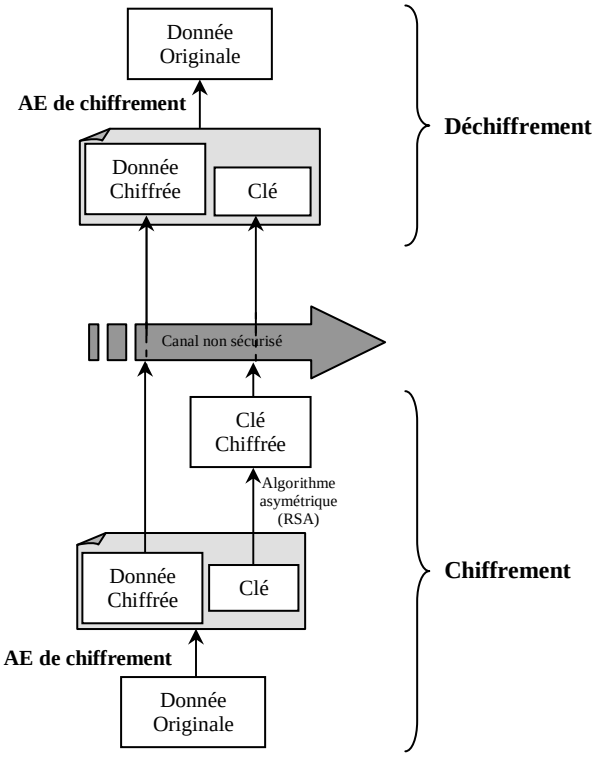


Figure III.34. *Schéma hybride de transmission sécurisée.*

Or, pour nos algorithmes PosESecL1 et PosESecL2, il est clair que la taille de la clé dépend strictement de la taille de la donnée. Le problème qui se pose lors du chiffrement d'une donnée de grande taille, c'est que la taille des clés générées elle aussi devient grande ce qui consommera un grand temps de chiffrement de clés du fait que les crypto-systèmes publiques sont déjà lents. Le problème s'aggrave de plus en plus lorsque la taille de la clé

devient égale ou dépasse la taille du codage de la donnée en termes de bits. Dans ce cas, l'utilisation de ces deux algorithmes devient inutile et il sera plus intéressant d'utiliser un crypto-système publique pour chiffrer la donnée au lieu de la clé générée malgré que la sécurité de tels crypto-systèmes soit grandement affectée par les choix paramétriques (p et q dans le cas de RSA [Zimm, 2005]). Pour remédier à cette anomalie, nous proposons de coder (ou bien comprimer) la clé générée par un système de codage sans perte, tel que le codage de Huffman par exemple [Huff, 1952], en vue de réduire sa taille avant de la chiffrée par un crypto-système asymétrique, puis d'envoyer au destinataire le package englobant la donnée chiffrée et la clé codée chiffrée. Lors de la réception, on procède inversement. On déchiffre la clé codée en utilisant la clé publique du destinataire, puis on la décode pour obtenir la clé proprement générée par PosESecL1 ou PosESecL2 qui va permettre de reproduire la donnée originale. Ainsi, le schéma du processus de sécurisation englobant cette dernière fonctionnalité sera le suivant :

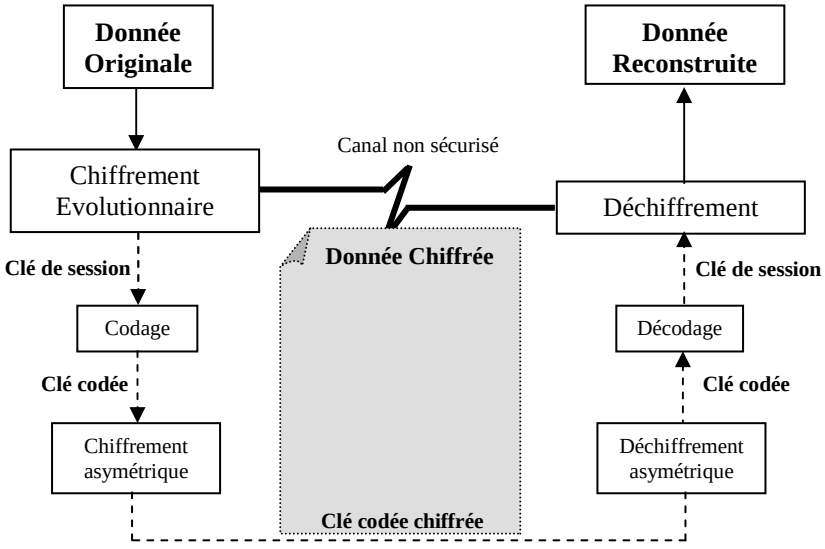


Figure III.35. *Schéma général du processus de chiffrement et de transmission sécurisée de la clé générée par chiffrement publique.*

Une deuxième solution de transmission sécurisée de la clé de session générée peut être envisagée dans le cas de manipulation de données images. Il s'agit de la technique de tatouage. Après avoir codé la clé générée par notre algorithme de la même manière proposée dans le cas de la première solution de transmission sécurisée de la clé (en utilisant un codage de Huffman par exemple), nous proposons, ici, de marquer l'image chiffrée par la clé de session codée. Ainsi, le schéma du processus de sécurisation englobant cette deuxième solution de transmission de la clé peut être résumée par le schéma de la figure III.36.

Cette deuxième solution de transmission est conditionnée par la taille de la clé codée pour satisfaire le compromis Robustesse-Capacité-Invisibilité lors du tatouage [Katz, 2000], [Barn, 2004], [Koba, 1990].

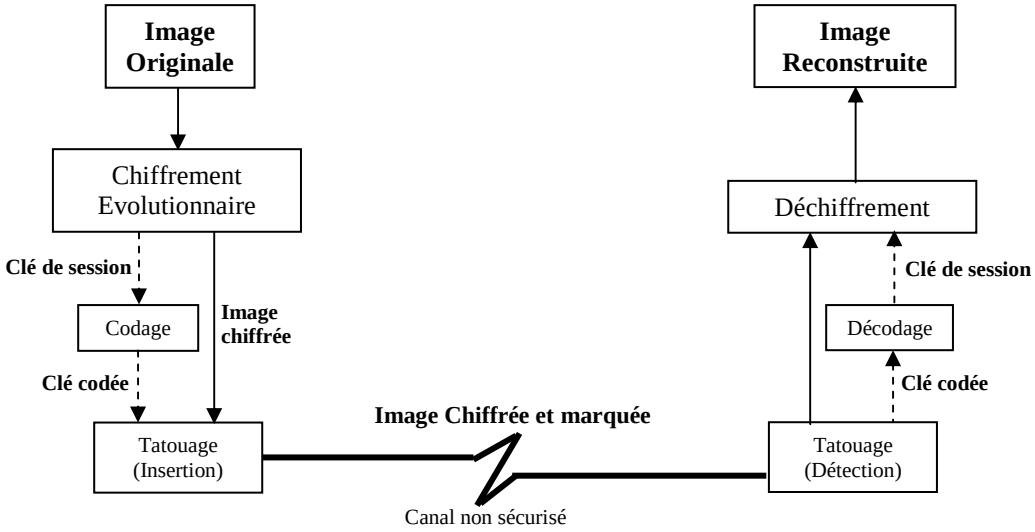


Figure III.36. *Schéma général du processus de chiffrement d'images et de transmission sécurisée de la clé générée par tatouage.*

III.6. Conclusion

À travers le présent travail, nous avons démontré que les algorithmes évolutionnaires, inspirés fondamentalement de la sélection naturelle des espèces, peuvent être également appliqués au cryptage de données. En effet, avec ces algorithmes, nous avons ramené le problème de chiffrement en un problème d'optimisation.

L'application de ces algorithmes requiert le réglage d'un certain nombre de paramètres pour la phase de chiffrement. La taille de la population initiale, le taux d'exploration et le nombre de générations affectent le temps de convergence des algorithmes, quant aux probabilités *Pc* , *Pm*, elles possèdent, théoriquement, une influence directe sur la qualité du résultat. Or, dans notre problème, les opérateurs génétiques (croisement et mutation) n'agissent pas sur l'information traitée (caractères du texte ou pixels de l'image) mais sur leurs coordonnées ; ils sont utilisés comme un outil pour explorer le reste de la population. De ce fait, les paramètres P_c , P_m affectent également la convergence des algorithmes.

Dans la pratique, les paramètres des algorithmes évolutionnaires sont réglés approximativement par tâtonnement [Gold, 1989]. Suite à une série de tests sur diverses données et en utilisant les fonctions fitness proposées, nous avons retenu certaines valeurs qui permettent d'avoir un résultat exploitable (donnée chiffrée) en un temps de calcul variant d'un algorithme à l'autre.

Les résultats obtenus montrent que les schémas proposés présentent des aptitudes dans la confusion et dans la sensibilité à la donnée originale qui les rend loin des attaques différentielles. De même, et pour pénaliser ou plus ou moins compliquer la tâche de l'attaque exhaustive, une deuxième alternative (PosESecL2) a été proposée en augmentant de trois ou de quatre fois la taille de la clé générée par la première alternative (PosESecL1). Une troisième alternative de cryptage robuste et sûre a été également définie. Il s'agit d'OEEA opérant suivant un mode de chiffrement (chiffrement à base d'occurrences) inexploitable par aucune attaque avancée.

Et pour résumer l'efficacité des deux modes de cryptage exploités, le cryptage à base de positions et le cryptage à base d'occurrences, le plus correcte est de dire que le deuxième

mode est plus important que le premier. En effet, ce mode est bâti autour de la notion d'occurrences, chose qui ne peut en aucun cas guider ou être exploitée par un cryptanalyste puisqu'elle ne présente aucune information utile pour aucune attaque possible. Ce qui n'est pas le cas pour le premier mode de cryptage à base de positions où une recherche exhaustive finira, un jour, par retrouver la bonne combinaison de positions d'éléments d'une donnée et ainsi de remettre en cause l'algorithme de cryptage utilisé.

Ainsi c'est les qualités du mode de chiffrement à base d'occurrences qui nous a motivées à le tester à travers d'autres métaheuristiques qu'elles soient à population ou à une seule solution. D'ailleurs le prochain chapitre présente une métaheuristique de colonie de fourmis traitant le problème de cryptage en opérant suivant ce mode de cryptage proposé.

Finalement, il sera utile de rappeler que l'objectif en cryptage est d'assurer la sécurité des données pendant leur durée de validité, donc, toute attaque tardive n'aura aucune importance.

CHAPITRE IV

CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

IV.1. Introduction

L'auto-organisation est un phénomène décrit dans plusieurs disciplines, notamment en biologie dans la branche qu'est l'éthologie (étude des comportements des espèces animales dans leurs environnements). Une définition a été proposée par Deneubourg [Dene, 1977] **:** *«l'auto-organisation est un processus dans lequel, un modèle de niveau global émerge uniquement, d'un grand nombre d'interactions entre les composants de bas niveau du système. De plus, les règles spécifiant les interactions entre ces composants sont suivies en utilisant uniquement des informations locales, sans références au modèle global ».*

Autrement dit, l'auto-organisation explique l'émergence d'un comportement collectif macroscopique par des interactions simples au niveau microscopique. L'auto-organisation n'exclut pas la complexité au niveau individuel. Elle suppose simplement qu'à un certain niveau, les individus se comportent comme des entités simples [Monm, 2000].

Depuis longtemps déjà, des comportements auto-organisés ont été découverts dans la nature. Par exemple, dans une colonie de fourmis, on peut observer différentes castes spécialisées dans un certain nombre de tâches : élevage de couvain, recherche de nourriture, construction de nid, etc. Aussi, le nid est construit sans que les insectes soient dirigés, ils répondent à un certain nombre de stimuli provenant de leur environnement.

D'importantes recherches ont eu pour intérêt principal l'étude de ces comportements intelligents afin de savoir comment ces populations interagissaient, accomplissaient des tâches et évoluaient. Ces recherches ont abouti à des modélisations inspirées du principe qu'un groupe d'entités plutôt simples et obéissant à des règles locales de coordination, sont capables d'engendrer des comportements globaux beaucoup plus complexes. Ces modélisations caractérisent ce que l'on dénomme « l'intelligence collective » [Bona, 1994], [Monm, 2000].

Dans ce chapitre, nous nous intéressons à l'algorithme de colonies de fourmis. Nous l'avons étudié, adapté et appliqué au problème de cryptage de données texte ou images. Le vocabulaire employé par cet algorithme est directement calqué sur celui de l'éthologie, nous parlerons donc, de fourmis, de connaissance collective, de phéromone, d'évaporation, etc.

Dans la première section de ce chapitre, nous présentons notre approche utilisée pour la conception du système de chiffrement dont l'objectif principal est l'obtention d'un système de

chiffrement symétrique non déterministe dont la clé secrète est générée pendant l'application de l'algorithme ; donc elle n'est pas connue à l'avance ce qui rend difficile le travail du cryptanalyste. Pour ce faire et étant donné que dans le précédent chapitre le problème de cryptage a été ramené en un problème d'optimisation, il a fallu, tout d'abord, coder le problème d'une manière adéquate permettant, ainsi, d'effectuer les opérations propres aux fourmis tel que : le dépôt, l'incrémentation et l'évaporation de phéromone, etc. Ainsi, le codage adopté est celui validé par la première métaheuristique d'AEs testée qui est le codage à base d'occurrences. Ce dernier permet la concrétisation des déplacements de fourmis par exploitation de la notion de permutation.

Ensuite, nous avons bâti notre algorithme en définissant soigneusement : la fonction d'évaluation cherchant à maximiser la différence entre l'information initiale et celle codée, le mécanisme de sélection adopté pour le déplacement des fourmis, le dépôt et l'incrémentation de phéromone, l'ensemble de la connaissance collective, etc.

IV.2. Motivation

Les algorithmes de colonies de fourmis forment une classe des métaheuristiques récemment proposée pour les problèmes d'optimisation difficile. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. Une colonie de fourmis communiquent indirectement via des modifications dynamiques de leur environnement (les pistes de phéromone) et construisent ainsi une solution à un problème, en s'appuyant sur leur expérience collective. C'est d'ailleurs cette dernière notion que nous cherchons à exploiter à travers l'utilisation de la métaheuristique de colonies de fourmis pour la résolution du problème de cryptage.

Ainsi, les objectifs que nous nous sommes fixés consistent à comprendre et isoler les mécanismes intéressants de cette métaheuristique, utiliser l'approche de chiffrement à base d'occurrences validée dans le précédent chapitre, suivant cette optique et appliquer l'algorithme ainsi conçu au problème de chiffrement de données texte ou images.

D'une manière générale, notre objectif est de développer un algorithme de chiffrement à base de colonies de fourmis par du constat simple que les colonies de fourmis résolvent des problèmes complexes, bien que l'intelligence d'une fourmi soit limitée ce qui est équivalent au fait de dire que, l'intelligence du système entier est plus grande que celle de la simple somme de ses parties.

IV.3. Algorithme proposé

Afin de bien comprendre notre algorithme de chiffrement proposé, AntCrypt, une description complète des différentes étapes de l'algorithme profondément inspirées de la métaheuristique *ACO (Ant Colony Optimisation)*, sera présentée. Des retouches ont été apportées sur l'algorithme de base pour l'adapter au problème étudié qui est celui de cryptage. Ainsi, le schéma de l'algorithme adopté est le suivant :

Algorithme IV.1 *AntCrypt*

Entrées
Codage de la donnée originale
Ensemble de paramètres (*m* : taille de la population, ρ taux d'évaporation)
DEBUT
1) Initialiser la connaissance collective // connaissance collective $\leftarrow \Phi$;
2) Créer la population initiale comportant *m* solutions initiales (*m* : nombre de fourmis) ;
Répéter
3) Construire les solutions;
4) Enrichir la connaissance collective ;
5) Evaluation des solutions de la connaissance collective ;
6) Sélection ;
7) Mise à jour de phéromone ;
Jusqu'à Satisfaction du critère d'arrêt
Retourner la solution trouvée ;
FIN

IV.3.1. Codage adopté

Pour prouver encore une fois l'efficacité de l'approche proposée et précédemment présentée exploitant un chiffrement bâti autour de la notion de nombres d'occurrences d'éléments de la donnée à chiffrer, le codage adopté par AntCrypt sera le même que celui d'OEEA.

La figure IV.1 rappelle le codage de données à utiliser.

$O(e_1)$	$O(e_2)$	.....	$O(e_i)$	.....	$O(e_{l-1})$	$O(e_l)$

Figure IV.1. *Codage d'une solution sous AntCrypt.*

Où : $O(e_i)$ est le nombre d'occurrences de l'élément e_i dans la donnée, l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Ainsi, notre algorithme AntCrypt cherche à changer itérativement la répartition des nombres d'occurrences $O(e_i)$ sur les différents éléments d'une certaine donnée avec génération aléatoire de positions dans la solution afin de créer le maximum de désordre dans leurs positions. Cela donne, aussi, l'avantage de produire des éléments (caractères/pixels) inexistants dans la donnée initiale.

IV.3.2. Création de la solution initiale

Initialement la connaissance collective (le savoir partagé) est vide. Dans cette étape on crée la population initiale comportant *m* solutions (*m* : le nombre de fourmis).

Nous désignons par *DonneeInitiale* le codage de la donnée originale. Donc, il est représenté par un vecteur dont les éléments contiennent les nombres d'occurrences des 1393 caractères (respectivement des 256 valeurs possibles de chacune des composantes R, V et B codant les pixels) dans le texte (respectivement dans l'image). Ces éléments sont notés :

$$O(e_1), O(e_2), ..., O(e_n) / n = \begin{cases} 1393 : Texte \\ 768 : \text{Im}age \end{cases}$$

Nous générons ensuite *m* fourmis susceptibles de créer, chacune, une nouvelle solution. Pour ce faire, chacune des fourmis est positionnée aléatoirement sur un élément de *DonneeInitiale*. Nous avons choisi de noter les différentes positions courantes par *PosCourante*. Ensuite, chaque fourmi va se déplacer un certain nombre de fois vers d'autres éléments (d'autres nœuds) notés chacun *PosSuivante* par application de la roulette aléatoire, et elle permute lors de chaque déplacement le nombre d'occurrence de l'élément « *DonneeInitiale[PosCourante]* » avec celui de l'élément « *DonneeInitiale[PosSuivante]* ». Une fois une fourmi s'arrête de se déplacer, nous obtenons une solution qui sera ajoutée à l'ensemble de connaissance collective. Alors à la fin de cette étape, la connaissance collective sera augmentée de *m* nouvelles solutions (chaque solution est obtenue par une fourmi).

IV.3.3. Construction de solutions

Après l'étape d'initialisation et dans le but de construire de nouvelles solutions sur une itération donnée de l'algorithme, les fourmis vont choisir à partir de la connaissance collective cette fois-ci leurs points de départ pour construire de nouvelles solutions. Ces points représentent les solutions ayant les plus grandes quantités de phéromone, c.-à-d. celles qui perturbent le plus la donnée initiale par rapport à une version cryptée représentée par la solution choisie. Donc le mécanisme permettant la construction d'une solution est le suivant :

1) Choisir les *m* meilleures solutions de la connaissance collective ;
Pour i = 1 : *m* ***faire***
    2) Choisir une solution *Sol* parmi les *m* solutions choisies en (1) ;
    Pour j = 1 : NbrMaxDép ***faire***
        3) Positionner aléatoirement la fourmi sur un élément *Sol[PosCourante]*;
        4) Déplacer la fourmi suivant la méthode de roulette aléatoire vers une autre position *PosSuivante* ;
        5) Permuter les deux éléments *Sol[PosCourante]* et *Sol[PosSuivante]* ;
    Fin pour
    6) Ajouter *Sol*, la solution qui vient d'être construite, à la connaissance collective.
Fin pour

IV.3.4. Evaluation

La fonction d'évaluation (ou d'adaptation) quantifie la qualité des solutions calculées. Celle que nous avons définie pour associer des valeurs d'adaptation à nos solutions calculées est la suivante :

$$F(Sol_j) = \sum_{i=1}^{n} \left| O_j(e_i) - O_0(e_i) \right| \quad \text{(IV.1)}$$

Tels que :

Sol_j : Solution j,

n : Nombre des éléments du vecteur *DonneeInitiale*,

$O_j(e_i)$: Nombre d'occurrences de l'élément (caractère / pixel) i dans la solution j,

$O_0(e_i)$: Nombre de répétitions de l'élément (caractère / pixel) i dans la donnée initiale.

IV.3.5. Sélection

Le rôle de la sélection est de distinguer les solutions sur la base de leur qualité, en particulier, pour permettre aux meilleures solutions d'une génération donnée d'être copiées dans la génération suivante.

Dans notre cas, lors du passage d'une génération g à la génération qui suit (g+1), nous favorisons les m meilleures solutions parmi celles de la connaissance collective globale y compris celles construites durant la génération g.

IV.3.6. Manipulation de phéromone

IV.3.6.1. Incrémentation de phéromone

Suivant la fonction F donnée en (IV.1), nous devons choisir la quantité de phéromone déposée pour chaque solution. Le mode de dépôt sélectionné peut fortement modifier le mode de convergence de l'algorithme. On pourrait choisir de déposer la même quantité à chaque solution. Cependant, on peut aussi décider de donner une puissance plus forte aux phéromones déposées sur les solutions améliorant l'altitude. C'est cette seconde solution qui a été appliquée dans notre algorithme.

Pour chaque solution *Sol*, nous calculons leur efficacité selon la formule (IV.1). Si cette solution existe déjà dans la connaissance collective on incrémente leur quantité de phéromone comme suit sans la réinsérer :

$$Phéromone(Sol_i) = Phéromone(Sol_i) + \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.2)}$$

Tel que EffMax est l'efficacité de l'une des meilleures solutions calculées de manière déterministe en permutant dans un ordre choisi le $i^{ème}$ élément avec le meilleur des l-i éléments restants (en maximisant la différence entre les deux éléments en question).

Sinon, si la solution n'existe pas, elle sera ajoutée à la connaissance collective en lui attribuant une quantité de phéromone selon la formule suivante :

$$Phéromone(Sol_i) = \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.3)}$$

IV.3.6.2. Évaporation de phéromone

Comme dans la nature, les phéromones sont des substances chimiques qui s'évaporent au fil du temps. Donc, il sera nécessaire que l'algorithme imitant ce coté de la nature représente ce phénomène afin d'éviter aux fourmis de converger vers un maximum local.

$$Phéromone(Sol_i) = Phéromone(Sol_i) \times (1 - \rho) \qquad \text{(IV.4)}$$

Où ρ est le taux d'évaporation.

L'évaporation de phéromone de toutes les solutions après une période de temps permettra de redonner de l'intérêt à d'autres solutions qui peuvent nous amener à la meilleure solution. Le taux d'évaporation ρ doit être bien choisi, puisque s'il est très faible, les phéromones s'accumulent et atteignent la borne max et nous risquons alors de s'enfermer dans un maximum local, et s'il est trop grand, les phéromones atteignent rapidement la borne min et beaucoup de solutions n'auront pas la chance d'être choisies par les fourmis.

Dans notre application, ce taux doit aussi dépendre des valeurs des éléments du codage. Si la différence entre deux éléments est grande, c.à.d. la fonction F va prendre de grandes valeurs correspondantes à de grandes quantités de phéromones, donc le taux d'évaporation peut prendre une valeur importante. De même, il dépend du nombre de fourmis travaillant sur une même génération : beaucoup de fourmis dans une génération augmentent la possibilité d'avoir deux solutions identiques, c.-à-d. la phéromone sera incrémentée plusieurs fois de suite, alors le taux d'évaporation sera important.

Ainsi et après un certain nombre de générations fixé expérimentalement, nous avons varié un pourcentage d'évaporation de 0% jusqu'à 0.5% en s'inspirant, au début, de l'application de l'algorithme ACO pour la résolution du problème de TSP [Barg, 2005]. Ce choix a été, ensuite, validé par expérimentation.

IV.3.7. Critère d'arrêt

Après un certain nombre de générations fixé expérimentalement, l'algorithme converge vers la meilleure solution trouvée. Reste de générer ou de reconstituer la donnée cryptée à partir du codage de la solution proposée. Ceci correspond à l'opération inverse de l'opération de codage. Il s'agit d'une opération de décodage qui procède comme suit : à partir du codage de la solution proposée qui est un vecteur de nombres d'occurrences des éléments de la donnée chiffrée, pour chaque élément *i* du vecteur *Sol*, on génère *Sol[i]* positions aléatoires du caractère/pixel correspondant à l'élément *i* dans la donnée cryptée.

IV.3.8. Déchiffrement

Nous arrivons maintenant au processus inverse qui permet de rendre la donnée à nouveau intelligible sans aucune perte d'information ; il s'agit du processus de déchiffrement.

Notre algorithme proposé est un algorithme symétrique, donc la clé générée doit être maintenue secrète. Cette clé s'obtient au fur et à mesure du calcul de l'information chiffrée au fil des générations. Sa valeur finale correspond aux permutations des positions des nombres d'occurrences des éléments de la donnée chiffrée pour obtenir celles des nombres d'occurrences des éléments de la donnée en clair. Le processus de déchiffrement consiste donc à replacer les éléments dans leurs positions initiales suite à l'introduction de la bonne clé.

IV.3.9. Réglage des paramètres et résultats

Cette section est consacrée d'une part, à l'optimisation ou la fixation du jeu paramétrique et d'autre part, à l'application de l'algorithme proposé *AntCrypt* sur les mêmes données de test utilisées pour l'évaluation des AEs proposés et présentés dans le précédent chapitre. Ils s'agissent des données faisant l'objet de la figure III.1. Dans un premier temps, nous allons exposer notre stratégie pour établir et fixer le choix paramétrique de notre algorithme. Puis nous allons présenter les résultats d'application d'*AntCrypt* sur les données modèles (textes et images), à savoir la donnée chiffrée, la clé générée, la quantité de phéromone, l'efficacité, le temps de chiffrement et de déchiffrement.

IV.3.9.1. Réglage des paramètres

Les figures IV.2 et IV.3 présentent les résultats des différents tests effectués en termes d'efficacité et de temps de chiffrement. Ces valeurs représentent la moyenne de dix (10) exécutions successives appliquées sur l'image de test Lena pour différentes valeurs de nombre de générations et de fourmis.

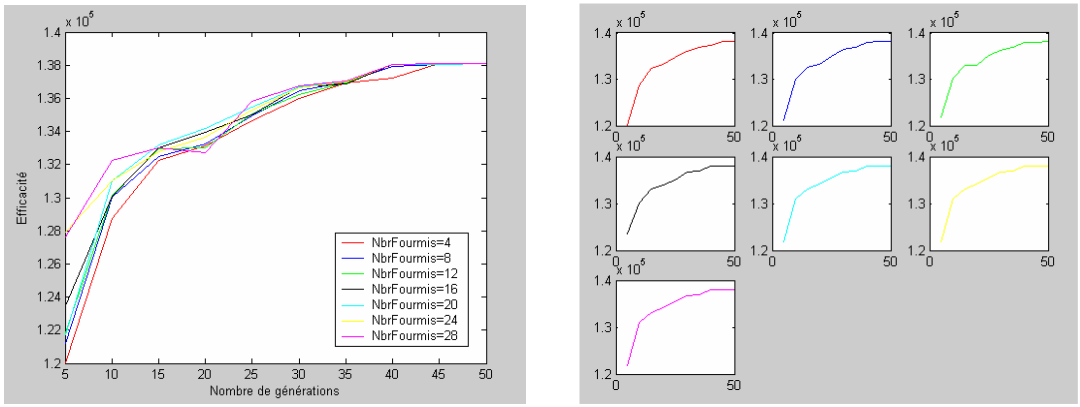


Figure IV.2. *Influence du nombre de générations et du nombre de fourmis sur l'efficacité.*

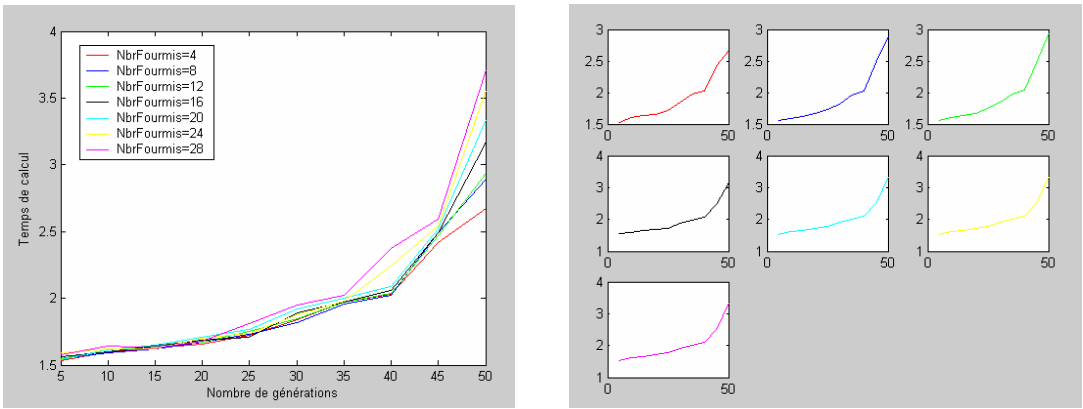


Figure IV.3. *Influence du nombre de générations et du nombre de fourmis sur le temps de calcul.*

D'après les résultats résumés à travers les figures IV.2 et IV.3, nous constatons que la majorité des résultats des tests ont des valeurs d'efficacité assez proches. La meilleure valeur d'efficacité est obtenue au bout de 50 générations à 20 fourmis (Efficacité = 138077) mais avec un temps de chiffrement assez important par rapport à l'ensemble de test (3,17 s). Cependant, le meilleur temps de chiffrement est obtenu pour le test qui a été déroulé sur 5 générations et 4 fourmis (1,53 s) mais malheureusement avec la plus faible valeur d'efficacité (Efficacité = 120009).

Les tests déroulés sur 45 générations avec 4, 20, 24 ou 28 fourmis, et 50 générations avec 4, 8, 12, 20, 24 ou 28 fourmis ont donné de très bonnes valeurs d'efficacité (138073, 138070, 138073 et 138074, 138075, 138074, 138076, 138073, 138072, 138075 respectivement) mais après de longs temps de calcul (2.42 s, 2.52 s, 2.54 s, 2.59 s et 2.67 s, 2.94 s, 3.34 s, 3.56 s et 3.71 s).

D'autres tests déroulés sur 5 et 10 générations ont montré, cette fois-ci, de bons temps de calcul (compris entre 1.53 s et 1.64 s) mais de mauvaises valeurs d'efficacité (comprises entre 120009 et 132263).

Donc, notre objectif sera de réaliser un compromis entre la valeur d'efficacité et le temps de calcul. En effet, le test déroulé sur 40 générations avec 12 fourmis semble le plus satisfaisant. Il permet d'atteindre une bonne valeur d'efficacité (Efficacité = 138063) en un temps de calcul très raisonnable (2.04 s). Ca sera le paramétrage à adopter pour *AntCrypt*.

IV.3.9.2. Résultats

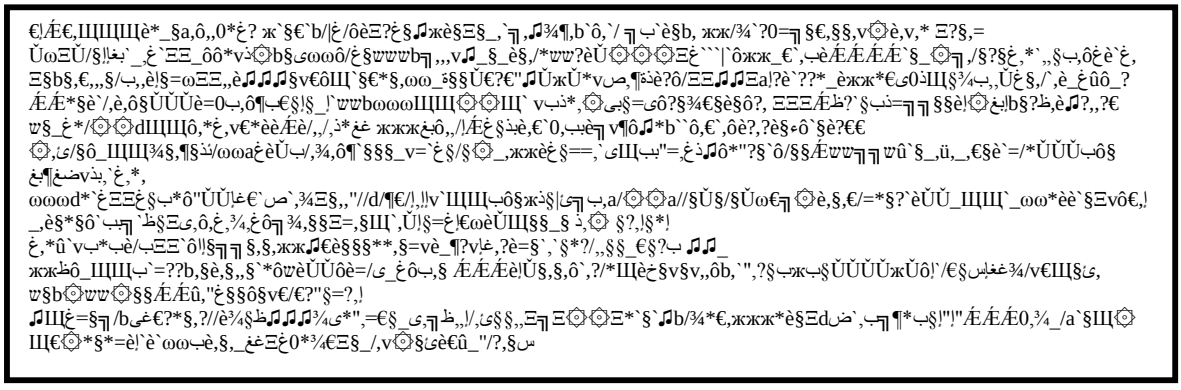
Ci-dessous, nous présentons les résultats d'application d'*AntCrypt* sur les mêmes données modèles utilisées pour tester nos trois algorithmes présentés dans le précédent chapitre : Texte1, Texte2, image Lena et image Logo suivant le paramétrage fixé dans la section précédente (résumé à travers le tableau IV.1). Toutefois, et pour démontrer l'aspect non déterministe de notre approche de chiffrement par exploitation de métaheuristiques, dont l'ACO fera l'objet ici, nous présentons, cette fois-ci, pour chaque donnée originale deux versions chiffrées en deux instances différentes accompagnées des clés de chiffrement générées dans chaque cas.

	Valeurs des paramètres d'*AntCrypt*
Nombre de fourmis	12
Nombre de générations	40

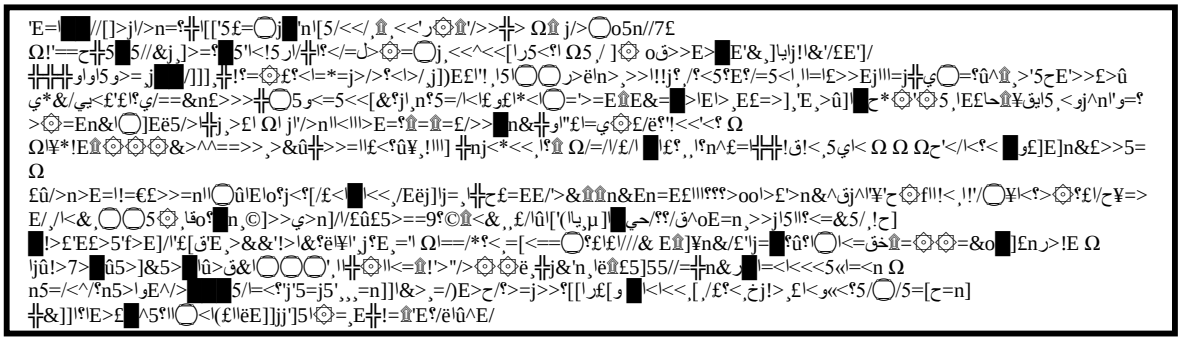
Tableau IV.1. *Valeurs adoptées pour les paramètres de AntCrypt.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure IV.4. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets) :

124	398	407	770	921	266	896	19	794	90	918	245
843	914	614	547	809	904	560	80	375	908	879	886
901	816	830	543	573	900	156	196	98	254	906	314
230	443	917	497	903	418	288	907	289	680	147	861
820	642	902	259	916	840	30	898	905	102	822	635
317	161	920	465	292	458	546	91	911	246	910	833
111	912	155	193	640	919	923	643	357	681	121	725
915	892	581	899	922	216	909	913	789	549	179	578
28	552	3	650	1070	379	850	996	934	933	1104	285
459	997	1124	1242	1026	897	1171	1075	8	521	64	281
1017	955	985	872	262	485	774	16	85	709	279	97
628	120	374	87	286	1112	277	862	518	47	255	1097
1203	45	163	60	599	12	461	5	848	1240	92	1032

331	174	888	83	1073	20	676	342	494	280	422	487
607	1020	958	61	1154	601	95	1100	112	386	1238	1196
1185	1119	393	669	1	756	1172	1012	1102	274	394	363
237	755	267	6	219	389	893	187	1225	437	175	683
834	659	1215	1085	1138	819	447	688	436	764	1158	117
206	1024	675	695	264	319	345	76	127	1183	484	1006
1191	540	1205	210	384	689	655	991	71	928	1130	416
1129	324	761	1134	1089	1029	449	1200	704	231	1013	466
474	1166	813	302	506	1050	490	1204	1092	382	194	744
1066	682	164	72	630	972	15	1213	738	309	1199	531
1149	1084	1202	971	535	395	1031	731	988	737	41	1188
1142	584	336	839	113	818	1245	306	723	657	620	403
252	858	994	423	1036	563	1217	1137	135	108	1133	1088
1160	810	829	946	1058	1068	952	944	751	673	1110	79
1232	835	733	557	1193	982	1179	298	1052	1101	198	1144
225	232	949	400	229	981	545	846	954	432	778	411
634	1161	1042	457	18	250	68	1123	866	462	1243	1051
1120	768	960	990	1189	1182	491	827	185	1145	1148	714
1115	428	574	995	1067	653	1091	978	27	415	132	940
145	409	665	1234	956	622	727	1227	786	402	1009	859
1236	1224	870	1039	387	239	1028	565	1043	594	176	856
787	424	323	588	1033	715	945	1195	359	1162	1165	793
791	930	684	299	716	1076	605	101	1241	149	826	412
1181	480	364	1187	1004	14	1226	579	438	1222	542	613
125	664	1176	197	202	1208	593	520	183	1207	248	604
434	1139	1127	1014	211	891	169	747	968	55	1190	339
1178	625	758	698	330	1038	446	425	611	668	646	1170
1174	144	166	283	1244	857	1231	1131	366	48	31	34
1107	52	814	315	24	414	983	1135	860	11	452	937
539	337	812	935	639	507	863	378	637	301	1055	1219
1077	1198	440	943	451	658	478	361	392	82	103	586
693	881	1210	652	1235	37	1001	970	1156	1175	808	184
278	662	799	341	1041	244	736	1030	109	730	984	195
1019	678	728	420	294	1212	796	1218	303	1192	627	645
1079	1109	1046	481	272	488	1116	1096	222	238	21	349
1034	2	1128	1011	130	157	445	167	705	1087	1201	33
470	489	10	598	453	128	39	1061	817	1209	1216	741
926	1184	1008	942	524	969	372	308	118	228	473	815
312	623	1122	516	1078	962	1143	929	1233	989	316	602
941	1125	1223	648	1072	1136	58	1062	224	931	571	1173
987	649	158	1040	612	998	233	864	241	140	9	621
177	88	1047	321	57	1000	1037	1117	340	1151	961	1221
595	973	138	152	959	483	213	479	207	477	191	287
740	522	977	654	284	227	332	141	234	105	842	305
1080	957	572	355	310	1140	660	258	78	322	831	1054
201	1229	467	1045	273	353	629	441	618	63	1074	849
527	924	43	173	710	295	948	708	1057	876	606	36
967	1025	1081	1147	1132	868	979	1082	966	844	1237	1010
1239	845	772	51	391	1168	855	865	823	65	62	874
275	276	159	885	182	1105	1118	1002	603	592	999	759
1035	762	1069	1083	1007	1005	932	1094	824	745	110	828
154	677	638	703	851	204	1180	806	208	760	884	771
610	1230	25	1211	439	925	1022	765	1155	670	672	519
469	356	548	679	1056	1015	236	1167	964	803	1106	218
1194	591	663	587	260	50	69	1141	953	1071	975	701
212	396	550	307	1063	203	811	743	510	84	333	1153
739	566	807	1159	1018	162	1157	385	1023	35	1114	22
644	1108	261	1093	711	963	296	328	706	122	358	597
938	965	504	1197	249	788	1098	790	325	1214	1228	401
1027	525	1086	951	624	529	1099	992	1150	136	667	380
1126	1111	199	1048	1206	580	404	697	251	719	475	40
585	1044	1095	1053	686	350	575	1163	1003	1060	1064	223
950	146	986	1177	752	304	726	123	853	742	168	492
454	766	804	1021	533	247	46	939	42	1121	221	555
226	980	327	17	180	974	798	927	209	1152	692	554
370	126	151	129	596	377	800	1169	1146	408	463	417
569	1103	615	498	326	59	456	1186	1049	137	139	291
427	632	685	1113	433	1220	153	1164	722	354	189	976
523	100	763	509	1016	444	399	369	1065	936	825	220
360	265	947	1059	749	568	1090	838	852	631	365	993

329	268	651	1330	390	500	1331	1332	460	320	468	729
1333	1274	583	617	753	564	56	782	1306	661	1334	263
165	1335	1318	795	713	371	367	1336	134	1305	271	748
1337	81	532	1299	343	486	1338	1339	472	1264	1302	413
1340	362	769	801	633	1273	517	534	867	94	335	732
1247	1290	1316	1260	89	781	718	1341	450	1323	551	1342
1343	1344	894	1292	687	515	1345	608	1346	1347	1246	351
368	767	1248	570	430	1322	735	235	1348	1349	1284	1350
1288	1278	802	847	257	499	746	805	1267	114	1295	514
1351	188	785	558	1317	1352	784	70	537	832	1353	448
181	1354	297	99	44	442	217	609	501	410	1355	837
1356	66	1314	313	38	1357	777	541	559	1358	1320	426
170	1312	1319	1279	186	32	882	171	429	776	1359	1360
880	720	671	1275	406	215	431	1253	347	792	717	1361
200	1362	1313	1363	54	754	1251	476	511	1364	435	1325
1365	1291	338	1257	1315	699	1282	641	712	381	96	1327
482	883	1280	421	1366	513	1367	26	1368	1250	779	1276
1326	242	143	1307	300	1249	1369	508	23	721	690	1298
773	895	887	1270	875	53	576	67	589	455	890	636
626	1328	405	536	647	582	119	656	696	1370	282	290
334	619	1371	1372	512	503	1373	1374	854	544	877	1375
205	493	106	1308	783	77	577	293	1376	1254	344	1252
148	526	256	1377	464	616	214	878	1294	1303	1321	1378
1297	1379	1272	1311	1300	1266	240	590	1380	530	243	1287
567	1258	1293	160	495	707	150	1381	666	702	873	505
528	674	1382	562	694	86	889	397	1277	93	1296	1285
1256	104	131	270	1383	311	49	190	1281	841	388	734
1283	1265	346	775	115	600	1263	1255	1384	352	1310	172
1259	1309	561	869	13	691	142	133	871	373	821	836
269	502	75	1385	253	1386	116	318	1301	376	1387	556
348	74	178	1268	724	107	1324	553	7	1388	1286	192
1389	29	73	780	1329	750	538	797	471	1262	1390	419
757	1391	1261	1392	1304	1289	496	4	700	1269	383	1271
1393											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets) :

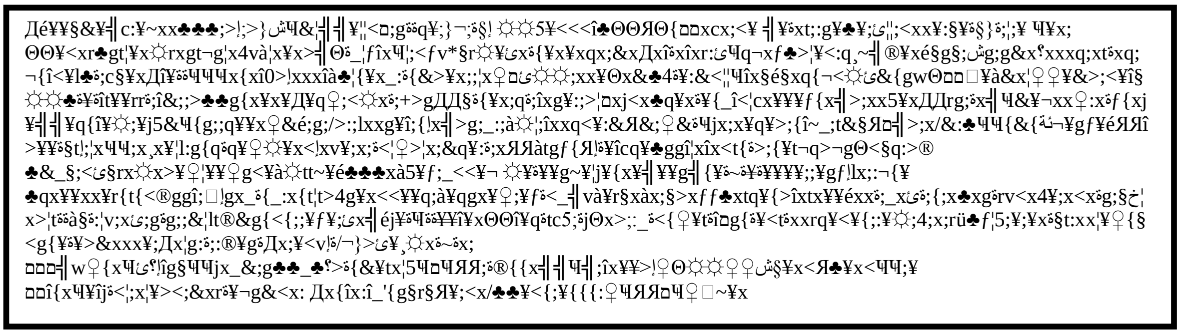
80	34	87	94	51	64	91	46	18	32	55	81
44	2	63	86	54	72	45	16	88	84	20	75
82	97	39	13	6	36	67	95	12	53	17	92
24	98	89	14	74	60	25	41	40	76	10	57
66	93	5	11	9	70	3	31	73	61	42	43
23	27	59	28	78	37	52	69	7	71	77	8
99	50	85	96	30	21	29	22	58	19	90	68
26	47	79	15	35	48	38	878	603	723	164	621
401	997	320	274	4	126	364	628	908	150	286	287
600	815	170	482	424	159	307	220	139	213	596	967
217	133	405	644	511	861	909	334	802	964	597	691
449	942	485	721	538	945	862	870	495	513	505	957
623	883	229	506	955	419	767	751	917	56	143	237
831	972	799	62	324	224	479	750	864	255	771	330
378	316	474	564	161	403	172	748	461	223	978	778
166	318	904	540	985	167	741	202	467	569	49	434
793	283	765	568	248	102	250	653	816	359	297	602
232	249	207	196	365	190	754	947	455	490	338	931
788	454	698	649	774	308	588	702	708	631	33	444
557	463	503	913	886	1	715	116	550	516	189	849
244	685	458	584	285	764	975	418	589	877	921	487
309	632	144	701	850	317	932	563	239	643	610	192
348	574	533	732	993	212	809	299	417	790	501	464
922	567	867	331	650	231	466	722	895	666	168	891
529	541	558	762	279	730	635	824	486	177	313	520
581	560	607	140	977	719	859	339	638	599	380	491
662	777	532	890	758	169	745	382	83	580	900	965
534	987	968	395	104	892	107	826	225	670	222	200
893	387	992	292	545	499	961	353	590	919	869	291
772	575	191	160	605	305	180	811	142	410	832	236
450	389	868	314	659	699	163	247	507	335	481	293
110	982	673	214	634	363	675	842	65	995	343	576
814	546	175	488	425	280	910	677	680	265	718	438
497	716	875	414	366	692	710	976	344	548	970	376

851	421	781	837	927	792	388	874	860	836	798	570
627	743	912	937	204	310	803	203	509	686	306	665
131	776	728	808	682	938	657	197	843	735	779	549
739	593	179	640	846	185	717	370	753	379	178	360
857	465	579	565	899	756	158	452	606	737	132	326
629	729	498	784	295	429	221	648	925	375	500	385
408	182	235	616	431	654	768	171	478	346	694	300
609	994	797	125	357	926	766	863	103	257	145	789
924	420	696	321	528	329	646	476	981	148	402	123
227	818	374	396	954	833	749	121	233	198	935	795
462	720	105	153	897	724	705	416	604	230	536	655
539	866	228	697	246	773	998	526	951	426	352	974
591	988	852	647	573	806	642	127	427	100	619	404
713	459	782	328	783	827	433	496	551	208	349	880
807	369	146	371	118	523	181	521	928	613	277	439
939	661	645	946	510	393	484	633	578	601	106	906
787	887	819	278	907	902	553	276	151	770	272	134
394	681	165	358	432	668	810	556	399	129	820	350
136	117	205	812	622	262	817	652	984	109	260	397
916	448	618	559	898	530	742	780	973	689	885	264
847	245	514	687	630	747	524	986	493	303	259	477
270	267	195	442	958	991	215	547	347	269	914	162
667	114	571	537	733	663	124	865	377	586	583	587
761	184	936	135	744	940	368	296	684	941	712	592
137	113	206	706	412	933	241	594	271	688	822	543
561	980	216	332	844	660	535	304	996	949	894	731
423	341	755	734	284	356	918	183	201	504	830	392
390	585	956	457	446	440	492	775	174	651	582	841
456	615	199	149	554	522	238	362	266	813	796	111
959	989	614	876	953	608	119	409	282	544	428	834
639	351	354	664	637	855	301	873	517	519	625	896
963	725	400	736	355	209	595	290	671	508	835	658
381	256	430	979	345	871	786	901	176	714	839	288
437	120	319	243	840	598	983	251	157	226	752	342
700	669	821	435	856	470	273	407	315	726	234	888
406	727	252	704	340	473	443	929	542	823	740	337
785	641	384	693	460	872	218	445	469	155	193	853
386	828	138	298	999	854	881	943	483	101	759	769
147	930	611	678	112	672	624	709	311	141	577	518
791	422	626	480	336	612	950	905	948	617	327	920
471	128	829	562	695	884	240	858	281	122	254	323
367	707	683	453	794	512	711	889	944	475	263	302
372	186	656	294	911	531	845	969	253	738	801	805
966	489	472	383	289	990	210	258	441	879	391	451
525	322	188	763	173	312	413	261	154	962	515	156
800	971	411	882	219	436	398	527	415	555	333	760
674	757	447	620	572	268	952	242	115	502	494	187
194	152	552	804	703	468	108	903	838	636	130	275
960	325	848	676	923	825	679	690	361	211	746	566
934	915	373	1097	1243	1324	1351	1271	1352	1040	1080	1241
1277	1136	1353	1089	1051	1354	1355	1039	1077	1342	1258	1320
1037	1064	1202	1242	1312	1281	1123	1054	1170	1267	1285	1222
1340	1349	1129	1111	1296	1174	1252	1345	1356	1166	1357	1115
1275	1328	1250	1128	1263	1171	1098	1284	1033	1231	1276	1005
1058	1313	1311	1059	1124	1232	1157	1013	1100	1132	1019	1099
1103	1023	1112	1225	1240	1304	1194	1215	1358	1247	1198	1334
1335	1122	1018	1210	1164	1341	1156	1127	1106	1006	1300	1323
1114	1359	1245	1113	1066	1154	1150	1036	1344	1289	1360	1238
1298	1034	1035	1093	1048	1008	1318	1253	1309	1028	1331	1361
1016	1195	1286	1362	1363	1050	1090	1038	1137	1119	1287	1079
1163	1182	1151	1125	1043	1303	1025	1325	1133	1042	1364	1002
1234	1075	1347	1088	1365	1212	1117	1230	1301	1294	1071	1214
1200	1307	1049	1264	1190	1060	1147	1299	1332	1160	1165	1366
1109	1180	1343	1305	1315	1065	1187	1055	1052	1191	1367	1110
1196	1368	1074	1205	1257	1262	1168	1336	1108	1308	1045	1369
1330	1278	1239	1134	1221	1149	1235	1104	1209	1370	1053	1265
1282	1107	1094	1233	1069	1153	1213	1244	1327	1266	1146	1229
1091	1141	1105	1346	1226	1041	1070	1186	1142	1056	1302	1063
1199	1321	1162	1193	1086	1101	1270	1083	1310	1017	1192	1177
1179	1095	1248	1184	1219	1121	1022	1273	1172	1371	1046	1372

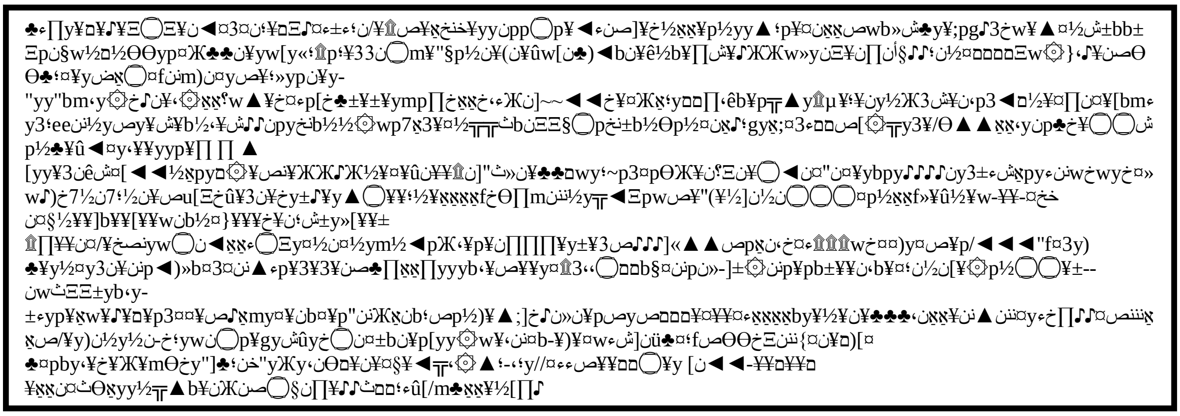
1044	1178	1096	1208	1224	1183	1003	1084	1012	1131	1130	1348
1062	1316	1159	1188	1021	1373	1009	1135	1374	1375	1203	1211
1228	1010	1087	1268	1376	1333	1081	1377	1322	1220	1145	1072
1057	1288	1061	1047	1249	1255	1120	1030	1306	1261	1189	1246
1314	1144	1378	1254	1379	1126	1295	1138	1176	1319	1152	1337
1216	1201	1236	1380	1293	1076	1185	1291	1158	1143	1381	1029
1297	1067	1259	1382	1140	1155	1068	1269	1031	1383	1384	1118
1339	1260	1290	1102	1385	1386	1082	1280	1167	1000	1078	1085
1073	1387	1197	1272	1007	1181	1227	1011	1001	1161	1206	1251
1116	1204	1388	1173	1032	1350	1024	1389	1207	1292	1217	1326
1237	1390	1223	1218	1148	1391	1392	1279	1274	1026	1027	1256
1092	1014	1015	1393	1317	1139	1338	1329	1169	1004	1175	1283
1020											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)



(b)



(c)

Figure IV.5. (a) *Donnée originale Texte2,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

700	340	911	855	661	6	451	910	17	3	151	267
937	143	936	914	475	880	745	874	939	897	155	935
281	730	209	938	173	638	777	191	329	930	384	421
673	131	485	462	512	259	736	644	457	728	932	593
933	65	920	924	926	918	349	196	232	309	553	782
917	369	919	927	455	878	255	921	913	929	156	931
912	759	400	922	616	915	594	691	243	934	592	916
819	361	128	923	486	24	925	159	748	301	928	140
411	379	1093	16	599	1169	1108	1079	1040	985	1021	141
799	1006	1067	1107	82	572	1014	325	977	498	9	943
974	670	394	805	270	49	178	658	808	604	392	1229
642	456	1058	80	1184	78	669	472	1045	1214	445	1011
195	492	242	952	389	1069	814	839	1060	591	983	895
285	248	67	947	1194	460	380	38	46	584	482	430
631	1236	478	1012	1004	297	570	1240	228	332	390	1127
1153	40	992	991	1151	625	44	1106	47	1189	99	823
986	706	395	85	559	205	28	641	887	197	319	979
1094	554	882	1066	822	1166	531	564	949	305	398	654
694	544	200	160	801	1017	189	1031	1083	1143	476	1196
1002	198	894	348	1052	619	69	214	1192	1099	120	1134
318	300	534	73	862	104	425	328	1092	1141	677	296
692	1212	466	612	1190	144	960	686	1188	53	404	668
683	381	106	672	523	1145	688	42	1053	342	102	779
292	757	1055	1183	852	514	34	890	702	525	1233	863
870	79	444	858	1241	851	904	1250	948	346	224	31
807	754	528	942	794	1140	511	1101	347	295	1198	289
436	408	587	308	269	403	247	1074	25	1148	27	264
194	489	1160	163	1056	1225	298	322	965	1202	563	237
1035	1179	941	56	1051	1057	1064	434	978	1015	532	413
602	1049	71	772	162	1123	483	1142	231	1152	844	1078
1235	1159	356	418	1047	1082	737	273	1137	676	1144	263
372	1039	530	1098	440	284	1191	1081	94	1205	29	783
1216	603	630	859	877	598	473	4	274	944	674	681
709	1154	565	1008	837	662	549	671	1003	215	710	1228
1243	758	1016	962	1238	175	1217	1158	518	57	1172	628
764	876	1223	606	7	253	657	968	1245	766	999	542
324	775	399	738	989	1097	1009	515	238	406	770	245
993	499	1167	30	217	230	1077	299	787	74	537	1029
784	802	645	76	1147	1197	415	452	1126	45	517	629
1227	1000	940	569	632	618	164	172	101	18	88	216
1135	950	397	377	879	809	1226	1176	527	227	1203	431
833	276	970	87	1027	256	480	843	409	637	286	1013
1248	142	725	605	743	1114	567	643	791	640	576	1090
1244	150	689	1018	650	589	311	516	995	1178	519	1175
611	828	621	857	1122	1020	1065	294	474	1230	959	376
869	501	898	370	545	1117	816	1071	1110	43	526	623
115	836	655	680	26	494	1195	447	1023	385	513	548
969	161	250	1185	1034	110	111	860	15	561	793	1170
961	820	133	432	1116	97	114	279	93	193	495	422
1246	1059	997	112	212	1210	1130	33	219	60	469	698
1231	174	812	479	763	1207	1218	277	50	450	416	41
954	582	116	488	127	1128	552	697	1209	1080	306	490
1085	1129	539	981	953	337	449	211	726	1061	1124	510
715	1091	861	83	1007	827	477	330	800	849	546	8
747	419	10	186	454	500	66	583	899	357	401	1001
971	659	856	226	1234	240	233	91	761	343	1046	1146
595	107	1186	1237	1103	1139	1076	1048	647	579	722	353
608	386	660	1115	596	1042	610	636	405	718	290	529
1138	234	433	1112	1164	964	945	742	967	773	20	77
1070	1221	1072	804	1224	663	1161	721	1120	1118	458	797
1193	707	1119	326	453	35	505	1030	533	126	1041	64
769	753	14	63	1024	1100	387	149	744	239	1200	703
1037	504	522	32	956	875	536	723	998	345	1171	435
1068	1155	366	832	1213	491	1028	1173	829	429	23	374
1181	976	11	973	393	1187	988	1232	135	1102	169	493
278	664	134	90	821	885	1163	762	741	627	108	892
1199	996	367	958	982	1165	417	1174	95	1150	557	1033
740	333	891	382	129	972	1247	1109	1121	266	204	1206

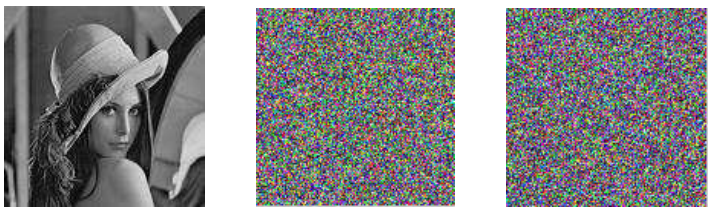
1010	388	187	966	1204	955	987	70	790	1239	503	352
957	262	1084	272	59	866	414	1168	1219	1201	1222	907
1111	317	963	1113	1026	497	487	818	810	109	442	439
190	780	423	896	130	363	1086	798	617	1050	1087	994
1105	208	1211	580	796	89	291	68	1025	251	1182	1005
682	1104	378	980	538	646	118	323	699	1157	359	1136
236	1156	1131	1075	1043	1149	853	96	229	351	179	1220
180	138	145	424	177	1019	1032	560	975	136	1132	1125
261	774	685	568	886	675	315	202	158	1036	168	1133
713	601	1215	600	176	1054	1208	817	865	678	739	864
789	1095	1162	509	1022	614	327	1249	946	607	806	760
185	1062	221	667	902	566	316	635	665	39	813	717
786	951	834	624	438	1088	58	1177	984	1044	62	719
1251	900	1096	100	1073	166	990	695	1180	463	2	334
124	1242	1038	123	148	889	615	121	1089	651	407	481
749	1063	1271	470	86	597	465	1332	1307	260	1342	1275
881	368	371	48	1343	1301	1270	1344	838	1313	1345	666
1346	132	265	871	302	840	84	502	842	1347	222	765
573	252	1348	1322	732	246	1349	835	1334	13	1319	1339
464	1350	778	771	1351	355	268	590	845	1311	210	223
1289	577	184	1282	1352	735	1269	1293	1353	1278	459	1263
241	207	1274	1354	652	1325	1298	446	12	588	1268	873
884	1355	622	1252	125	344	551	213	304	1356	1265	1357
1296	609	1330	908	693	98	5	868	555	1320	225	428
687	1331	383	541	1308	81	188	756	1267	558	154	1272
1358	1359	341	1258	52	847	467	1360	776	724	1361	22
1279	103	280	36	220	1362	795	634	578	461	571	484
181	303	321	690	1363	734	905	1285	574	19	733	412
396	313	402	1364	1365	331	139	550	51	1366	1281	365
335	746	1367	586	257	336	613	287	1253	803	883	792
653	850	72	585	310	831	1329	360	729	192	1368	218
426	21	496	727	1273	815	750	320	1326	1318	1369	1327
1370	1371	639	684	1262	358	1372	903	1373	1257	581	825
312	906	117	1374	1341	649	1375	767	1376	1264	556	1377
373	235	1378	826	1379	1336	731	872	1316	203	468	1304
1380	848	708	167	543	1291	1261	626	1256	888	712	206
1340	307	1294	854	535	338	1337	1315	1259	909	410	37
244	620	1277	443	1338	362	75	1287	471	867	153	768
575	1381	811	824	1314	171	1323	105	1254	1290	1328	271
1303	1266	846	183	441	1300	648	521	1324	1255	146	893
375	1382	1383	508	354	1283	696	54	254	282	283	656
785	788	448	841	1310	506	137	1317	1295	165	830	1333
339	1286	701	714	679	751	1299	540	704	249	147	182
1260	1384	364	1302	157	293	1306	1385	633	92	420	1309
427	288	716	1335	391	1321	55	258	1280	1288	201	61
1386	152	1387	1388	901	705	350	1389	752	1390	1	1391
720	711	755	113	562	1284	1292	1392	1297	314	437	547
1393	507	1305	520	1312	199	119	524	781	170	122	275
1276											

Clé de chiffrement de la deuxième version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

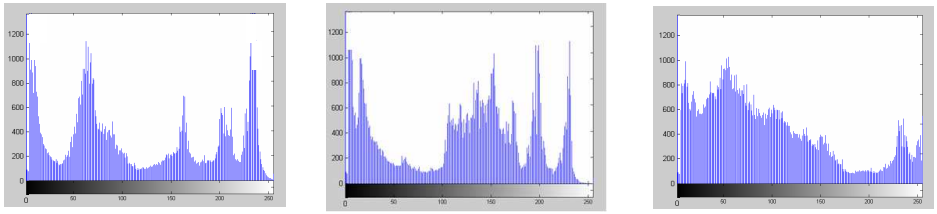
6	162	2	1	279	439	507	352	360	804	758	567
472	465	78	992	500	327	830	580	948	101	535	173
756	991	805	522	928	10	147	416	461	606	769	803
827	692	341	982	129	12	244	944	197	477	831	139
684	658	255	593	356	49	36	172	504	108	272	375
248	575	302	299	81	358	496	773	607	476	852	562
382	726	134	846	667	641	98	73	735	230	548	540
902	484	64	727	643	542	640	380	556	632	723	
835	436	748	109	721	654	619	212	277	894	211	475
705	335	257	844	856	781	938	462	961	11	623	62
651	254	74	4	746	265	525	130	943	185	52	473
457	160	251	860	59	939	447	121	946	511	757	934
258	136	974	350	35	50	483	491	965	32	941	453
339	344	514	929	137	470	27	502	724	131	184	512
362	677	213	871	802	981	39	5	570	345	848	76
328	884	412	238	68	391	376	488	325	292	882	71
142	970	810	196	220	796	730	132	296	622	550	490
590	574	3	560	87	826	158	8	518	817	973	635

819	409	689	498	206	353	526	890	41	617	440	647
396	915	935	797	96	242	975	297	285	608	177	264
782	563	99	203	659	7	544	104	761	629	719	271
333	329	877	581	332	838	778	611	704	762	674	156
972	899	610	930	366	785	927	976	956	31	263	117
312	315	664	775	752	337	413	996	878	853	963	46
637	866	111	538	310	105	30	418	445	179	814	933
151	867	597	274	86	467	609	984	273	152	834	43
408	303	594	44	505	696	419	516	631	732	794	712
284	588	682	435	874	521	858	589	969	906	33	138
398	722	698	529	372	524	93	378	122	486	554	494
448	655	103	833	547	808	9	148	924	855	728	713
386	691	751	226	468	239	311	688	482	628	28	737
753	566	373	602	543	168	118	318	326	813	14	307
309	275	485	818	942	870	653	481	868	950	993	861
690	499	766	648	443	349	679	280	493	673	824	56
424	199	321	889	621	962	166	218	977	714	487	444
336	582	281	402	624	553	387	747	441	390	546	261
743	306	492	603	58	616	267	235	89	528	703	630
811	364	497	381	715	800	250	665	893	94	379	361
913	627	532	262	530	585	980	295	388	21	697	241
903	828	971	851	995	686	672	469	888	615	519	905
595	908	501	832	169	433	897	718	63	508	228	80
816	385	922	115	862	91	423	428	95	559	427	873
786	779	676	909	15	604	979	710	354	680	370	578
601	739	150	245	157	907	342	128	953	417	454	404
881	223	240	82	17	270	646	792	125	153	940	896
266	694	571	552	135	127	898	986	901	669	772	229
842	837	916	346	371	523	47	661	464	771	783	839
790	18	308	460	577	702	154	289	770	990	57	463
114	69	24	642	400	120	671	232	650	97	48	249
774	895	605	178	919	314	863	442	926	53	558	634
921	989	458	29	587	568	741	406	434	145	657	561
791	920	369	340	246	645	821	749	429	614	403	539
330	509	144	945	225	918	541	394	112	537	706	850
405	60	879	900	845	222	374	789	474	625	549	425
209	584	276	317	421	652	426	67	843	599	198	355
947	338	557	305	37	431	319	678	290	876	155	638
596	806	451	331	124	744	809	54	411	51	182	883
912	825	170	807	16	983	857	904	666	836	955	660
988	61	175	420	159	395	742	951	515	133	26	869
815	268	410	205	219	176	191	446	729	347	252	864
534	958	849	207	466	234	65	531	90	269	75	478
384	221	181	795	183	551	22	968	459	911	479	399
414	286	734	670	192	685	925	959	545	777	300	287
174	415	357	106	88	840	960	987	42	859	636	780
471	745	294	701	592	164	600	750	733	377	764	320
949	208	422	716	681	291	140	301	449	711	83	393
165	668	343	952	84	917	700	892	126	113	639	195
954	759	966	573	85	644	102	231	224	92	885	914
180	143	613	40	141	994	887	348	875	200	20	367
188	316	456	110	720	489	503	569	450	260	626	822
119	38	383	793	767	708	186	171	167	579	736	812
572	72	784	163	77	365	555	392	243	765	452	146
283	187	322	282	107	760	683	886	407	693	662	216
841	23	707	363	564	210	359	55	999	964	401	738
217	324	847	506	598	116	201	13	586	253	576	695
823	798	865	591	699	323	256	190	389	872	788	620
288	829	533	763	247	717	313	298	19	985	854	149
278	520	931	998	455	100	687	820	34	351	768	193
923	967	740	755	480	432	25	227	725	161	799	932
189	675	937	663	204	66	957	237	565	334	891	583
618	787	233	304	259	397	997	536	656	612	437	527
495	202	368	936	510	438	633	649	801	754	910	776
880	123	194	214	236	513	978	293	215	430	517	709
731	1165	1126	1054	1145	1227	1122	1168	1335	1190	1338	1157
1224	1003	1310	1038	45	1301	1153	1295	1316	1032	1318	1329
1167	1115	70	1004	1339	1340	1341	1081	1150	1294	1342	1343
1319	1086	1125	1018	1269	1118	1012	1214	1344	1129	79	1332
1061	1213	1111	1205	1305	1130	1252	1093	1315	1057	1037	1299

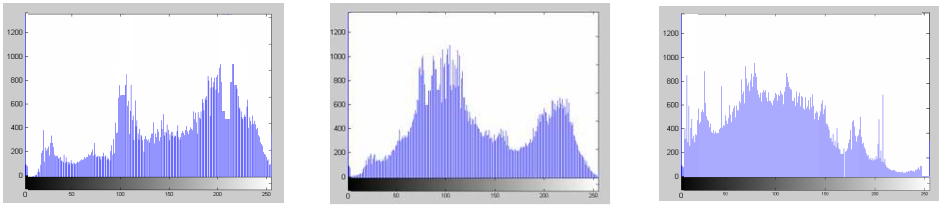
1083	1277	1062	1219	1345	1238	1346	1096	1221	1022	1103	1195
1257	1142	1050	1347	1274	1077	1088	1006	1058	1220	1046	1020
1171	1268	1116	1348	1278	1013	1188	1291	1237	1314	1296	1349
1350	1351	1109	1192	1352	1353	1189	1267	1229	1005	1155	1107
1243	1060	1008	1091	1128	1064	1023	1041	1293	1245	1248	1102
1113	1354	1263	1131	1355	1163	1356	1015	1039	1357	1247	1000
1053	1311	1358	1283	1049	1124	1065	1359	1360	1361	1281	1173
1254	1181	1114	1276	1191	1336	1104	1223	1184	1362	1079	1123
1226	1259	1137	1273	1112	1070	1363	1287	1035	1197	1364	1365
1140	1366	1367	1368	1127	1288	1369	1199	1244	1010	1133	1105
1024	1370	1371	1048	1180	1312	1317	1045	1139	1330	1177	1019
1040	1029	1055	1230	1225	1285	1275	1334	1372	1085	1036	1076
1206	1258	1208	1333	1028	1002	1154	1320	1186	1166	1161	1030
1135	1261	1095	1101	1067	1280	1136	1194	1097	1302	1260	1099
1034	1373	1172	1063	1374	1132	1094	1080	1337	1234	1265	1175
1066	1169	1047	1087	1246	1121	1375	1193	1068	1292	1376	1377
1138	1378	1325	1075	1326	1232	1203	1249	1092	1073	1379	1297
1152	1001	1071	1306	1026	1256	1242	1304	1380	1148	1156	1321
1201	1303	1021	1239	1210	1222	1381	1290	1251	1204	1025	1264
1382	1383	1084	1187	1228	1271	1146	1185	1078	1324	1384	1385
1215	1216	1176	1170	1100	1098	1212	1016	1031	1218	1209	1159
1196	1217	1178	1233	1241	1106	1162	1266	1141	1143	1090	1240
1250	1298	1328	1108	1307	1386	1262	1089	1387	1074	1027	1313
1388	1389	1056	1179	1009	1151	1119	1044	1255	1331	1134	1286
1272	1117	1033	1147	1231	1052	1149	1207	1309	1200	1164	1282
1390	1007	1160	1198	1082	1072	1308	1284	1253	1174	1069	1391
1202	1392	1182	1017	1270	1235	1236	1042	1043	1211	1322	1051
1183	1011	1120	1323	1158	1289	1279	1327	1110	1393	1014	1300
1059	1144										



(a) ***(b)*** ***(c)***



(d)



(e)

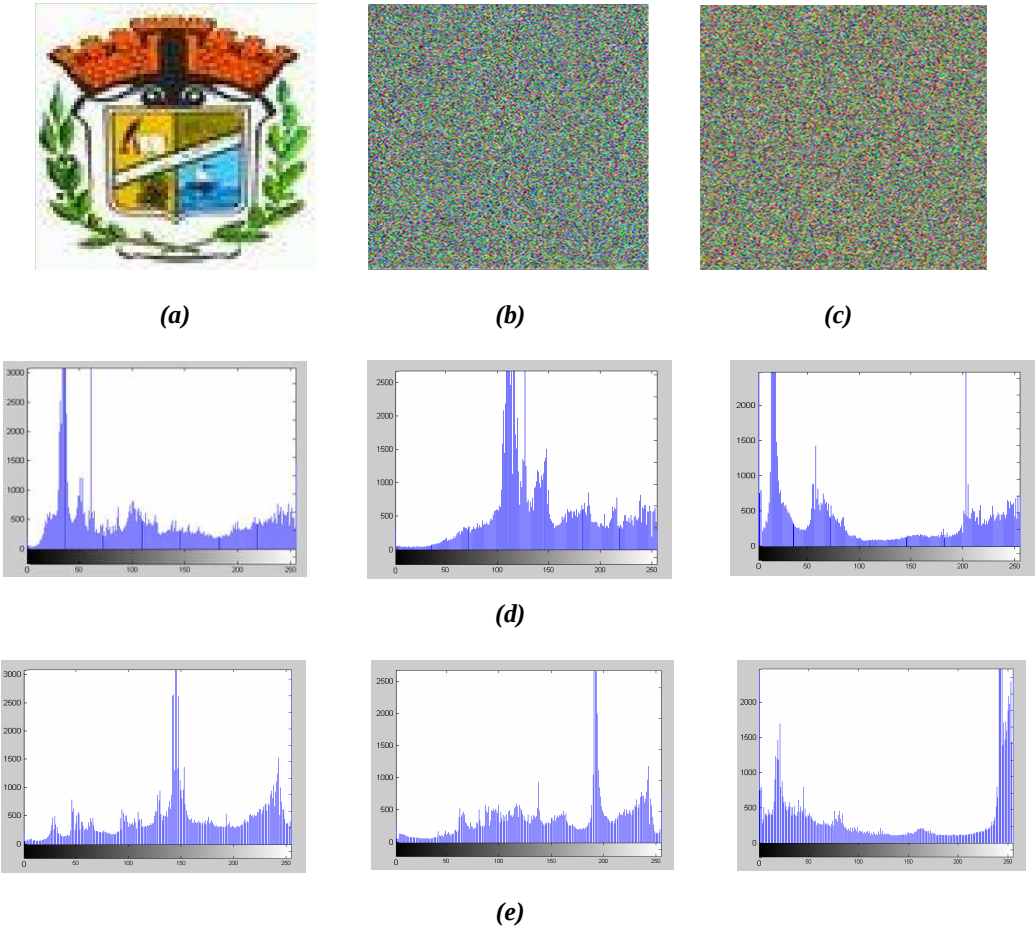
Figure IV.6. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

5	623	186	3	733	275	321	55	398	496	437	199
363	266	63	526	685	316	438	200	324	525	595	106
194	114	109	386	198	126	674	374	751	755	176	273
541	467	369	279	516	661	197	609	447	4	86	679
716	196	117	422	736	430	129	474	261	762	527	630
759	737	615	362	694	367	687	583	581	646	19	676
234	645	203	574	284	580	332	389	517	613	743	637
732	272	209	431	453	408	495	557	651	394	30	242
337	735	551	566	473	477	391	104	50	79	401	11
535	514	500	269	352	143	372	216	664	267	341	499
184	713	543	553	458	300	22	51	718	325	712	68
532	704	512	501	1	7	547	690	763	256	522	536
130	381	215	511	224	371	647	94	485	70	78	494
761	220	436	622	21	577	149	725	582	443	322	334
191	576	424	697	513	10	296	96	550	345	492	88
154	585	506	239	549	456	519	590	702	399	426	505
212	339	120	727	631	692	166	159	201	67	684	742
102	440	537	225	29	655	709	562	567	308	171	644
539	452	264	502	592	416	510	274	111	753	621	561
235	208	226	45	468	370	175	471	410	747	349	42
304	387	677	230	439	157	534	121	170	657	8	552
617	213	691	455	150	43	223	678	484	95	301	36
442	503	598	579	449	610	731	156	635	563	90	249
103	247	432	524	206	633	6	533	584	160	564	721
754	750	91	538	162	722	508	285	250	353	254	744
726	717	597	283	185	559	625	393	570	669	69	504
174	421	457	329	355	87	268	616	136	195	639	336
100	720	668	518	54	116	214	605	319	303	190	58
23	405	342	548	428	376	231	470	373	34	320	670
49	475	132	181	588	271	418	314	167	703	243	404
364	123	419	297	227	48	291	624	461	81	528	738
192	105	766	619	62	25	187	330	115	636	441	450
152	545	760	498	365	221	71	383	153	530	134	660
662	189	757	396	348	368	602	596	238	714	205	671
749	614	435	493	317	395	672	307	182	481	17	47
415	338	27	57	682	640	648	765	60	219	356	689
155	173	673	629	683	529	600	658	448	32	292	384
125	172	327	681	113	740	351	603	24	312	675	148
40	643	15	309	402	294	93	217	232	122	434	135
326	183	281	315	593	480	118	53	288	358	89	142
128	39	699	131	241	41	521	66	343	282	741	734
218	277	420	202	540	13	642	486	589	204	723	478
730	715	340	710	764	80	653	472	211	445	483	411
459	464	433	18	392	138	618	406	377	571	193	591
360	366	158	680	252	469	382	145	52	35	462	72
407	278	133	73	101	412	26	165	745	344	575	12
599	400	739	695	748	310	568	139	293	140	608	403
375	708	587	768	298	666	76	554	251	606	414	572
245	97	222	74	270	507	388	44	601	696	546	594
229	107	706	168	311	350	487	112	667	255	33	444
361	177	454	169	164	479	509	82	686	463	515	228
663	244	357	649	711	259	628	265	397	707	728	423
665	2	632	626	460	654	210	290	124	767	163	333
280	429	295	489	451	698	565	127	37	641	491	688
161	531	302	719	179	75	729	724	604	482	693	347
240	413	61	490	746	378	586	258	84	627	427	248
263	335	16	542	523	233	556	31	144	257	38	64
331	92	544	701	650	466	56	573	188	659	306	578
253	287	756	236	425	299	119	611	65	9	555	323
141	417	110	318	359	20	146	77	656	758	652	207
289	151	237	98	385	620	700	85	137	634	147	260
354	180	276	560	59	380	465	313	569	346	409	638
558	108	328	612	476	488	46	446	262	286	246	178
607	83	705	752	390	497	99	305	28	379	14	520

Clé de chiffrement de la deuxième version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

81	506	289	165	107	184	553	751	570	17	729	724
457	204	65	760	510	256	116	708	314	296	315	381
538	6	734	302	480	33	139	345	54	144	113	57
393	424	399	163	470	71	599	733	51	653	151	30
175	743	321	556	187	232	112	323	363	707	588	183
235	260	128	37	477	453	462	267	88	304	351	56
308	359	762	464	121	374	294	382	715	392	478	609
647	205	431	253	89	594	666	262	622	592	224	322
311	164	277	29	341	291	154	646	699	522	495	74
625	47	320	395	644	77	43	100	494	723	402	706
702	105	152	244	283	737	404	340	750	629	396	764
125	412	49	387	22	364	484	643	337	242	131	389
342	690	120	572	567	435	745	176	466	13	209	403
517	508	279	265	4	252	117	132	475	587	391	310
635	331	492	502	360	372	380	456	710	104	96	669
565	416	257	531	3	326	612	171	568	1	563	444
434	507	41	259	505	577	84	192	147	447	597	747
347	711	335	744	383	421	540	367	394	660	143	281
657	178	339	418	562	436	301	766	27	94	516	66
703	161	91	313	58	63	136	610	203	45	162	292
195	483	317	705	228	62	196	476	18	539	64	368
141	451	503	39	169	307	229	297	672	350	670	626
596	309	9	19	740	353	519	160	673	134	614	400
541	75	649	605	748	223	82	137	388	525	667	655
535	640	90	585	659	448	243	295	170	48	299	237
754	533	726	452	548	528	231	230	590	700	24	586
182	678	732	101	573	290	156	272	207	631	607	701
177	445	155	365	545	73	38	583	284	80	560	138
591	469	559	97	423	398	119	208	603	527	512	514
767	78	12	173	490	95	324	126	619	349	69	488
665	662	460	693	459	180	420	604	509	720	598	595
529	174	5	50	87	271	683	561	696	422	188	59
245	620	439	482	240	580	361	385	685	593	755	142
358	274	415	497	571	698	677	370	536	679	31	663
641	219	468	736	375	222	189	757	432	158	216	639
471	413	633	455	23	85	642	407	168	651	675	238
206	450	123	146	465	518	276	739	546	319	199	145
611	106	689	636	688	714	461	768	348	632	214	133
442	333	46	129	118	344	654	10	68	722	627	379
581	575	758	730	390	661	411	682	172	249	334	14
652	695	11	336	576	487	278	725	268	384	61	630
191	520	211	521	239	589	756	26	286	409	417	127
227	501	55	298	426	713	735	35	148	159	437	401
186	330	99	638	618	550	410	226	15	44	40	430
742	103	140	458	111	369	449	373	537	202	615	83
270	674	472	600	564	608	397	645	215	234	721	130
261	153	269	515	579	316	405	741	621	727	628	656
254	354	352	122	72	763	76	287	34	752	181	446
544	21	236	566	441	524	617	300	513	79	248	467
115	731	474	408	704	491	305	547	557	377	20	427
719	697	102	357	52	346	526	709	582	185	718	149
542	481	7	716	318	761	606	312	288	201	429	109
549	493	338	285	378	454	86	438	613	419	601	329
303	293	712	110	728	247	433	157	443	282	212	218
197	634	694	746	489	135	558	623	717	680	246	648
264	664	366	574	676	124	362	36	53	8	213	200
530	668	114	555	543	485	650	498	233	42	150	325
166	98	511	343	749	738	414	552	753	251	250	255
356	280	386	266	551	28	273	328	93	499	275	92
486	108	198	534	32	532	194	70	569	217	691	658
406	578	16	263	327	2	221	496	765	463	332	681
616	500	25	428	554	479	306	67	637	584	371	190
692	473	258	60	759	220	425	440	671	523	167	225
210	355	684	602	179	687	376	193	686	624	504	241



(a) ***(b)*** ***(c)***

(d)

(e)

Figure IV.7. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée*
(d) *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

92	39	9	38	13	16	45	76	98	47	24	8
83	40	3	78	80	90	61	26	68	89	53	52
19	31	50	18	12	14	54	64	2	46	58	65
35	71	5	28	74	36	44	73	79	99	96	62
63	94	29	15	37	11	75	10	30	43	7	51
91	55	33	66	57	23	81	41	70	56	77	88
4	95	25	1	69	86	22	82	97	67	42	59
20	87	32	49	72	48	27	425	652	573	441	543
468	129	748	420	738	513	353	453	569	700	185	525
707	143	512	264	237	198	755	658	487	221	741	557
484	647	227	257	708	162	179	579	197	724	60	326
629	757	548	220	413	115	266	258	120	443	218	639
253	669	673	260	712	202	345	430	225	561	284	449
481	109	294	661	248	552	710	354	433	644	372	630
161	684	648	21	476	432	110	489	331	503	403	111
507	363	637	680	17	734	556	588	438	692	666	367
753	723	613	287	34	604	505	168	273	267	421	564
239	542	742	382	642	471	508	278	614	333	359	234
136	764	285	765	545	672	259	114	214	572	170	520
147	458	502	352	244	659	423	688	537	175	386	706
609	678	328	149	272	462	393	567	379	6	144	330
523	145	562	417	119	761	767	226	478	355	726	518
737	722	583	127	440	238	397	429	339	247	173	450
296	635	254	378	341	213	488	134	409	292	731	427

399	494	193	280	314	93	704	123	469	148	634	657
303	344	501	705	404	733	171	627	662	551	311	470
743	510	401	157	554	346	491	340	698	256	240	347
125	611	727	589	358	568	374	325	631	307	361	745
308	756	746	164	369	131	306	448	640	245	565	118
586	668	167	580	313	553	758	189	84	124	350	262
645	703	676	574	540	255	348	575	515	286	128	664
398	366	531	113	217	456	300	681	224	223	593	699
691	422	585	506	402	434	641	270	528	446	426	696
301	656	690	302	529	151	682	204	735	571	416	663
165	108	514	435	444	689	190	290	158	602	140	335
299	582	174	677	516	616	203	736	104	454	536	625
279	577	396	495	312	750	310	349	559	205	315	141
728	269	192	250	208	695	718	709	740	618	687	587
550	283	322	142	209	282	199	187	276	336	653	229
321	176	730	626	714	342	327	219	599	139	608	112
752	566	603	581	390	210	466	373	650	576	152	447
233	163	384	196	242	362	768	180	188	570	461	601
126	216	395	365	497	309	751	201	230	739	595	177
178	249	133	605	766	116	207	394	667	318	281	194
530	600	117	376	235	636	623	646	85	732	598	555
437	241	693	334	222	100	293	612	406	184	492	563
560	547	534	679	407	215	535	351	674	332	475	670
271	212	275	320	558	749	721	418	181	186	633	371
701	591	385	517	533	467	638	686	490	590	364	375
154	606	683	388	762	759	200	474	729	649	405	360
500	166	655	389	343	538	628	493	524	596	483	594
527	172	243	549	754	304	473	415	477	103	597	615
464	295	725	368	412	195	277	231	713	632	297	480
211	442	459	620	532	660	651	319	410	544	485	419
445	744	101	252	191	121	298	486	265	439	182	463
150	452	392	457	400	146	106	747	526	357	370	665
317	482	763	377	720	451	578	621	455	356	324	521
130	288	228	156	323	122	504	643	246	522	496	619
436	697	694	675	431	316	584	232	607	107	381	671
719	261	428	411	408	498	105	159	717	610	539	715
153	268	546	617	387	102	160	414	685	716	465	460
338	511	391	622	289	291	479	138	702	305	711	424
654	155	499	169	329	137	183	251	383	132	592	472
206	263	541	135	760	509	274	624	337	380	519	236

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

528	192	38	202	98	315	156	142	97	339	304	457
569	700	485	538	200	292	180	544	443	684	130	287
352	689	571	468	536	101	169	244	639	353	215	24
735	501	545	266	628	575	359	745	653	1	694	535
411	37	9	342	491	497	146	278	620	88	533	80
151	380	298	137	521	508	409	61	123	191	728	662
558	115	398	514	454	50	323	541	429	410	686	488
228	232	247	559	378	234	181	261	233	345	243	643
758	387	384	33	8	761	412	346	371	695	750	40
308	477	128	179	487	493	294	604	11	504	129	710
68	39	196	739	725	753	43	63	187	73	26	749
401	578	162	176	404	395	699	302	140	270	157	272
563	341	249	81	381	92	652	58	673	289	67	712
547	768	94	322	172	218	241	69	721	438	618	645
242	723	611	556	635	688	297	193	641	474	553	62
654	25	502	310	301	148	269	428	377	70	106	592
672	617	258	439	708	658	34	737	274	407	113	112
279	12	209	119	326	531	207	677	584	42	282	330
199	237	268	648	251	731	447	338	744	685	126	557
216	610	621	483	328	467	205	674	184	259	44	210
649	532	75	600	114	625	473	290	334	83	596	752
256	434	470	82	650	425	141	482	763	534	116	201
354	385	96	549	601	726	576	717	283	173	730	212
707	525	522	373	452	281	348	117	227	318	589	565
208	663	214	505	711	609	276	93	526	31	767	23
636	687	667	372	570	303	581	369	252	510	343	546
540	219	177	418	182	722	760	495	171	656	255	78

329	691	608	293	580	690	229	27	262	513	554	223
296	22	152	432	57	333	159	211	347	516	665	675
727	676	124	450	713	299	564	86	679	4	757	226
311	335	393	402	161	217	702	102	741	54	260	397
637	213	118	548	230	32	364	622	382	111	254	379
568	107	174	165	366	195	417	153	759	512	697	764
367	696	742	76	597	394	435	158	480	361	103	149
120	448	706	755	716	178	166	520	602	537	422	740
682	577	327	357	475	168	560	585	598	436	35	413
110	189	414	250	127	552	358	332	403	190	668	41
587	424	122	588	567	391	612	17	376	416	631	291
355	175	307	222	456	154	340	66	427	586	331	238
542	84	370	484	295	566	183	471	437	236	529	583
593	52	459	319	614	271	449	613	131	698	55	6
313	19	280	60	550	325	349	59	46	714	396	671
453	45	605	245	337	105	85	350	469	167	664	400
669	351	623	660	197	720	591	594	224	16	246	51
421	562	692	2	724	478	388	442	666	53	3	320
419	659	132	91	383	423	235	121	71	462	461	481
543	20	147	160	515	415	316	87	527	203	300	754
286	138	681	288	15	284	579	511	143	704	524	65
509	632	198	574	64	163	29	455	231	139	624	733
638	539	765	275	309	680	472	263	108	321	463	607
640	693	523	135	630	619	701	389	683	627	732	170
709	99	603	420	426	312	441	374	440	392	747	77
719	136	661	317	756	368	406	48	476	489	305	748
109	715	465	615	285	496	375	155	21	446	519	408
561	18	498	220	734	336	5	530	188	762	492	399
464	13	518	555	499	766	433	431	204	458	273	751
248	74	89	479	186	100	705	506	551	500	405	572
28	267	14	743	444	164	150	240	430	362	655	257
616	265	657	185	494	703	644	642	56	134	646	390
445	104	678	647	729	718	363	466	306	599	629	517
360	10	507	746	225	736	595	47	253	651	194	95
582	633	490	486	451	344	606	738	7	634	277	386
503	144	324	30	670	36	365	125	145	356	79	264
314	590	90	626	460	573	206	133	49	239	72	221

Le tableau suivant résume les résultats relatifs au temps de calcul (temps de chiffrement), la taille de clé, la quantité de phéromone et l'efficacité dans le cas de chiffrement des différentes données.

			Taille donnée (éléments)	**Taille clé (bits)**	**Efficacité**	**Temps calcul (s)**	**Phéromone**
Données Texte	Texte1	Version-Chiff1	1084	15323	11784.0	5.63	9,06
		Version-Chiff2			12030.0	6.02	9,13
	Texte2	Version-Chiff1	1151	15323	14498.0	6.7	8,67
		Version-Chiff2			14712.0	6.58	8,65
Données Images	Lena	Version-Chiff1	131 X 131	7680	135362.0	2.16	2.75
		Version-Chiff2			130074.0	2.31	1.17
	Logo	Version-Chiff1	420 X 395	7680	244926.0	2.87	23.11
		Version-Chiff2			226188.0	2.84	21.81

Tableau IV.2. *Résultats obtenus par AntCrypt.*

IV.4. Discussion et évaluation des résultats

La première chose à remarquer, d'après les résultats résumés à travers le tableau IV.2, c'est que le temps de chiffrement des données texte est beaucoup plus grand que celui des données images malgré que la taille des données images est généralement plus grande en comparaison à celle des données texte. Cela revient au fait que la taille du vecteur codant les données texte (1393 éléments) est beaucoup plus grande que celle du vecteur codant les données images (768 éléments), ce qui nécessitera beaucoup plus de temps de calcul lors de l'application du processus de résolution à base de fourmis. Toutefois, le temps de chiffrement est raisonnable pour les deux types de données texte ou images.

Nous remarquons, aussi, que les temps de calcul des différentes versions d'une même donnée sont proches, car ils sont indépendants de la taille de la donnée à chiffrer (une différence apparaît quand la dissemblance entre les tailles de deux données de même type sera importante et qui représente un temps de lecture ou de codage). En effet, la taille du vecteur codant toutes les données de même type et soumises au chiffrement est la même (1393 éléments pour le cas de texte / 768 éléments pour le cas d'images), donc sa manipulation sur l'ensemble des générations prend le même temps. La toute petite différence correspond au temps de codage des données avant le lancement du processus de chiffrement. Toutefois, la légère différence entre le temps de chiffrement des deux versions d'une même donnée revient au caractère aléatoire exploité dans certaines phases de l'algorithme.

Pour le cas de chiffrement des deux données images Lena et Logo, nous constatons que le temps de chiffrement de l'image Logo est plus grand que celui de l'image Lena. La différence est due au fait que le temps de chiffrement est la somme des temps de lecture puis codage de l'image qui varie suivant la taille de l'image, plus le temps de manipulation du vecteur codant les solutions qui n'a aucune relation avec la taille de l'image. Donc, la différence en temps de chiffrement des deux images est justifiée.

La valeur de la phéromone dépend de la valeur d'efficacité maximale et de l'efficacité de la solution calculée. Il est bien clair que pour de grandes valeurs d'efficacité, une amélioration de la quantité de phéromone sera marquée. Prenons l'exemple du premier message où pour la première version chiffrée nous avons obtenu 2166 / 9,06 pour efficacité / phéromone. Cependant, pour la deuxième version chiffrée, nous avons eu 2184 / 9,13. Donc, l'amélioration de l'efficacité est certainement accompagnée d'une amélioration de la quantité de phéromone. Il sera utile de rappeler, ici, que l'évaporation est la cause de diminution de la quantité de phéromone.

En mesurant le taux de différence entre les deux versions chiffrées de l'image Lena et l'image originale (l'image Lena), le tableau suivant récapitule les résultats obtenus en termes de NPCR, MAE et MSE. Il est clair que les deux versions chiffrées sont presque totalement différentes de l'image originale. Autrement dit, les pixels formant les versions chiffrées sont presque tous changés, soit de positions, soit de nombres d'occurrences et de positions ; ce qui donnera lieu à un très bon niveau de confusion.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.9207	1.08	0.73
Lena-version chiffrée 2	0.9082	1.03	0.69

TableauIV.3. *Niveaux de confusion de AntCrypt.*

Maintenant comparant AntCrypt avec le plus rapide de nos trois algorithmes développés et présentés dans le précédent chapitre, bien sûre sur le plan de temps de calcul. Le tableau suivant rappelle les temps de convergence des deux algorithmes OEEA et AntCrypt. Ce dernier a montré un temps de calcul meilleur que celui de OEEA que ça soit pour chiffrer des données textes ou images malgré qu'aucune explication exacte ne peut être donnée du fait que les deux algorithmes utilisent un même mode de chiffrement (chiffrement à base d'occurrences), donc un même codage qui est exploité par les composants de la métaheuristique impliquée (opérateurs génétiques et mécanismes de la sélection naturelle pour OEEA et fourmis et mécanismes de manipulation de phéromone pour AntCrypt). Et vu que le temps de convergence de notre algorithme de chiffrement basé sur les mécanismes évolutionnaires (OEEA) est légèrement plus grand que celui basé sur les fourmis (AntCrypt), donc possible que les premiers mécanismes consomment plus de temps de calcul que les deuxièmes, mais aucune preuve n'est connue !

	Temps calcul (s)	
	Texte 1	Lena
AntCrypt	5.825	2.235
OEEA	6.54	3.37

Tableau IV.4. *Temps de calcul de AntCrypt en comparaison avec le temps de calcul de OEEA.*

Remarque :

Dans le tableau IV.4, les temps de calcul de l'algorithme AntCrypt pour les deux cas de données (Texte 1 et Lena), représentent la moyenne des temps de calcul des deux versions chiffrées (Voir tableau IV.2).

De même et vu que la méthode de chiffrement présentée dans ce chapitre (Cryptage à base de fourmis) et celle présentée dans le précédent chapitre (Cryptage évolutionnaire) appartiennent à une même catégorie mère qui est celle de métaheuristiques, donc toute les deux vont bénéficier de quelques caractéristiques cryptanalytiques en commun. Il s'agit de l'aspect non déterministe démontré à travers les résultats présentés ci-dessus où deux versions chiffrées différemment correspondent toutes les deux à une même donnée originale (voir figures IV.4, IV.5, IV.6 et IV.7), et d'une robustesse contre les types d'attaques les plus avancées (attaque exhaustive, attaque statistique, attaque fréquentielle et attaque différentielle) comme c'était démontré dans la section III.5 du chapitre précédent.

IV.5. Conclusion

ACO est une méthode stochastique qui requiert de plus en plus l'attention de la communauté scientifique. Cette métaheuristique a prouvé sa performance pour plusieurs problèmes d'optimisation combinatoire. En effet, l'utilisation des traces de phéromone permet d'exploiter l'expérience de recherche acquise par les fourmis et ainsi renforcer l'apprentissage pour la construction de solutions dans les itérations futures de l'algorithme. En même temps, l'information heuristique peut guider les fourmis vers les zones prometteuses de l'espace de recherche.

Ainsi et dans ce chapitre, nous avons présenté, interprété et discuté les résultats expérimentaux obtenus par l'application de notre algorithme de chiffrement proposé *AntCrypt* sur des données modèles texte et images. Nous sommes arrivés à fixer nos paramètres par expériences, en choisissant 40 générations et 12 fourmis en essayant de réaliser un compromis entre l'efficacité et le temps de chiffrement.

L'algorithme, ainsi développé et qui a été nommé *AntCrypt*, a été comparé avec nos algorithmes développés à base d'algorithmes évolutionnaires où il a montré de bonnes caractéristiques telles qu'un bon niveau confusionnel, un aspect non déterministe, une taille sécurisée de clés, lui offrant une bonne résistibilité contre les attaques les plus avancées. En plus le temps de convergence de cet algorithme est meilleur que celui du plus rapide de nos trois algorithmes proposés dans le précédent chapitre.

CHAPITRE V

CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

V.1. Introduction

Nous nous sommes intéressés, dans les deux chapitres précédents, à la résolution du problème de cryptage de données texte et images par application de métaheuristiques à populations. Les approches que nous avons proposé sont caractérisées par leur qualité de confusion, leur temps d'exécution réduits pour certains d'eux (OEEA et AntCrypt) et leur robustesse face aux attaques avancées.

Dans ce chapitre, nous nous intéressons à l'exploitation d'une métaheuristique appartenant à une autre catégorie, les métaheuristiques à trajectoire. Il s'agit de la recherche tabou. Cette dernière s'appuie sur une recherche locale combinée à un mécanisme de prévention des cycles, grâce à un système de mémoire des mouvements précédemment appliqués ou des configurations visitées (la liste tabou).

De manière générale, une recherche locale démarre d'une solution initiale possible et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant. Un voisinage d'une certaine solution correspond à des éléments adjacents à cette solution dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration dans la solution courante ne puisse être faite.

Nous présentons, après la description de l'approche proposée, les résultats numériques obtenus afin de les comparer aux résultats obtenus par nos autres méthodes proposées (cryptage évolutionnaire et cryptage par colonies de fourmis) et certaines autres méthodes de la littérature.

V.2. Motivation

La recherche Tabou est une métaheuristique basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

L'originalité de la méthode de recherche tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objective f et l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum

quelconque par acceptation de la dégradation de la fonction objective, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste d'une certaine longueur, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

V.3. Algorithme proposé

À partir d'une solution initiale, le principe général de la recherche tabou est le suivant (voir Algorithme V.1). À chaque itération de la recherche, une partie ou l'ensemble des voisins de la solution est exploré, et un des mouvements parmi ceux de coût minimal est sélectionné. Ce mouvement est appliqué quel que soit son coût, à la condition de ne pas créer un cycle dans le processus de recherche locale à court terme (en menant à une configuration dont les caractéristiques sont stockées dans la liste tabou). Une exception est faite (critère d'aspiration) lorsque le mouvement permet d'atteindre une solution de meilleure qualité que la meilleure solution enregistrée à ce stade. La liste tabou est mise à jour à chaque itération de la recherche en fonction des mouvements choisis. Ce mécanisme permet de sortir des minima locaux en acceptant des mouvements détériorants et en empêchant parallèlement le retour immédiat à la solution de minimum local qui vient d'être quittée.

Algorithme V.1 *Schéma général d'un algorithme tabou*

```
Engendrer une configuration initiale s
s* ← s
T ← ∅ liste tabou
tant que condition d'arrêt non satisfaite faire
    m ← meilleur mouvement (i.e. minimisant f) parmi ceux non tabou
    ou ceux vérifiant un critère d'aspiration
    Modifier s en effectuant le mouvement m
    Mettre T à jour
    si f(s) < f(s*) alors
        s* ← s
    fin
fin
retourner s*
```

La succession des étapes suivantes montre plus de détails du schéma du mécanisme utilisé pour la construction d'une solution au problème étudié :

1) Détermination d'une solution initiale $S_{(0)}$ codant l'image originale.
2) Création puis la mise à jour d'une liste tabou qui mémorisera les déplacements effectués.
3) Choix aléatoire d'un nombre d'emplacements ou d'éléments de la solution courante ($i^{ème}$ solution) à déplacer.
4) Génération aléatoire d'un ensemble de voisins pour chacun des éléments choisis dans l'étape précédente.
5) Choix du meilleur voisin de l'ensemble généré.
6) Calcul tabou de la solution suivante $S_{(i+1)}$.
7) Création et la mise à jour d'une table de hachage qui comportera les dernières meilleures solutions obtenues.
8) Évaluation de la solution calculée $S_{(i+1)}$ obtenue à chaque itération.
9) Validation de la solution calculée $S_{(i+1)}$.
10) Vérification du critère d'arrêt de l'algorithme.

En ce qui suit, nous décrivons les principales étapes de l'algorithme de chiffrement tabou proposé et que nous l'avons appelé *TabuCrypt*.

V.3.1. Création de la solution initiale

La solution initiale exploitée par *TabuCrypt* s'obtient en codant les données originales suivant le même mode de codage adopté par *OEEA* ou *AntCrypt*. C'est d'ailleurs le même codage utilisé pour modéliser toutes les autres solutions.

V.3.2. Choix des éléments à déplacer

Les solutions sont représentées soit, par un vecteur englobant 1393 éléments pour le cas de manipulation de données texte soit, par un autre regroupant 768 éléments pour le cas de manipulation de données images. *TabuCrypt* cherche à réarranger aléatoirement le vecteur correspondant à la solution initiale codant la donnée originale en permutant les éléments entre eux. Toutefois, la taille du vecteur est assez grande (1393 éléments ou même 768 éléments), c'est pourquoi la manipulation un par un de ses éléments sur l'ensemble des itérations de l'algorithme nécessitera un grand temps de calcul. Ainsi, sauf un sous ensemble du vecteur globale sera manipuler pour déplacement. Cela permet, d'un coté, d'optimiser le temps de calcul et, d'un autre coté, de converger petit à petit vers la solution finale en évitant, ainsi, le problème de convergence prématurée.

V.3.3. Génération de voisinage

Le voisinage est le responsable sur le fait de pousser l'algorithme à l'amélioration de la qualité des résultats. Dans notre cas, il est généré de la manière suivante :

Pour chacun des éléments du sous-ensemble construit dans l'étape précédente, nous générons aléatoirement un ensemble de voisins candidats au déplacement avec l'élément en question. Lors de l'élection d'un voisin, les conditions suivantes doivent être vérifiées :

- ✓ Un voisin ne doit pas être élu plus qu'une fois, c'est-à-dire, tous les voisins de l'ensemble de voisins sont différents les uns des autres.
- ✓ Les voisins élus n'appartiennent pas dans la liste tabou.

Une fois l'ensemble des voisins construit, nous choisissons celui qui diffère le plus de l'élément à déplacer. Formellement, soit E_V l'ensemble de n voisins V, et E_i l'élément candidat au déplacement. La sélection du voisin v avec lequel l'élément E_i sera permuté se fait comme suit :

$$v = V_j \,/\, j = \overline{1,n} : Max\left(\left|E_i - V_j\right|\right) \qquad \text{(V.1)}$$

V.3.4. Mise à jour de la liste tabou

Dans notre cas et à une itération donnée, la liste tabou regroupe les voisins qui ont participé à des déplacements au cours des itérations précédentes avec l'un des éléments du sous-ensemble construit dans l'étape 2 décrite dans la section V.3.2. Ainsi, la mise à jour de cette liste (liste tabou) se fait à chaque fois qu'un élément sera permuté avec un voisin choisi suivant l'étape 3 décrite dans la section V.3.3 en ajoutant ce dernier à la liste des éléments tabous. A la fin de chaque itération, le contenu de la liste tabou sera effacé.

La politique utilisée pour rendre tabous certains éléments évite la modification des nombres d'occurrences originaux, puisque le processus de calcul de la solution ne doit pas modifier les éléments du vecteur initial mais plutôt il cherche à changer leur dispersion.

V.3.5. Calcul de solutions

Au cours d'une itération donnée, le calcul de la solution proposée à cette itération se base sur la solution calculée à l'itération précédente. En effet, des éléments du vecteur présentant la solution à l'itération précédente échangeront leurs positions avec celles des voisins désignés pour chacun d'eux. Cette nouvelle dispersion des éléments forme le nouveau vecteur solution. Il est à noter que les éléments non élus pour un déplacement garderont leurs positions sans modification.

V.3.6. Mise à jour de la table de hachage et évaluation des solutions

La table de hachage sert à mémoriser les différents points visités de l'espace de recherche en gardant les différentes solutions calculées sur l'ensemble des itérations. Le but sera d'interdire la recherche à revenir vers des points déjà visités à travers les précédentes itérations. Ainsi, une solution générée à la fin d'une certaine itération sera comparée au contenu de la table de hachage avant qu'elle soit soumise à une validation finale en vu de l'accepter comme intéressante solution à partir de laquelle la recherche peut continuer sur l'ensemble des itérations restantes. Si cette solution représente un nouveau point de l'espace non encore visité, elle sera ajoutée à la table de hachage pour quelle fera l'objet des futurs tests. Cette politique sert à échapper aux minima locaux et à favoriser l'exploration de nouveaux points de l'espace de solutions possibles.

La validation définitive d'une solution S_i calculée à une itération i donnée, consiste tout d'abord, à l'évaluer. Dans notre cas, l'évaluation se fait suivant la fonction d'évaluation illustrée par la formule suivante, notant que S_0 soit la solution initiale (voir étape 1).

$$F(S_i) = \sum_{j=1}^{l} \left| S_{i_j} - S_{0_j} \right| \qquad \text{(V.2)}$$

Où l : représente le nombre de valeurs possibles que peut prendre un élément d'une donnée (égale à 1393 pour le cas de données texte et 768 pour le cas de données images).

Une fois l'évaluation faite, la qualité de la solution en question sera comparée à celle de la solution calculée à l'itération précédente. Si elle est meilleure que la précédente, la recherche à travers la prochaine itération va se continuer à partir du point représenté par cette solution, sinon elle sera rejetée et la recherche continuera à partir de la solution calculée à l'itération précédente.

Ces points peuvent être résumés comme suit : à chaque itération la solution calculée est ajoutée à la table de hachage puis elle sera examinée selon les trois critères suivants :

1) Si la solution S_i obtenue n'est pas une nouvelle solution alors elle sera rejetée et les étapes de 2 à 6 sont à refaire.

2) Si la solution S_i obtenue est une nouvelle solution mais plus mauvaise que celle de l'itération *i-1* (S_{i-1}), de même elle est rejetée, mais sauvegardée dans la table de hachage, et les étapes de 2 à 6 sont à refaire.

3) Si la solution S_i obtenue est une nouvelle solution mais, cette fois ci, est meilleure que celle de l'itération (i-1), alors elle sera retenue pour continuer la recherche à partir de la prochaine itération et, ainsi, elle servira à calculer la solution suivante S_{i+1}. En même temps, elle sera mémorisée dans la table de hachage.

Remarque :
Dans notre algorithme, on considère comme *mouvement global* le passage d'une solution courante S_i vers une solution suivante S_{i+1}, en respectant les conditions citées précédemment. Toutefois, les *mouvements élémentaires* sont les déplacements des éléments et leurs voisins vers leurs positions mutuelles. Ainsi, un mouvement global émerge comme résultat des mouvements élémentaires.

V.3.7. Critère d'arrêt

Pour toute recherche itérative un critère d'arrêt est obligatoire pour éviter le problème de boucles infinies. Ce dernier qui doit être fixé à l'avance permet de déterminer le nombre de fois que la tâche à exécuter sera répétée.

Pour notre algorithme, nous avons choisi de fixer, expérimentalement, le nombre d'itérations. Ce choix influe directement sur longueur du temps de calcul et de la qualité de la solution construite.

V.3.8. Mécanismes avancés en recherche tabou

V.3.8.1. Intensification / Exploitation

L'intensification consiste à approfondir la recherche dans certaines régions de l'espace, identifiées comme susceptibles de contenir un optimum global.
Pour notre cas, les mouvements globaux en plus des mouvements élémentaires serviront à mieux intensifier la recherche. En effet, les mouvements élémentaires traduisent les déplacements des éléments d'un vecteur solution vers les positions de leurs voisins et vise versa. Ces voisins sont les éléments optimaux (les meilleurs voisins) de l'ensemble des voisins candidats au déplacement (voir étape 3). Donc, nous cherchons à exploiter les qualités des éléments. De même, les mouvements globaux cherchent à exploiter les qualités des solutions (les vecteurs d'éléments) en n'acceptant de poursuivre la recherche qu'à partir des solutions améliorantes lors du passage d'une itération à l'itération qui suit.

V.3.8.2. Diversification / Exploration

La diversification vise à utiliser des mouvements encore jamais réalisés afin d'explorer de nouvelles régions de l'espace de recherche et des régions éloignées du voisinage actuel. Dans notre cas, cette caractéristique est accomplie par la sélection aléatoire d'un nouveau voisinage non tabou pour chaque élément.

V.3.8.3. Critère d'aspiration

Le critère d'aspiration autorise un mouvement tabou sous certaines conditions. Il consiste à tester si la solution produite de statut tabou présente un coût inférieur ou de qualité meilleure que ceux de la meilleure solution trouvée jusqu'à présent. Si c'est le cas, le statut tabou de la solution est levé.

Dans notre cas, la notion de tabou est exploitée lors de la construction d'une certaine solution en interdisant les mouvements élémentaires vers les éléments voisins avec lesquels les éléments du sous-ensemble construit comme l'indique l'étape 2 ont échangé de positions. Ainsi, le mécanisme d'aspiration n'a pas de place dans notre méthode du fait que toute libération d'un élément de la liste tabou entraîne une modification des éléments des vecteurs et, par conséquent la modification des caractéristiques servira à calculer ultérieurement la donnée déchiffrée, tandis que notre but est, plutôt, modifier la répartition des éléments des vecteurs.

V.3.9. Déchiffrement

L'opération inverse au chiffrement est le déchiffrement qui permet de regénérer la donnée originale précédemment chiffrée par notre algorithme de chiffrement tabou *TabuCrypt*. Pour ce faire, ce processus exploite une information secrète calculée lors du chiffrement à coté de l'image chiffrée. Il s'agit d'une clé de session secrète. Elle représente les permutations des positions des l éléments (1393 éléments ou 768 éléments) du vecteur codant une certaine donnée à travers les différentes itérations. Une fois elle parviendra sans modification au destinataire approprié, elle servira à calculer la donnée originale.

V.3.10. Réglage des paramètres et résultats

Le schéma général de l'algorithme finalement implémentant les étapes de l'approche de cryptage tabou proposé est le suivant :

Algorithme V.2 *TabuCrypt*

Étape 1 : Générer une solution initiale de gain total G_0.
Étape 2 : Initialiser
- le nombre d'itération iter (à 1),
- la table de Hachage H (ajouter la solution initiale à H),
- la liste tabou T (vide).

Répéter les étapes de 3 à 11 jusqu'à iter = iterMax.
Étape 3 : Initialiser le compteur d'éléments (composants d'une solution)
Répéter les étapes de 4 à 6 jusqu'à
Étape 4 : Sélectionner aléatoirement un ensemble de voisins des éléments non tabous, puis choisir le meilleur de l'ensemble.
Étape 5 : Déclarer comme tabou le voisin choisi (voisin ajouté à T).
Étape 6 : Déplacer l'élément courant et le voisin choisi vers leurs positions mutuelles.
Étape 7 : Evaluer la nouvelle solution calculée pour mesurer son gain G_{iter}.
Étape 8 : Si : la solution calculée est déjà contenue dans H *alors :* Solution ignorée, *Sinon :* Ajouter la solution à H
Étape 9 : Si $G_{iter} > G_{iter-1}$ *alors* continuer le calcul à partir de $Solution_i$ *Sinon* Continuer le calcul à partir de la $solution_{iter-1}$
Étape 10 : Vider T.
Étape 11 : iter + +.

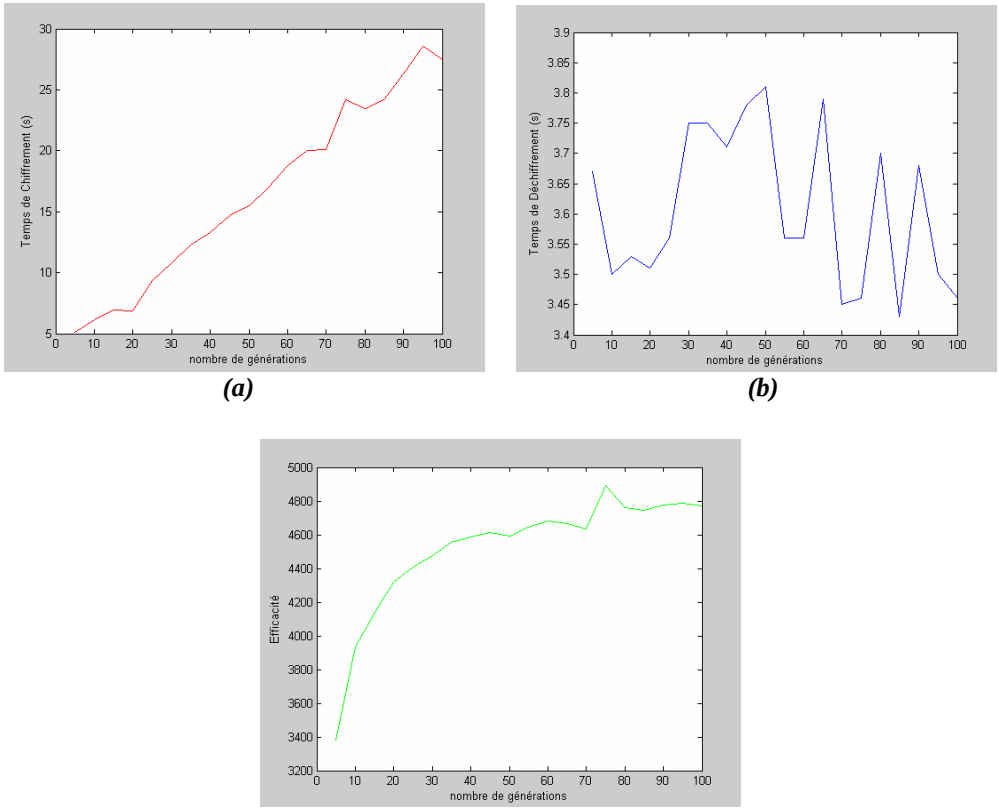
Nous précisons, dans cette section, les valeurs des paramètres de *TabuCrypt* ainsi que les résultats de son application sur les données modèles précédemment présentées (Texte1, Texte2, Image Lena et image Logo).

V.3.10.1. Réglage des paramètres

Les principaux paramètres de *TabuCrypt* concernent le nombre de générations ou d'itérations de l'algorithme et le nombre de voisins seront présentés.

Après *IterMax* itérations, fixée expérimentalement, la procédure de recherche tabou s'arrête et fournit la meilleure solution trouvée. La figure V.1 récapitule la moyenne des différentes valeurs de test et leurs impacts sur le temps de convergence et la qualité de la solution trouvée représentant une version chiffrée de l'image de test Lena.

D'après les résultats résumés dans la figure, ci-dessous, nous constatons que la meilleure efficacité a été obtenue pour un nombre de générations égale à 75 après un temps de chiffrement de 24.20 s et un temps de déchiffrement de 3.46 s. De même, les nombres de générations 80, 85, 90, 95 et 100 ont donné de bonnes valeurs d'efficacité (4760, 4744, 4777, 4789 et 4771, respectivement), mais leurs temps de chiffrement sont assez longs (23.46 s, 24.18 s, 26.29 s, 28.54 s et 27.48 s, respectivement).



(a) ***(b)*** ***(c)***

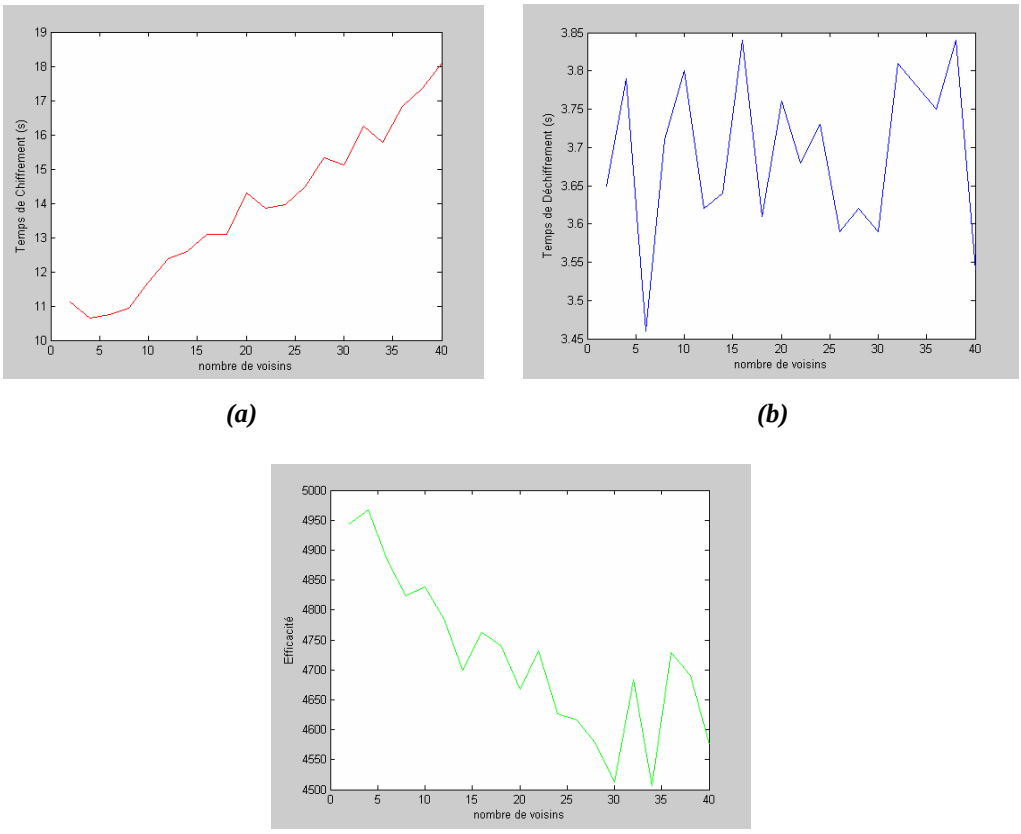
Figure V.1. *Résultats obtenus suivant les différentes valeurs de test de nombre d'itérations (générations) :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Toutefois, les résultats obtenus, pour le test avec 45 générations, sont acceptables que se soit sur le plan efficacité (4612) ou temps de chiffrement (14.70 s). D'autres valeurs de nombre de générations (55, 60, 65 et 70) ont donné des résultats d'efficacité de même ordre que celle du nombre de générations 45, mais après des temps de chiffrement relativement plus longs (29.93, 28.76, 20.04 et 20.12, respectivement). Ainsi, la valeur de *IterMax* a été fixée à 45 générations.

L'autre paramètre à régler est le nombre de voisin. Pour ce faire, nous avons utilisé le même mécanisme que celui utilisé pour régler le paramètre relatif au nombre d'itérations. Les résultats des différents tests sont résumés à travers la figure V.2 en donnant l'influence des différents nombres de voisins testés sur le temps de convergence et l'efficacité de la solution trouvée.

D'après le contenu des figures V.1 et V.2, la meilleure efficacité (4967) a été obtenue avec un nombre de voisins qui vaut 4 après un temps de chiffrement assez raisonnable et représentant le plus petit temps de calcul sur l'ensemble des tests effectués (10.64s).

Ainsi, nous avons opté à régler le paramètre relatif au nombre de voisins en lui attribuant la valeur 4.



(a) ***(b)***

(c)

Figure V.2. *Résultats obtenus suivant les différentes valeurs de test de nombre de voisins :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Le tableau suivant (tableau V.1) résumé les valeurs de paramètres adoptés par *TabuCrypt.*

	Valeurs des paramètres de*TabuCrypt*
Nombre de voisins	4
Nombre de générations	45

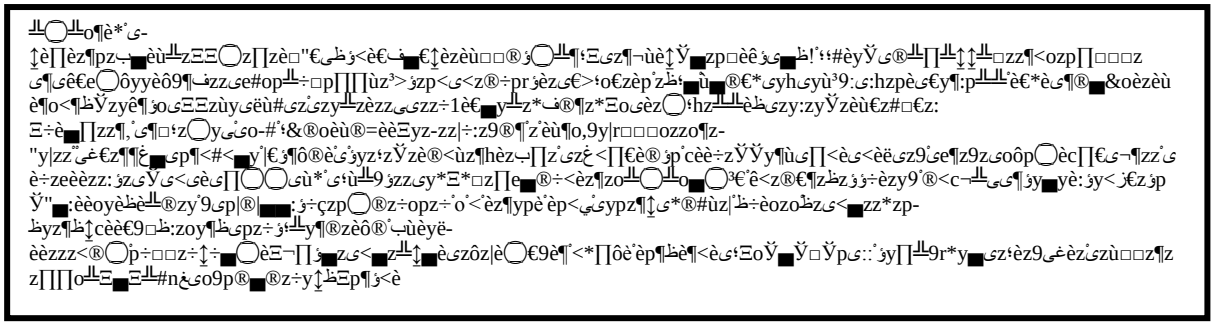
Tableau V.1. *Valeurs adoptées pour les paramètres de TabuCrypt.*

V.3.10.2. Résultats

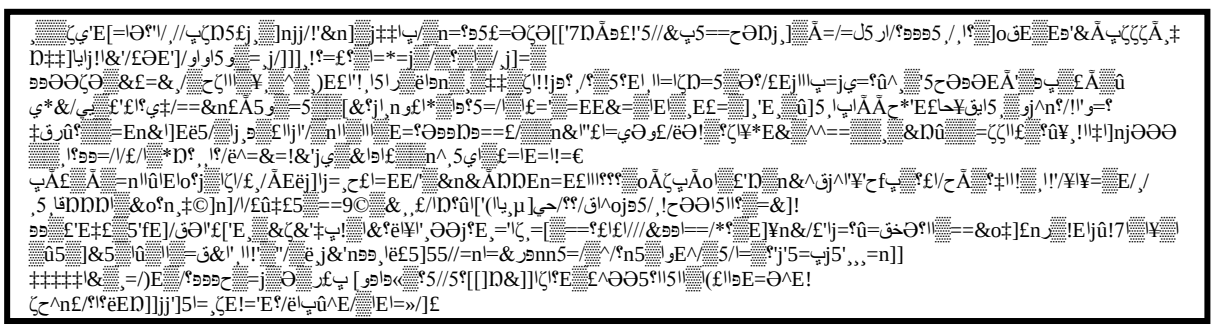
A ce niveau nous présentons les résultats de chiffrement par *TabuCrypt* des données tests. Deux versions chiffrées de chacune des données tests sont présentées accompagnées des clés générées dans chaque cas. Le tableau V.2 résume les temps de chiffrement et de déchiffrement ainsi que l'efficacité de chacun des exemples.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure V.3. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets)

728	898	353	668	901	875	906	437	425	40	739	524
797	890	877	786	357	654	889	258	566	884	883	899
399	721	896	133	905	483	410	895	735	117	900	876
878	303	479	907	706	879	100	813	91	453	152	634
874	186	597	902	480	341	99	42	736	438	456	893
794	344	881	267	887	474	617	191	885	56	892	886
897	377	601	904	34	98	672	880	903	673	496	760
373	882	894	888	347	463	750	891	520	248	465	571
624	116	573	744	866	646	926	68	1228	131	311	532
299	518	139	1134	1120	734	714	290	22	97	541	1047
1234	961	1110	819	52	153	255	1006	560	960	1077	942
1109	154	836	30	412	1119	1202	1037	1003	383	1238	335
392	1035	1241	119	448	409	1115	266	1036	857	971	194
713	912	422	378	500	782	274	1048	1102	651	1129	1091
810	696	1040	994	321	362	46	563	1148	149	339	858
503	328	110	770	1053	773	272	598	77	1087	1187	1121
551	418	225	801	1185	1082	955	557	737	506	464	1070
1240	168	187	843	796	82	867	450	200	764	190	982
1076	967	873	260	695	224	1018	207	1049	277	3	652
792	1014	1213	936	466	621	829	2	400	1085	38	23
1159	570	509	816	973	164	205	1201	112	1078	870	333
364	1212	350	250	102	536	172	629	705	178	381	722
283	1100	823	426	84	122	134	1128	591	361	798	24

1224	1111	179	1143	526	1219	1166	176	834	10	1221	1093
1062	1232	167	587	647	342	490	1041	1114	785	525	1020
473	1233	856	562	423	579	527	427	1194	215	653	242
1237	767	795	1153	862	952	998	644	1081	461	182	232
29	539	1025	1223	434	411	320	932	1204	499	11	546
1112	558	958	1160	486	467	1039	436	1245	1151	421	920
126	692	174	802	581	1066	852	355	1023	180	125	821
1104	129	265	1094	158	1101	824	871	555	1156	106	681
1145	645	803	1208	114	1068	1080	1222	269	996	1074	263
1064	1075	1184	213	699	123	312	1056	1130	175	468	754
380	1063	616	234	1200	768	865	1154	108	1069	631	698
360	1207	397	356	711	540	484	1180	1215	941	1220	596
604	1244	774	916	1050	749	420	169	1029	1242	424	376
1107	1178	820	352	372	118	443	818	701	1136	690	950
1196	93	724	543	1155	476	1034	963	196	978	1086	472
1230	781	610	273	1125	1137	531	120	554	264	969	763
165	407	390	1209	583	49	648	183	331	104	985	130
1126	1152	977	43	917	508	614	992	428	1067	582	983
987	454	497	230	530	285	374	1097	758	308	354	495
664	128	825	441	990	288	951	1131	1008	1004	632	121
752	249	105	682	1195	460	1162	981	746	297	228	700
1116	1095	296	83	1164	1198	528	502	27	762	769	1139
655	863	514	1028	276	811	1177	663	432	12	1002	716
826	96	59	458	1225	1181	136	416	830	1072	937	209
94	986	784	940	146	658	204	976	945	918	185	1012
643	611	1169	1142	1079	1173	793	457	1191	393	212	635
239	984	261	231	1010	1186	1203	576	1124	1140	404	738
991	33	1163	358	586	1098	324	968	1083	642	972	76
1013	1182	1092	751	16	88	817	957	569	135	1117	535
622	670	1009	291	832	329	28	979	72	236	1144	462
298	79	626	772	221	238	487	559	1236	556	946	633
702	69	235	1005	109	140	319	452	1054	155	1019	304
485	756	512	74	398	216	1210	925	1171	911	1243	842
997	478	1239	293	921	429	160	933	1088	431	145	1229
661	295	322	35	1055	1043	1016	201	1150	637	620	482
369	790	253	640	719	491	1061	1042	237	113	914	1122
64	348	171	727	233	254	522	206	1174	81	115	511
1193	928	1218	346	931	405	1	613	1214	394	709	592
359	307	553	708	850	163	999	166	39	1011	370	949
710	1113	1157	935	1022	860	147	78	948	198	1211	177
330	19	771	211	684	993	677	980	1206	927	723	804
1065	970	1044	1099	729	523	442	908	1217	1165	159	9
828	995	150	244	1231	15	325	137	1189	1205	844	691
922	964	564	89	281	243	671	218	657	1000	1172	939
1059	1032	683	649	783	989	757	606	385	965	470	954
389	195	959	806	1060	615	676	630	219	1170	1052	294
257	864	869	371	715	301	840	318	1118	1158	268	1027
627	435	1103	47	1015	1175	975	1192	313	1084	1132	529
67	787	953	1167	1135	924	919	1073	1031	1058	853	271
384	471	812	947	18	1161	765	567	934	217	66	759
545	475	8	1147	505	1001	1045	599	1141	1235	1176	584
284	1216	929	704	1089	943	501	717	445	966	693	605
259	1168	733	743	747	1138	550	336	343	53	814	1133
144	913	439	1051	872	1127	909	1024	459	915	17	504
944	1096	1046	73	726	1105	1026	956	1188	718	549	1146
48	910	494	189	222	1017	827	334	612	585	745	678
208	580	974	938	1123	396	1190	1030	367	55	337	366
962	246	21	4	777	1179	1149	1227	220	930	1226	1021
800	659	1183	731	202	1197	403	1090	988	533	379	1057
1106	1071	515	282	451	1038	574	1033	1199	226	1108	923
1007	280	111	544	406	61	748	1274	402	753	181	1374
143	306	868	1264	6	193	489	618	1351	667	1346	1375
1267	32	732	687	286	688	507	363	1308	859	95	575
855	1376	345	1358	65	1361	1257	565	44	1371	214	703
447	36	561	1329	799	203	625	720	278	1287	845	210
444	1280	1327	1303	414	1268	1273	846	477	446	1253	578
251	1377	822	229	1328	1295	847	419	1320	837	1378	831
1373	1362	538	1379	662	1254	1334	778	1331	58	45	602
87	287	854	839	1292	780	1322	309	619	401	279	588
849	1350	332	791	351	156	1247	1355	639	157	132	1380
1282	1270	1381	641	375	730	1300	498	388	594	519	1382
689	1313	1258	542	595	162	1383	493	275	252	5	1367
1348	368	776	227	1319	675	413	1338	20	1318	188	338
516	548	568	590	1294	415	1343	609	417	1356	54	1315
223	1323	1316	1357	685	1246	1384	433	41	815	1360	1296
755	1345	316	808	650	310	449	1284	623	1347	1302	101

1364	481	1293	142	170	517	680	1314	1324	1281	1385	317
1321	572	833	1261	141	62	697	305	1330	51	173	1312
669	148	1256	656	92	391	161	382	1251	600	665	805
686	1342	589	1369	534	349	1249	838	1332	1336	521	1272
387	1262	1340	1301	70	552	192	1386	1311	603	1277	766
440	789	638	292	37	1279	1291	1372	725	1387	247	851
1299	86	1388	674	848	1310	1359	57	138	1286	124	1285
1266	1248	809	1252	340	289	1389	775	488	71	694	607
31	1349	628	1263	1390	1255	365	1341	300	1275	262	1309
1250	761	25	50	85	408	256	577	835	841	742	1370
707	712	666	1283	1354	455	1317	386	430	1269	492	1339
14	660	1288	314	1368	1306	151	1353	807	90	1265	788
608	510	1391	1335	327	1304	741	7	1365	197	245	861
1297	1259	80	679	1333	75	1326	1352	779	26	199	270
547	537	636	1289	740	323	326	395	1298	60	1344	302
1392	1337	1271	593	127	63	184	13	513	103	1307	469
1290	1393	1260	1366	1305	1325	1278	1276	107	241	315	240
1363											

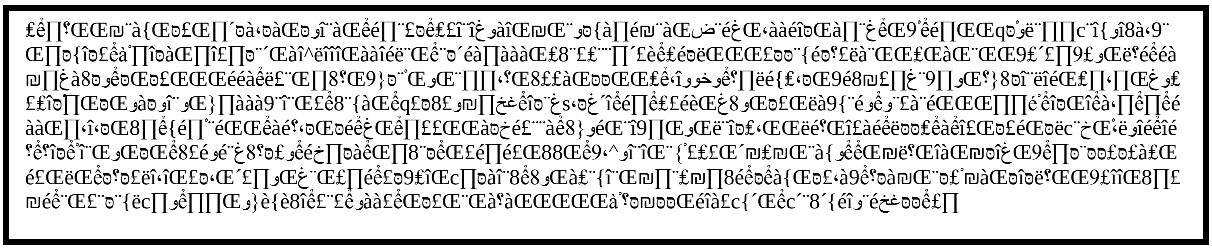
Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets)

7	36	33	79	39	24	81	16	5	73	3	1
20	88	30	44	56	41	11	62	65	47	98	26
814	806	645	389	313	988	737	838	346	946	243	890
300	123	778	333	207	229	663	571	127	816	821	355
1220	1344	1025	1270	1147	1345	1020	1218	1338	1037	1085	1041
1016	1226	1114	1073	1346	1190	1195	1253	1227	90	40	70
15	95	6	53	87	34	91	25	82	76	45	13
93	14	55	28	37	17	72	66	85	61	22	64
21	51	8	77	35	78	86	38	10	42	18	99
31	60	2	27	54	92	94	80	19	71	69	29
67	46	84	89	12	9	96	4	57	58	63	49
59	75	50	23	83	48	526	275	178	752	965	810
453	272	436	898	463	245	321	586	113	506	208	419
708	183	640	451	376	195	372	414	942	895	238	302
443	631	474	712	434	917	277	392	187	467	607	557
845	647	688	188	52	97	630	680	865	449	583	164
262	105	634	317	639	119	928	464	714	802	797	932
823	822	953	684	529	694	270	991	482	154	956	978
440	135	197	670	280	351	987	248	881	779	933	327
606	214	662	695	292	456	909	608	158	285	491	877
311	771	395	891	938	803	563	612	561	963	715	501
447	153	498	134	792	220	215	811	874	427	283	418
538	68	32	43	74	687	345	661	924	284	486	359
839	495	115	893	732	103	307	442	305	675	600	136
157	603	106	417	122	918	441	785	930	982	181	466
255	758	765	794	817	271	854	597	494	900	219	198
484	315	194	218	923	382	667	309	901	408	120	558
204	772	336	761	470	266	519	996	240	582	795	969
168	142	541	559	540	853	564	578	936	916	788	882
274	861	527	365	651	367	707	665	162	435	469	929
589	979	939	290	753	755	776	584	656	269	391	416
860	700	786	514	344	566	492	319	993	139	384	790
256	475	252	437	244	145	276	544	374	598	373	828
258	652	870	685	784	318	801	927	353	109	180	324
689	858	910	108	940	508	525	137	749	697	975	296
489	650	503	329	101	570	326	850	716	581	246	627
403	782	375	556	985	949	126	907	422	179	282	221
161	944	981	604	517	349	212	783	138	906	835	873
594	545	880	655	462	967	539	457	203	528	717	952
341	316	366	813	591	836	299	769	926	951	547	904
848	844	840	800	744	576	819	995	412	413	147	588
812	424	356	404	554	869	619	186	390	150	692	825
740	852	343	601	831	431	312	989	535	481	377	702
764	286	855	682	976	261	515	768	750	254	696	883
654	169	304	618	889	727	710	805	362	799	360	621
543	530	288	428	379	118	587	531	423	833	426	210
342	241	905	787	572	439	560	954	748	653	402	773
394	980	448	649	223	488	760	415	368	959	611	913
287	493	569	729	253	767	132	322	536	479	553	476
648	410	815	964	562	458	323	217	385	830	144	234
173	617	643	143	660	628	461	551	473	193	337	257
644	698	673	293	104	872	757	334	843	826	510	452
289	107	573	213	251	459	516	512	110	998	438	242
974	971	314	471	669	550	620	465	148	580	593	777

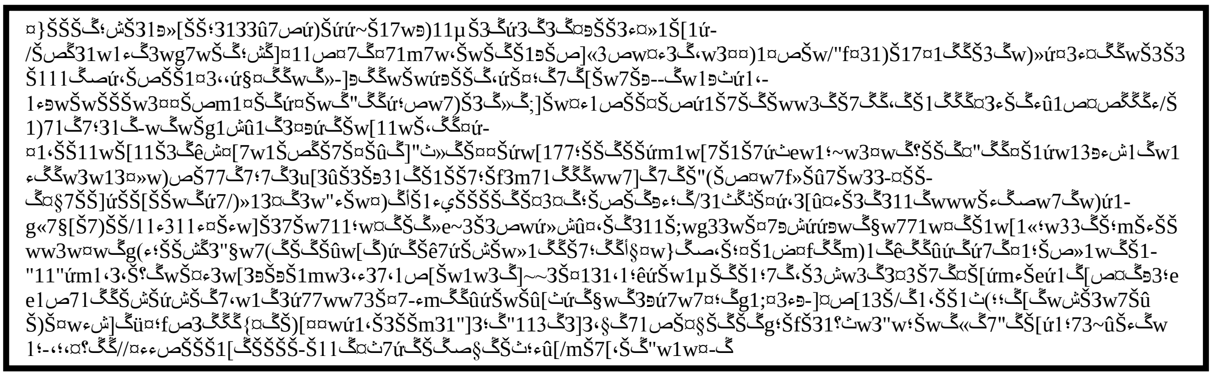
192	990	155	230	614	504	638	983	915	775	842	149
236	363	759	999	140	745	851	885	736	766	941	294
871	867	994	152	914	407	396	555	690	886	585	896
369	613	731	966	249	774	460	170	505	331	970	721
846	807	892	295	577	298	804	961	191	796	948	159
590	791	674	117	742	720	738	490	405	664	599	184
432	166	509	762	672	986	615	542	306	165	130	622
455	728	683	445	820	658	957	605	177	279	176	335
859	129	364	706	202	879	809	502	348	383	657	477
281	947	832	671	857	868	546	595	112	268	233	724
741	228	713	878	908	167	370	887	925	100	325	704
200	211	499	681	626	574	677	763	433	701	624	446
637	354	111	960	841	977	125	483	444	478	972	711
131	808	430	133	387	380	163	468	911	310	897	864
160	513	227	141	945	518	352	718	984	524	487	425
116	781	411	278	636	876	128	320	532	330	629	725
834	454	250	114	291	668	520	534	849	770	185	358
623	973	709	409	303	146	579	259	955	429	175	747
397	950	659	679	239	856	642	610	609	568	301	958
496	297	533	121	937	480	201	730	552	818	500	225
400	182	521	328	894	922	935	522	686	888	189	931
199	511	754	406	124	232	102	332	920	361	398	693
224	386	399	592	350	739	567	703	265	226	919	263
899	222	472	746	156	635	378	507	206	171	273	827
401	866	565	548	338	789	421	992	209	151	676	641
485	884	699	260	393	523	902	388	726	997	968	190
381	751	633	231	264	756	824	357	420	625	450	743
196	216	602	371	237	962	174	235	735	875	837	172
863	829	666	247	719	912	793	339	847	943	646	734
934	347	798	340	205	575	733	267	537	903	678	308
616	862	596	632	549	691	705	723	497	921	780	722
1229	1311	1126	1074	1105	1179	1155	1223	1023	1161	1175	1129
1184	1134	1305	1319	1088	1082	1347	1079	1281	1152	1144	1149
1288	1176	1186	1289	1254	1283	1348	1017	1159	1189	1349	1217
1131	1039	1350	1272	1064	1192	1188	1228	1115	1095	1264	1310
1058	1201	1198	1022	1014	1043	1108	1071	1351	1352	1027	1165
1151	1059	1028	1353	1141	1200	1143	1354	1247	1100	1029	1112
1005	1355	1257	1083	1194	1046	1356	1035	1024	1307	1101	1091
1278	1287	1092	1139	1120	1250	1007	1246	1225	1116	1051	1357
1244	1086	1209	1096	1090	1284	1358	1106	1127	1318	1309	1290
1193	1216	1076	1052	1259	1359	1117	1360	1361	1239	1047	1034
1003	1057	1277	1111	1093	1332	1301	1009	1183	1328	1178	1061
1099	1265	1215	1240	1362	1213	1269	1298	1049	1312	1363	1171
1263	1252	1364	1197	1077	1243	1365	1366	1145	1214	1142	1018
1315	1123	1146	1107	1060	1367	1113	1006	1160	1337	1249	1326
1011	1279	1368	1208	1196	1275	1056	1170	1110	1234	1335	1261
1294	1069	1031	1203	1063	1267	1258	1300	1202	1157	1241	1030
1150	1280	1206	1369	1181	1180	1021	1370	1158	1121	1097	1205
1128	1012	1293	1089	1341	1237	1187	1317	1102	1010	1068	1304
1371	1291	1372	1148	1153	1065	1256	1303	1306	1124	1295	1053
1013	1162	1268	1044	1373	1078	1032	1166	1177	1137	1026	1255
1374	1375	1133	1236	1191	1172	1048	1245	1282	1119	1296	1062
1130	1308	1376	1207	1299	1036	1377	1333	1136	1199	1378	1379
1292	1164	1045	1248	1343	1380	1314	1235	1321	1066	1232	1182
1271	1336	1320	1260	1251	1381	1382	1168	1286	1383	1384	1316
1322	1385	1042	1262	1173	1212	1038	1386	1040	1274	1154	1327
1273	1185	1387	1313	1210	1388	1019	1132	1033	1001	1156	1389
1122	1070	1098	1211	1055	1329	1323	1285	1072	1054	1325	1004
1302	1118	1231	1219	1204	1087	1222	1221	1125	1080	1342	1224
1015	1340	1390	1050	1330	1233	1094	1081	1067	1104	1174	1297
1391	1331	1075	1392	1339	1238	1230	1135	1324	1276	1163	1334
1103	1084	1002	1138	1167	1109	1242	1140	1008	1266	1169	1000
1393											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)



(b)



(c)

Figure V.4. (a) *Donnée originale Texte2**, (b)** Première version chiffrée, **(c)** Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets)

8	2	4	7	6	9	1	5	811	910	899	896
878	885	622	892	767	914	357	901	895	297	771	211
70	851	189	890	897	694	880	913	662	78	908	900
906	210	170	889	879	904	833	407	208	300	498	306
259	891	871	858	469	698	543	87	903	368	52	884
266	835	642	886	92	909	905	882	883	405	527	371
915	658	12	893	113	172	457	898	888	633	887	242
877	881	414	376	911	556	912	907	902	163	648	281
894	876	949	701	174	1155	643	192	649	213	466	1103
514	1076	1022	660	624	3	632	430	623	944	925	770
785	796	67	1194	230	707	799	1177	1218	245	508	1007
179	484	1209	1223	549	1118	1037	1167	740	73	1160	248
193	1088	287	1200	304	131	175	544	511	428	1173	176
1176	931	983	815	951	370	32	955	826	53	945	754
168	863	1009	1219	692	532	459	273	1018	356	730	255
636	604	539	11	984	79	930	817	383	581	829	1148
1133	764	961	1068	1127	1140	1225	630	956	403	1143	1235
822	601	755	1217	535	1073	960	1137	203	325	545	1036
745	1112	971	1097	1168	1232	223	437	805	1062	1204	321
272	374	1080	260	748	86	1045	434	429	1157	675	994
145	1142	587	585	1229	1069	1044	267	386	471	1163	147
1102	1182	747	593	798	1054	1019	82	981	978	348	1066
928	669	1192	1065	1038	311	1196	943	1141	597	610	63
741	1085	1075	234	1063	962	850	682	1156	401	843	670
81	1058	84	1000	151	525	60	1110	225	501	969	552
341	446	486	985	1093	580	372	1187	998	517	65	778

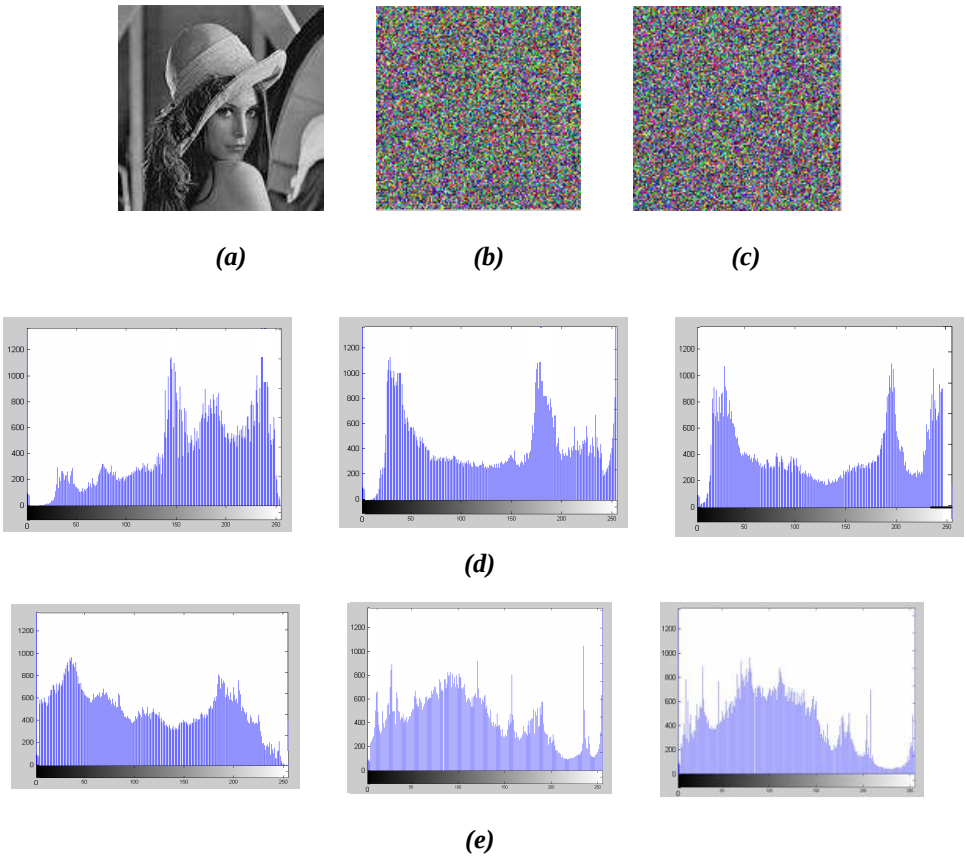
606	137	97	938	180	952	1230	90	35	1067	1084	1083
1014	463	237	814	650	319	935	379	315	309	46	676
1139	1060	1238	240	1152	645	689	846	221	1003	251	116
635	1215	182	28	467	460	171	495	377	674	80	1228
1027	392	452	54	502	957	1064	1079	228	733	932	359
609	997	647	834	1149	1004	751	166	923	830	1166	976
980	1198	1116	975	578	631	75	253	235	1095	1224	860
965	449	1109	43	979	150	763	202	271	252	394	59
516	1205	239	838	874	247	353	685	704	169	83	177
1012	389	114	39	1074	659	924	982	104	565	783	839
705	167	1222	987	656	292	57	93	194	1056	1165	282
861	500	526	564	34	411	922	1189	566	445	506	99
1119	793	257	637	398	1087	1122	1185	105	582	51	494
490	1024	199	207	696	782	596	521	412	138	50	1233
1030	993	1191	258	653	1023	1105	312	757	492	712	419
173	1055	406	948	21	713	119	378	402	929	1186	574
19	654	286	1098	327	246	852	1017	1039	183	592	936
1082	954	332	714	781	328	666	85	518	620	787	1033
20	920	347	750	546	756	1034	1181	1031	62	1035	497
22	522	519	346	1051	591	803	373	1239	1015	927	37
382	966	586	218	140	1124	410	1183	74	595	1180	942
841	1170	872	1125	742	1243	845	316	1104	106	554	159
802	967	1099	710	344	158	196	939	1195	1091	146	614
569	1002	562	1226	641	279	567	416	1178	100	1132	236
823	49	809	296	360	254	575	1013	1048	313	441	801
206	992	1193	157	48	723	1234	280	184	768	615	415
691	824	1213	762	743	753	187	233	404	262	479	1016
422	365	481	1171	661	837	323	963	686	772	605	1227
599	572	777	1072	443	717	1244	1135	1131	735	448	1047
109	657	941	1078	1071	531	865	794	1032	25	107	1100
238	621	120	334	71	795	1077	1001	1159	626	1161	451
342	23	699	1040	919	1145	510	336	715	274	739	468
1231	1172	1237	831	1150	1144	423	14	832	1188	31	1162
68	918	385	338	766	867	1050	528	216	305	577	524
972	709	1081	222	639	399	72	1212	953	1216	124	112
432	1203	1201	126	570	1146	1211	101	426	125	1136	178
209	706	1005	1174	1086	921	926	33	917	358	1011	161
977	450	42	1057	688	728	1111	847	219	456	322	970
197	999	1199	720	366	1175	201	627	1130	320	536	290
1236	995	229	868	478	1214	330	589	1008	625	355	690
968	310	973	337	488	1006	672	77	148	746	808	603
608	1070	343	667	553	354	752	989	946	520	568	628
1061	725	693	1089	110	1117	1151	326	629	1108	1092	294
855	759	16	792	1208	651	391	776	58	1164	256	800
263	916	1120	1179	424	579	547	1115	369	160	974	538
15	89	541	934	947	18	480	1049	1029	465	612	191
958	1028	986	1134	774	548	1138	1210	1197	937	1221	806
1096	542	613	1123	433	678	804	726	76	933	1090	571
1158	1043	959	1094	1041	10	1052	1106	1020	1169	724	1241
862	990	988	269	461	644	711	1153	421	602	677	684
1107	594	1206	1242	164	853	1021	475	277	1113	214	1025
1053	440	491	573	1207	1059	683	134	139	1046	318	864
285	1121	560	1026	1154	397	996	190	509	1190	350	702
1128	417	1042	438	950	673	1010	1114	515	991	665	810
91	869	964	493	188	395	108	844	152	769	335	827
1101	1129	88	435	64	472	1147	807	551	56	1126	719
1202	791	857	136	779	217	1240	444	1184	499	409	1220
668	224	940	299	1343	697	1344	616	442	95	1336	1334
1286	204	646	836	220	489	301	700	1313	288	820	561
773	1297	1345	156	607	473	476	1259	721	133	98	840
1346	351	464	559	1347	384	1348	1304	375	1349	563	1268
436	873	652	439	142	731	1264	314	111	681	393	611
784	1302	558	1270	1350	205	1351	165	13	303	361	1269
1352	734	618	381	732	363	1353	1354	664	557	380	1294
513	94	530	727	1291	244	1355	1338	367	425	454	758
333	1356	1300	1357	1248	339	1342	1341	588	1282	250	317
760	122	1308	512	1358	69	1299	129	102	1251	1359	534
331	788	1298	396	1360	268	264	103	427	482	1325	1307
523	1361	329	1324	1362	1363	1290	1364	859	455	155	663
1340	1333	819	261	1365	1366	1329	1328	786	293	284	1367
154	1331	270	1265	849	132	27	135	729	130	749	505
1368	231	638	1369	390	1370	345	540	1371	1278	1293	276
1372	1250	308	737	17	1373	241	298	362	1262	198	127
447	153	474	1272	1252	825	349	121	1315	1303	215	47
1337	1305	249	1260	186	775	671	1320	387	1374	291	870
289	1375	400	41	227	1289	144	30	55	1339	744	1316

1296	128	1319	504	1376	1245	1377	529	1378	848	431	485
1255	418	1254	708	352	1267	212	232	640	1246	1314	458
1379	722	1327	1380	634	181	118	797	123	590	1280	679
1381	1326	1258	195	1271	507	38	1382	66	1279	453	1256
1309	1383	738	483	340	655	736	533	1312	1384	420	364
842	149	780	1263	1332	1275	1274	283	816	1284	1330	1266
866	1385	117	598	818	703	200	716	408	1285	854	576
295	1386	1292	115	695	1249	1318	40	185	302	1306	1283
1288	45	487	1323	875	143	600	477	583	1281	1253	550
1387	278	619	828	324	1257	813	812	24	790	503	496
36	1388	1389	243	96	1301	1321	718	555	789	1247	388
26	1261	821	1276	162	761	1322	687	275	765	1390	1287
617	1310	141	265	44	1277	226	1391	1392	1311	29	1393
1317	856	584	1295	680	537	462	307	413	1273	61	1335
470											

Clé de chiffrement de la deuxième version chiffrée de Texte2 ($Taille_{Clé}$ = 1,8704 K octets)

681	257	54	272	127	416	212	193	126	445	404	584
108	735	917	617	694	270	386	241	702	562	898	175
381	459	904	737	596	689	130	228	326	836	460	289
37	958	638	703	356	817	741	466	971	858	1	909
688	524	53	13	448	624	634	199	369	807	116	686
102	121	204	490	397	187	670	650	522	163	256	950
868	720	153	510	660	580	68	427	698	546	523	900
621	306	312	331	721	488	314	243	350	313	452	325
844	986	82	499	494	48	12	991	526	453	479	910
978	56	409	607	172	240	620	627	391	779	15	641
174	931	71	90	55	265	963	947	981	60	84	250
95	39	977	513	748	220	235	516	507	916	401	190
360	213	362	725	447	333	105	491	857	78	881	383
89	934	705	999	123	426	231	293	323	91	943	557
803	846	324	945	791	718	826	903	396	258	839	602
713	83	859	38	639	411	400	201	359	545	487	92
138	766	879	799	346	558	928	864	49	961	365	519
149	146	371	18	281	158	430	684	277	885	754	58
374	435	269	318	358	850	335	953	567	443	970	899
169	719	291	789	809	613	432	595	275	882	247	347
62	282	851	685	97	775	151	814	601	384	439	771
980	343	552	598	106	853	541	192	612	993	687	154
271	461	496	125	708	776	948	744	939	375	232	952
286	927	675	671	483	575	373	455	155	305	421	761
731	279	869	288	642	933	788	367	122	676	46	997
35	31	830	902	873	480	736	402	751	477	336	654
450	704	696	295	236	534	244	944	988	629	230	861
341	100	433	906	786	390	750	905	307	42	351	659
714	301	395	34	205	549	77	438	215	285	454	662
871	883	949	884	165	573	935	398	728	112	888	5
985	304	412	440	505	514	219	292	920	131	965	72
349	509	833	287	157	707	308	47	472	810	492	144
339	489	734	139	233	223	474	261	533	209	987	656
912	994	475	911	968	98	772	506	553	214	610	469
132	202	159	568	925	983	938	237	224	669	777	693
538	964	891	746	431	464	604	227	722	756	773	554
50	527	142	252	529	334	171	712	465	437	515	255
874	57	758	540	161	760	733	503	792	27	486	532
822	385	462	234	407	299	583	210	446	88	544	757
436	319	700	109	478	614	393	732	246	599	556	316
682	753	767	70	586	423	796	361	570	793	177	913
73	8	414	30	372	81	709	429	456	79	64	936
508	878	578	63	780	329	442	135	110	457	597	226
870	512	876	458	812	866	266	942	765	768	302	26
330	69	76	537	724	907	3	946	608	500	561	872
4	424	23	535	865	181	120	493	539	315	160	93
589	588	611	701	200	216	661	531	419	115	677	273
399	982	380	188	890	382	24	377	749	655	194	922
673	87	653	823	267	740	85	221	44	582	310	189
813	956	834	695	995	366	410	889	600	353	140	425
590	783	838	908	672	185	819	806	918	501	892	816
955	229	930	128	778	536	542	413	560	484	559	504
975	99	941	186	867	420	984	476	518	66	606	622
405	976	141	937	592	797	378	631	485	211	33	566
666	521	723	28	635	297	957	441	7	683	251	992
626	511	101	591	21	665	716	636	996	551	548	274
585	363	979	332	96	117	609	249	129	923	644	710

637 517 738 51 43 357 969 563 222 203 322 547
470 860 344 798 355 863 248 628 921 845 841 184
847 502 65 565 134 886 848 951 940 471 594 406
774 818 663 45 468 14 645 972 303 959 770 337
855 260 124 752 824 623 619 574 451 781 962 9
825 368 498 640 196 428 877 473 166 197 463 354
415 764 118 815 587 739 276 183 67 321 94 298
564 577 136 785 897 801 444 36 808 364 895 657
820 759 179 311 495 658 147 742 376 691 16 579
167 896 370 434 625 804 467 571 805 745 387 379
664 6 2 690 652 795 967 300 422 726 827 837
572 191 831 61 150 901 225 706 924 327 729 593
320 253 17 152 543 667 954 715 254 893 697 137
263 180 11 143 74 990 19 278 875 763 119 345
605 264 408 22 576 699 802 497 309 854 929 86
680 678 842 168 482 113 394 649 679 727 296 782
835 787 615 618 338 389 25 743 692 52 550 843
114 919 207 828 80 40 284 178 730 10 238 966
242 569 794 832 103 755 392 182 75 481 849 647
821 259 974 104 294 880 418 973 603 674 998 530
616 59 32 262 348 651 195 20 133 340 894 632
156 388 403 643 829 176 581 960 29 245 198 217
762 711 717 41 239 862 747 668 148 914 555 206
449 926 280 352 417 111 932 852 342 784 633 173
887 989 162 107 218 170 328 646 208 790 915 528
290 811 769 630 648 317 840 520 283 145 164 268
525 856 800 1319 1015 1163 1032 1115 1334 1035 1004 1255
1286 1101 1309 1186 1325 1012 1335 1304 1336 1136 1337 1107
1172 1113 1249 1190 1227 1058 1010 1251 1111 1023 1338 1127
1302 1114 1033 1100 1228 1128 1112 1031 1020 1299 1109 1311
1339 1226 1081 1202 1076 1054 1295 1340 1069 1341 1198 1216
1342 1183 1343 1002 1148 1070 1176 1328 1303 1003 1121 1029
1277 1229 1089 1230 1256 1091 1175 1060 1042 1119 1182 1188
1074 1050 1132 1224 1333 1170 1078 1294 1168 1344 1149 1320
1239 1162 1203 1223 1052 1310 1269 1233 1210 1083 1265 1307
1275 1298 1045 1326 1345 1346 1272 1048 1317 1347 1261 1030
1263 1049 1285 1165 1348 1276 1349 1350 1027 1201 1235 1262
1189 1351 1313 1005 1063 1017 1195 1000 1150 1352 1016 1353
1062 1305 1130 1024 1354 1221 1073 1355 1179 1126 1356 1205
1161 1244 1240 1001 1129 1022 1197 1284 1087 1080 1006 1291
1137 1147 1152 1055 1181 1268 1252 1250 1327 1053 1160 1044
1134 1357 1072 1037 1151 1040 1225 1093 1008 1358 1056 1192
1246 1359 1271 1360 1047 1315 1090 1361 1013 1156 1362 1204
1157 1041 1082 1096 1184 1105 1123 1329 1363 1281 1097 1118
1241 1215 1364 1146 1288 1258 1365 1131 1366 1034 1092 1297
1300 1098 1367 1280 1368 1248 1018 1142 1245 1120 1177 1211
1267 1369 1173 1370 1371 1057 1372 1122 1214 1159 1237 1138
1021 1038 1166 1217 1064 1180 1208 1283 1106 1065 1185 1373
1278 1314 1374 1200 1322 1071 1220 1219 1316 1145 1169 1222
1375 1293 1067 1059 1104 1019 1260 1242 1257 1133 1376 1125
1043 1377 1167 1378 1379 1308 1140 1079 1085 1077 1232 1103
1380 1290 1381 1154 1331 1124 1266 1199 1253 1116 1270 1209
1158 1036 1102 1046 1382 1306 1095 1007 1088 1231 1108 1193
1238 1206 1039 1236 1187 1274 1383 1282 1094 1171 1384 1075
1264 1084 1254 1196 1259 1164 1155 1025 1385 1178 1386 1068
1086 1387 1318 1243 1207 1026 1321 1324 1323 1028 1213 1061
1218 1191 1212 1330 1388 1066 1332 1234 1117 1051 1153 1301
1273 1099 1389 1390 1144 1139 1143 1009 1247 1287 1014 1135
1391 1279 1174 1141 1392 1312 1110 1292 1296 1289 1011 1194
1393



(a) ***(b)*** ***(c)***

(d)

(e)

Figure V.5. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

369	423	529	171	397	412	474	15	689	64	112	390
267	21	118	304	6	28	574	465	356	338	717	638
339	110	243	458	508	450	361	608	635	115	593	749
151	742	575	716	428	109	486	584	294	722	521	141
87	695	373	225	452	602	679	13	208	85	468	634
709	298	70	297	514	601	439	359	320	251	222	292
319	765	456	614	142	645	625	288	383	217	125	476
396	718	274	236	729	483	567	114	487	492	303	542
426	628	675	713	296	505	370	730	409	395	764	641
669	460	337	264	230	204	623	358	293	418	150	520
134	328	469	762	440	533	651	551	535	286	438	462
103	355	347	63	158	16	561	205	410	643	703	432
519	693	247	620	346	40	736	366	129	652	164	327
371	506	363	154	241	447	434	624	585	163	748	376
380	144	407	59	143	667	252	10	536	548	766	360
335	455	311	497	200	466	249	751	35	119	165	341
36	741	398	448	656	18	65	233	734	17	531	512
595	660	104	586	81	435	299	101	92	309	454	436
710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579
352	739	698	306	253	577	248	735	515	342	285	61
219	138	568	705	658	427	511	532	210	357	307	153
633	711	98	433	619	691	60	20	137	733	485	324
659	419	719	684	269	392	754	68	259	284	467	295
48	300	224	541	600	715	630	613	172	66	425	558
604	38	692	34	557	263	258	384	51	317	375	111
325	146	701	472	388	607	289	94	644	377	745	580
500	598	496	240	443	594	364	554	74	637	187	712
655	747	576	145	47	501	417	490	622	162	491	626
578	699	750	14	128	493	282	226	459	351	589	209
582	545	740	596	147	275	121	362	470	53	193	291
333	290	179	523	527	42	30	402	738	525	113	374
664	646	507	609	685	756	189	700	148	260	654	464

191	350	378	176	257	538	599	603	11	702	372	571
237	12	96	170	192	653	88	416	540	25	180	522
642	166	72	246	194	707	680	33	639	159	215	75
728	227	516	681	344	90	99	559	127	513	265	7
23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555
69	708	345	494	271	238	726	393	534	132	385	587
686	647	368	203	3	481	482	332	199	167	687	560
539	495	743	44	583	91	610	690	336	31	261	403
379	255	725	498	323	386	198	572	220	484	73	322
677	606	340	272	343	58	503	381	26	421	517	761
79	676	334	348	760	49	663	235	597	566	313	627
39	414	618	518	479	55	752	186	694	563	649	175
223	133	549	420	277	510	727	305	391	228	632	404
196	206	84	52	184	755	212	581	763	429	107	697
724	590	77	737	445	71	682	502	509	120	254	413
758	640	149	616	43	195	242	239	211	611	453	757
218	86	19	310	387	2	105	117	424	244	504	116
449	591	8	229	605	399	629	182	57	461	314	588
177	234	190	102	106	714	122	617	612	216	673	312
431	326	621	80	382	721	93	45	156	665	553	767
475	174	329	394	415	547	188	331	543	365	308	46
67	670	330	157	354	302	389	480	367	232	768	27
207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631
41	471	131	201	245	54	202	444	556	544	720	270
32	688	759	442	279	478	136	197	123	273	671	437
262	457	672	268	349	95	50	488	706	546	183	221
178	160	528	696	62	9	214	615	573	489	430	97
744	22	753	524	446	213	746	731	181	231	283	661
570	401	276	78	400	4	562	526	683	301	89	477

Clé de chiffrement de la deuxième version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

156	453	55	623	217	430	383	574	180	389	257	147
405	685	168	355	721	754	738	227	187	474	143	551
127	578	506	528	477	713	723	319	702	417	384	67
512	575	469	301	86	237	703	204	366	2	268	247
340	452	642	21	12	343	128	214	390	220	262	76
495	281	286	66	397	169	445	170	219	24	48	531
261	338	478	282	635	44	191	209	735	323	410	514
255	234	43	54	516	291	97	344	720	695	129	332
461	117	522	270	89	537	298	161	515	346	758	737
253	35	75	739	240	719	131	27	548	98	455	195
78	683	710	178	595	387	400	403	633	353	350	250
687	37	462	235	119	53	200	385	192	530	179	419
392	488	61	457	45	125	1	576	58	312	552	509
546	706	276	241	617	73	585	681	751	753	517	677
399	91	473	557	701	26	302	637	632	32	565	755
135	539	47	266	110	505	17	631	647	727	303	577
427	411	493	23	743	458	259	122	285	114	489	544
182	72	582	194	264	120	439	279	669	718	292	207
101	407	446	158	475	536	284	630	215	724	25	256
172	622	650	734	87	638	450	521	416	665	762	519
700	251	491	335	287	608	742	34	426	104	370	290
7	315	77	480	330	486	183	619	730	19	331	141
599	555	229	202	359	731	757	273	504	308	167	155
230	542	654	304	165	423	360	733	705	662	648	763
28	433	92	664	694	716	175	760	656	190	171	415
263	627	470	51	100	589	4	707	311	148	88	696
16	466	498	590	111	224	363	468	239	408	162	621
36	333	373	626	561	113	341	70	717	511	379	3
371	109	508	378	463	673	693	221	367	30	388	8
212	750	748	667	374	33	358	661	289	116	447	674
566	173	364	747	314	401	500	185	328	140	634	198
676	449	527	490	260	106	588	79	672	68	759	502
591	142	443	94	420	581	698	108	471	572	296	188
436	38	395	437	545	614	163	525	605	704	121	160
587	318	294	764	307	615	107	761	337	151	422	604
326	69	412	223	300	213	472	249	678	690	580	658
765	11	57	573	99	124	184	523	402	501	226	479
649	361	137	193	618	181	174	39	248	299	579	484
62	670	345	351	636	382	347	454	41	118	644	675
74	620	745	218	252	610	265	438	280	564	60	362
404	503	271	712	418	569	550	406	236	13	600	603

96	82	547	126	568	369	199	534	602	612	485	596
365	31	441	203	682	49	216	729	85	456	722	154
641	258	254	524	668	483	616	435	598	317	532	549
768	613	46	465	628	680	460	225	297	186	246	645
321	133	459	520	196	584	145	413	59	692	231	375
688	102	444	210	293	322	643	714	274	242	339	201
749	65	325	767	691	736	726	424	766	601	144	376
349	159	429	130	583	233	18	529	709	646	625	208
269	380	377	64	594	487	451	244	639	305	176	711
20	112	756	653	71	684	464	157	29	563	189	606
553	651	699	232	50	352	138	728	497	467	152	310
103	10	507	327	715	586	245	434	15	14	746	640
205	81	56	541	526	309	177	393	134	663	562	741
725	368	597	391	394	348	272	288	238	660	40	153
136	533	295	396	571	744	84	146	342	425	164	629
123	689	567	52	320	708	166	740	316	275	559	611
481	686	267	657	93	556	197	752	386	5	336	494
139	554	428	90	83	732	313	592	671	243	448	431
518	132	95	496	277	607	535	80	306	22	540	381
570	372	6	334	652	356	211	659	149	543	499	421
228	278	492	513	357	329	679	432	482	510	115	655
9	624	283	63	150	666	354	324	476	697	409	609
440	414	593	105	442	560	558	206	222	42	538	398

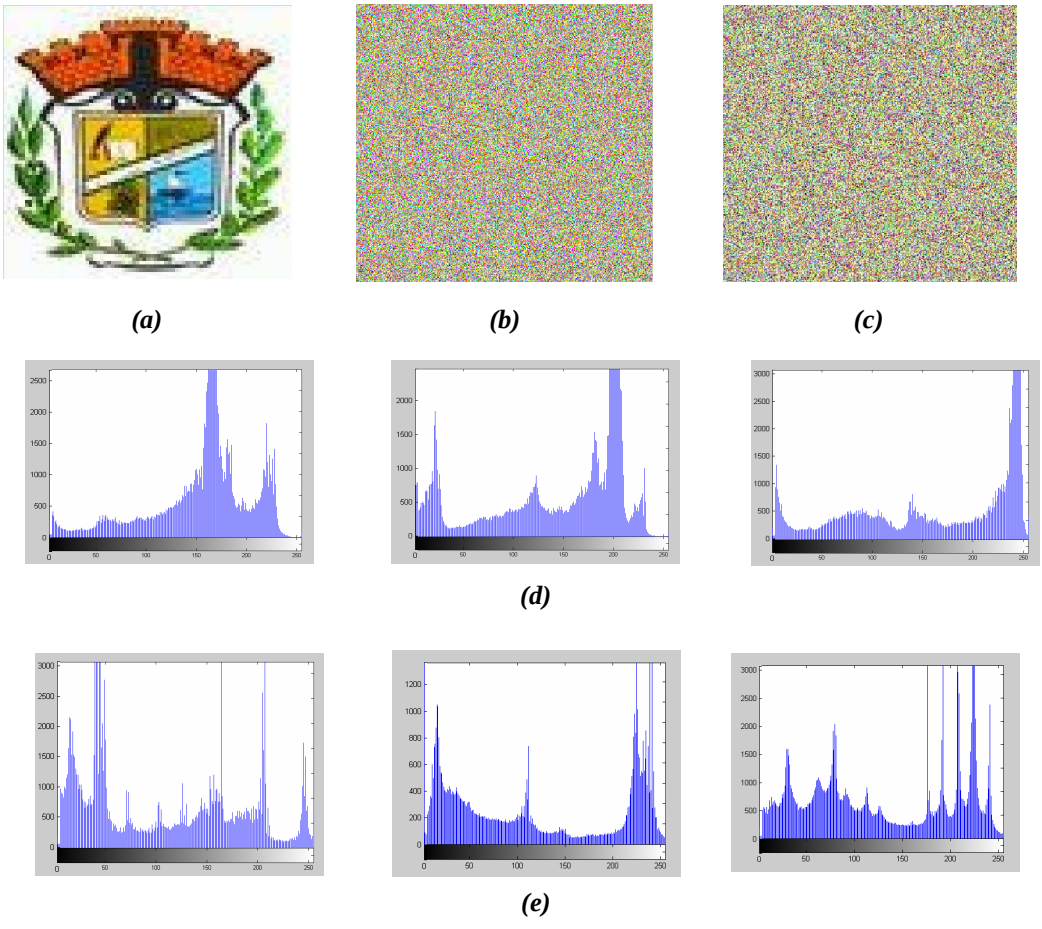


Figure V.6. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

65	9	12	62	15	21	94	7	28	11	81	3
61	36	78	70	91	25	56	97	85	4	30	24
83	32	57	40	54	31	63	8	95	26	75	55
19	96	20	38	49	92	47	44	1	98	52	89
35	67	6	72	69	53	74	60	14	43	84	23
22	82	88	73	90	76	39	13	59	33	58	68
41	50	86	46	5	51	27	99	79	10	71	66
87	48	16	2	45	64	42	80	17	34	77	29
37	93	128	625	414	134	459	456	642	446	447	205
635	230	755	387	503	179	646	371	252	744	662	574
575	475	768	734	494	638	389	441	312	571	445	214
367	308	498	643	659	705	175	676	237	530	674	370
527	626	415	330	333	472	600	238	339	137	730	257
262	144	660	139	217	323	739	329	578	420	618	735
463	199	729	207	340	336	409	551	295	296	477	222
468	276	151	258	725	178	764	245	132	366	525	305
196	717	363	484	658	136	342	397	408	198	598	443
589	455	256	239	287	216	181	290	338	111	334	751
345	392	560	697	355	585	418	18	610	162	534	154
467	241	250	694	464	461	614	500	743	170	244	741
404	259	211	645	318	650	593	442	187	282	639	288
648	267	655	220	573	436	552	344	164	225	727	165
226	667	693	416	116	265	536	532	686	522	471	587
316	260	486	514	488	268	601	628	767	365	382	314
528	195	493	430	108	227	212	145	194	753	716	675
547	152	517	374	182	539	356	677	724	760	377	434
546	119	702	204	590	481	394	114	509	395	301	419
473	490	656	728	168	583	538	426	745	518	149	141
131	279	644	140	580	701	325	348	341	285	203	298
651	263	723	235	462	691	622	722	129	437	354	163
335	398	647	122	469	603	553	737	269	511	101	240
110	361	400	402	317	186	733	277	766	142	264	172
399	613	703	588	231	315	106	669	531	380	233	180
319	504	275	113	425	153	372	748	424	281	157	234
249	581	466	200	289	533	411	630	579	223	221	376
679	569	596	188	421	758	519	146	427	243	123	331
671	508	678	206	304	364	273	440	561	324	297	713
670	274	118	576	715	611	752	620	379	489	369	750
343	278	103	479	621	485	224	526	453	229	604	352
765	653	413	714	499	169	595	762	746	465	410	403
537	100	609	327	117	683	617	570	740	492	373	417
310	505	155	543	246	390	689	102	747	710	594	616
497	381	406	756	563	548	487	632	757	720	512	636
749	474	631	294	433	476	347	661	197	718	393	709
577	174	712	665	592	328	599	435	311	375	719	385
280	480	299	261	292	423	242	159	283	696	542	761
641	349	582	458	565	171	432	431	731	520	732	597
541	605	666	209	368	202	612	337	680	156	272	652
619	633	657	135	125	150	148	506	255	495	104	313
535	540	401	521	412	320	554	634	572	684	654	332
523	627	449	690	183	460	251	682	602	664	742	685
470	309	161	350	515	266	452	640	501	444	567	606
502	429	322	483	711	454	428	726	357	544	378	177
184	291	407	763	624	591	201	218	510	391	556	124
478	293	566	557	623	629	189	721	321	219	270	232
208	302	708	608	586	362	213	388	672	707	248	568
307	271	405	130	115	663	185	253	143	681	698	545
176	422	754	695	450	396	668	109	439	210	607	105
692	673	759	637	112	516	699	121	584	286	736	358
107	147	326	353	562	228	127	513	215	687	555	491
700	649	529	190	738	191	507	138	615	704	383	524
496	193	359	133	384	564	306	550	438	173	236	386
549	167	346	120	448	457	126	300	558	192	254	451
559	482	706	303	284	160	247	688	166	351	158	360

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

498	116	41	239	486	387	497	89	154	262	303	25
129	485	272	506	489	496	446	502	195	127	234	350
240	282	491	338	501	304	256	484	29	209	382	212
238	512	511	494	335	166	495	79	437	381	299	499
480	508	505	481	358	73	482	483	492	392	427	359
507	269	510	236	487	493	243	372	167	231	182	37

94	106	490	87	333	500	71	64	16	504	488	160
503	438	97	227	1	317	336	331	324	76	150	478
509	90	203	716	401	386	194	442	477	328	529	4
762	474	132	3	709	407	735	376	86	596	2	48
651	228	704	200	345	380	201	626	18	5	23	24
385	660	531	732	145	248	562	677	373	663	725	648
736	712	453	84	365	52	569	128	170	183	526	754
43	294	746	146	224	83	112	745	434	448	764	582
400	541	436	635	339	597	390	632	717	26	738	551
383	721	237	93	156	144	463	623	375	627	727	130
152	173	728	731	281	722	220	68	707	656	153	204
193	680	252	54	95	550	429	465	759	70	542	34
55	528	149	414	763	590	232	729	218	646	639	600
458	49	472	57	678	630	664	374	593	533	750	585
190	344	384	117	631	676	539	205	208	300	577	657
697	196	696	394	141	689	131	559	445	513	99	69
669	706	693	578	32	142	756	247	169	768	100	742
217	290	59	264	393	755	330	701	703	332	743	726
266	444	221	549	199	163	659	370	270	250	621	702
268	614	740	552	21	96	435	766	245	622	326	28
278	417	316	752	15	343	733	634	430	618	311	181
340	189	325	532	557	624	765	636	747	9	470	571
140	595	341	538	178	710	308	402	296	573	63	271
125	610	553	449	215	699	20	411	519	263	692	357
690	162	464	439	471	126	666	105	242	720	147	681
604	420	12	523	714	615	739	11	35	155	184	136
192	426	118	318	546	705	368	580	65	253	283	186
682	460	53	670	719	399	724	440	749	364	450	222
45	202	561	121	389	19	109	164	47	261	223	246
77	558	408	82	476	179	423	535	56	616	8	139
38	760	80	406	521	405	527	298	441	249	454	396
723	447	708	748	91	348	684	120	180	715	314	210
617	655	751	658	683	312	391	168	607	61	598	33
404	424	673	570	601	418	295	642	346	613	694	643
159	661	111	257	50	327	287	674	589	289	39	62
219	285	443	143	30	640	274	462	757	362	124	310
397	753	60	259	214	254	216	81	92	412	548	574
555	293	276	534	695	667	671	605	473	302	665	176
433	594	255	265	591	587	516	456	267	206	518	455
369	319	306	98	431	603	292	395	291	378	602	103
191	638	107	688	40	174	581	356	85	102	469	679
711	347	46	213	409	119	138	566	547	371	342	377
211	628	641	165	554	609	425	78	421	413	108	575
525	612	568	10	280	114	349	520	536	468	517	652
451	649	88	197	198	416	226	110	134	157	185	172
592	556	337	415	66	524	734	75	241	279	113	698
744	654	315	422	351	27	515	320	687	611	686	700
297	572	653	650	361	543	629	761	14	475	619	175
586	51	579	288	355	540	466	637	244	730	606	564
353	36	691	235	230	563	379	398	44	522	567	576
7	403	718	388	363	685	565	758	599	329	133	273
137	459	229	122	625	584	17	275	322	251	187	367
410	334	645	560	158	123	633	583	305	101	151	741
428	537	457	452	313	354	207	737	277	366	675	233
258	309	148	620	479	647	58	161	352	767	323	22
42	432	467	713	588	225	668	177	321	419	104	6
360	188	644	301	135	74	72	461	672	31	307	286
171	67	260	284	530	545	608	544	115	13	662	514

Le tableau suivant (tableau V.2) représente une récapitulation des résultats de chiffrement des données tests présentées ci-dessus :

			Taille donnée (éléments)	Taille clé (bits)	Efficacité	Temps chiffrement (s)	Temps déchiffrement (s)
Données texte	Texte1	Version-Chiff1	1084	15323	15784.0	28.35	2,06
		Version-Chiff2			15030.0	28.02	2,13
	Texte2	Version-Chiff1	1151	15323	19498.0	29.16	2,67
		Version-Chiff2			18712.0	28.97.	2,65
Données images	Lena	Version-Chiff1	131 X 131	7680	139362.0	14.62	3.7
		Version-Chiff2			134074.0	14.21	4.01
	Logo	Version-Chiff1	420 X 395	7680	246426.0	15.06	3.11
		Version-Chiff2			225688.0	15.28	3.81

Tableau V.2. *Résultats obtenus par TabuCrypt.*

V.4. Discussion et évaluation des résultats

La première chose à remarquer d'après les résultats présentés dans la section précédente, est que le temps de chiffrement est proportionnellement dépendant de la taille de l'image à chiffrer. En effet, le temps de chiffrement de l'image Lena (34.826 secondes en moyenne) est plus petit par rapport au temps de chiffrement de l'image Logo (35.59 secondes en moyenne), du fait que la première est plus petite que la deuxième. La même remarque est valable pour le cas des deux données texte Texte1 et Texte2. Cette différence en temps revient au fait que le temps de chiffrement englobe le temps de lecture et de codage de la donnée à chiffrer en plus du temps de calcul, proprement dit, de la solution (temps de la recherche tabou). Le premier (temps de lecture et de codage) est strictement dépendant de la taille de la donnée à chiffrer, tandis que ce n'est pas le cas pour le deuxième (temps du calcul tabou). Donc, c'est surtout le temps de lecture et codage qui fait la différence en temps de chiffrement global d'une donnée par rapport à une autre.

De même, du coté du temps de déchiffrement c'est la phase de décodage de la donnée chiffrée qui fait la différence en temps de calcul.

Maintenant et en comparaison avec nos algorithmes développés et présentés précédemment à travers les deux précédents chapitres, le temps de calcul de ce dernier algorithme se trouve meilleur que le temps de calcul de certains de ces algorithmes et plus mauvais que le temps de calcul de certains autres algorithmes. La figure V.7 positionne le temps de calcul de l'algorithme *TabuCrypt* parmi nos quatre algorithmes PosESecL1, PosESecL2, OEEA et *AntCrypt*.

D'après le résumé de temps de calcul de nos cinq algorithmes développés par utilisation des trois métaheuristiques d'algorithmes évolutionnaires (PosESecL1, PosESecL2 et OEEA), de colonies de fourmis (*AntCrypt*) et de recherche taboue (*TabuCrypt*), nous constatons que *TabuCrypt* possède un temps de calcul meilleur que celui de PosESecL1 et de PosESecL2 mais plus grand que celui de OEEA et celui de *AntCrypt*.

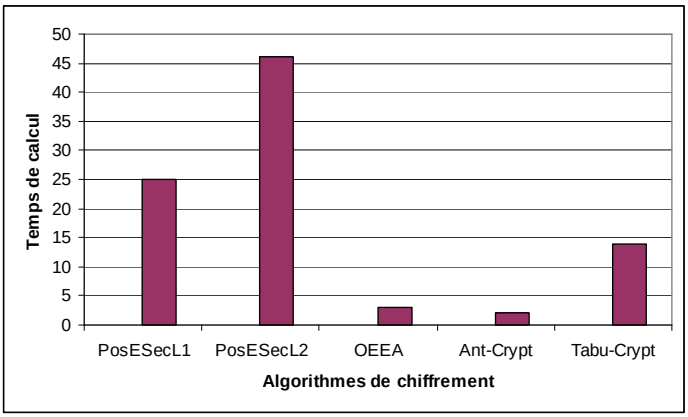


Figure V.7. *Comparaison des temps de calcul.*

Et comme *TabuCrypt* est un algorithme bâti par exploitation d'une métaheuristique, et nous l'avons expliqué précédemment, l'adaptabilité et l'efficacité d'application de telles méthodes dans un tel domaine à cause de leur large utilisation de l'aléatoire a été la source de notre motivation d'application de métaheuristique. L'algorithme développé sera, ainsi, doté d'un grand pouvoir confusionnel pour compliquer le plus possible la tâche de cryptanalyse.

En effet, l'aspect non déterministe innové à travers nos approches proposées, et par conséquent celle, ici, présentée, représente l'un des points forts de l'algorithme *TabuCrypt* tout comme nos autres algorithmes leur assurant, ainsi, une résistibilité contre les attaques différentielles. Ceci est assuré par une élection aléatoire, sur l'ensemble des générations, des voisins des éléments codant une solution intermédiaire pour calculer une autre solution intermédiaire ou une solution finale qui soit satisfaisante.
De même, l'attaque statistique sera presque impossible à appliquer grâce à ce dernier point en plus du fait que les données originales et leurs version chiffrées sur une instance, seront presque toutes différentes comme le montre le tableau suivant (tableau V.3) résumant les valeurs d'application des mesures de similarité NPCR, MAE et MSE.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.8927	0.97	0.68
Lena-version chiffrée 2	0.9104	1.05	0.71

Tableau V.3. *Niveaux de confusion de Tabu-Crypt.*

De son tour, l'attaque exhaustive sera mise à l'écart grâce à la taille de clé qui est largement sécurisée et qui est de taille égale à 15323 bits dans le cas de manipulation de données texte et qui égale à 7680 bits dans le cas de manipulation de données images.

Les attaques fréquentielle sont aussi pénalisées du fait, d'un coté, que ces dernières étudient les fréquences d'apparition des caractères dans un message (précisément un texte) écrit suivant une langue naturelle, alors que notre algorithme *TabuCrypt* peut traiter toutes sortes de message constitué de caractères du code Unicode. D'un autre coté, le message chiffré peut être constitué de caractères complètement ou partiellement différents de ceux constituant le message en clair.

V.5. Conclusion

La recherche Tabou est une méthode d'optimisation qui utilise la notion de voisinage d'une solution. Cette méthode permet, à partir d'une solution initiale, de visiter un ensemble de voisins. Ces voisins seront évalués et permettent l'évolution, à travers des règles spécifiques, vers une solution finale.

Ainsi et dans ce chapitre, nous avons présenté notre nouvel algorithme de chiffrement tabou, *TabuCrypt*, qui opère suivant le mode de chiffrement à base d'occurrence expliqué et validé dans le troisième chapitre.

Nous avons présenté en détails les différentes étapes du processus développé et nous l'avons testé sur différentes données test. Ainsi, notre objectif qui se résume en l'exploitation d'un voisinage en recherche locale de type taboue pour la manipulation de données texte et images, est atteint dans certains cotés vu les qualités de la méthode tabou (simplicité du principe, performance…) en donnant lieu à des résultats de bon niveau de confusion. Toutefois, le temps de convergence du processus de résolution est relativement lent par rapport à certains de nos précédents algorithmes proposés : OEEA et *AntCrypt*. Cela revient à l'aspect parallèle encapsulé à travers le principe soit des AEs ou des algorithmes de colonies de fourmis.

Conclusion et perspectives

Au cours de cette thèse nous avons proposé une nouvelle approche de cryptage de données texte et images par exploitation de métaheuristiques parmi lesquelles, nous nous sommes intéressés à celle inspirée du monde biologique et désignée par « approches biomimétiques ». Il s'agit des algorithmes évolutionnaires inspirés des principes de l'évolution naturelle des espèces. Par la suite, nous avons abordé une approche inspirée de modèles d'organisation naturels observés dans les sociétés animales, en particulier, dans une colonie de fourmis. Ces deux méthodes exploitées appartiennent à la catégorie des métaheuristiques à population. De même et pour l'autre catégorie qui est celle des métaheuristiques à trajectoire, nous sommes aussi arrivés à proposer un autre algorithme de cryptage utilisant le principe d'une recherche tabou.

Ainsi et après avoir étudié un panel assez diversifié des techniques de cryptage et avoir analysé les problèmes de sécurisation du transfert, nous avons élaboré nos propositions de stratégies de résolution d'un tel problème. Autrement dit et afin de détailler nos apports, notre thèse a été organisée autour de deux parties dont l'objectif de la première est de situer le lecteur dans le contexte de notre travail pour lui permettre de suivre les démarches réalisées dans cette étude. Dans la deuxième, nous avons détaillé la conception, la réalisation et l'évaluation de l'approche de résolution proposée.

Notre première contribution a porté sur l'élaboration et la conception de nouveaux algorithmes de chiffrement efficaces et robustes qui traitent le problème de taille de clés leur permettant, ainsi, d'être loin des attaques exhaustives.

Pour ce faire, nous avons commencé par une étude approfondie du problème tout en essayant de le ramener en un problème d'optimisation par une formulation adéquate aux métaheuristiques à appliquer : métaheuristique des algorithmes évolutionnaires, métaheuristique des algorithmes de colonies de fourmis et celle de recherche tabou. Pour la première, les obstacles à franchir étaient surtout la définition d'un codage adéquat de solutions. Dans ce sens, deux modes de codage différents ont été proposés : codage à base de positions et un deuxième à base d'occurrences donnant naissance, ainsi, à trois algorithmes de chiffrement, PosESecL1, PosESecL2 et OEEA de qualités différentes dont celui opérant suivant le mode de chiffrement par occurrences (OEEA) était le plus efficace et le plus robuste.

L'application de la deuxième métaheuristique a démontré l'efficacité du codage à base d'occurrences. À ce stade aussi, il a fallu définir avec précaution certains paramètres tels que la fonction d'évaluation, le mécanisme de manipulation de phéromone (incrémentation et évaporation) ainsi que celui de génération de la clé de chiffrement qui doit doter l'algorithme développé, *AntCrypt*, d'une résistance contre les attaques exhaustives vu que cet algorithme est de type symétrique.

De même, l'exploitation de la troisième métaheuristique a, aussi, abouti au développement d'un cinquième algorithme de chiffrement, *TabuCrypt*. Cela a nécessité la projection des composants d'une telle métaheuristique sur le problème de cryptage, à savoir la

liste tabou, stratégie de choix de voisins, table de hachage et les mécanismes avancés en recherche tabou (Intensification, diversification et aspiration). Les résultats étaient satisfaisants que ce soit sur le plan cryptographique que cryptanalytique, toutefois, il nécessite plus de temps de calcul que certains de nos autres algorithmes (*OEEA* et *AntCrypt*).

Notre deuxième contribution était l'innovation d'algorithmes de chiffrement non déterministes en se bénéficiant de l'aspect aléatoire de la sélection naturelle ou du déplacement des fourmis. En effet, l'aléatoire représente l'ennemi des cryptanalystes. En appliquant toutes les attaques redoutables, il est donc quasiment impossible de parvenir à l'information initiale ou au moins d'établir une relation entre les données chiffrées puisqu'on obtient toujours des versions chiffrées différentes à chaque application de l'algorithme même lors d'un chiffrement sur plusieurs instances d'une même donnée originale.

En fait, à travers notre étude nous sommes arrivés à présenter les métaheuristiques comme des méthodes de résolution approchées se voulant simples et adaptables à tout type de problèmes. Leur capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la solution trouvée. Du point de vue de la recherche opérationnelle, cet inconvénient n'est pas toujours un problème, tout spécialement quand une seule approximation de la solution optimale est recherchée. En nous situant dans ce cas, nous aurions pu mettre en évidence l'avantage de ces nouvelles approches pour résoudre le problème de cryptage de données texte ou images.

Toutefois et malgré que nos algorithmes développés soient de bonne qualité cryptographique, plusieurs autres idées et applications en relation avec ce présent travail pour le compléter, restent à concrétiser. En premier lieu, il est urgent d'aborder le problème de métaréglage [Dréo, 2004], non seulement pour régler automatiquement les paramètres des métaheuristiques, mais aussi pour maîtriser leurs fonctionnements.

De même et malgré que les algorithmes évolutionnaires ou même les algorithmes de colonies de fourmis permettent d'obtenir des solutions pas toujours optimales, mais possiblement satisfaisantes à de nombreux problèmes complexes, leur puissance d'optimisation est liée à la puissance sous-jacente des machines les exécutant.

Comme ce sont des algorithmes inspirés de la nature, ils sont fortement parallélisables et il se trouve que le récent développement des cartes graphiques, dédiées au calcul de rendu 3D dans un premier temps, puis généralisées au milieu des années 2000 avec les *General Purpose Graphic Processing Units (GPGPU)*, permet d'utiliser le grand nombre de cœurs de calcul qu'elles comportent pour autre chose que du rendu graphique.

Dans [Mait, 2011], l'auteur détaille et étudie comment les algorithmes évolutionnaires peuvent bénéficier de ces nouvelles architectures, avec leurs particularités. Plusieurs algorithmes ont été présentés, qui permettent l'utilisation efficace de ces processeurs pour l'évolution artificielle, que ce soit pour les algorithmes génétiques et stratégies d'évolution, mais aussi pour la programmation génétique. Différentes stratégies de parallélisation ont été utilisées, dont l'une nécessitant le développement d'un opérateur parallèle de réduction de la population. L'adaptation des paramètres de ces algorithmes pour permettre le portage de ces derniers sur GPGPU est détaillée, ainsi que certains points impactant directement les performances de ces portages.

Ainsi, de tels schémas peuvent être envisagés pour une parallélisation de nos algorithmes développés surtout PosESecL2 possédant un bon niveau confusionnel mais souffrant d'une lenteur de calcul vu la grande taille de chromosomes manipulés codant des données à chiffrer de grandes tailles.

D’un autre coté et vu que nos algorithmes de cryptage développés sont de type symétrique, une méthode de communication sécurisée de clé de chiffrement secrète peut être envisagée lors de la manipulation de données images. Il s’agit de marquer l'image cryptée dans des régions spécifiques, ou des régions déterminées pour l'utilisateur. Nous pouvons également utiliser des méthodes de marquage (watermarking) sans perte pour insérer la clef cryptée dans l'image cryptée. Il serait souhaitable de développer des évaluations sur la robustesse concernant les attaques. Ainsi, nous souhaitons faire des études concernant la cryptanalyse de nos méthodes pour classifier son niveau de sécurité face aux attaquants.

Par ailleurs, nos perspectives de recherche à long terme s’orientent vers l’étude et le développement d’autres méthodologies de cryptage de données. Nous nous intéresserons, en particulier à :

- La métaheuristique des automates cellulaires inventée par Ulam [Ulam, 1950] et Von Neumann [VonN, 1966] et basée sur le concept de la vie artificielle. Ce concept s’appuie sur des règles extrêmement simples permettant de construire des structures très complexes et esthétiques, d’où sa ressemblance avec le principe d’auto-organisation en biologie animale, notamment avec les mécanismes de la colonie de fourmis. Ces règles peuvent stipuler, par exemple, qu'une naissance nécessite un certain rassemblement de population, que les cellules ne peuvent survivre à un trop grand isolement et qu'une trop forte concentration les étouffe. Donc, nous comptons adapter ces règles au problème de cryptage où de génération en génération, l’application de ces règles permettra l’émergence d’une solution au problème.

- L’approche des « hyper-heuristiques » introduite dans [Cowl, 2000], [Cowl, 2002], [Burk, 2003], [Burk, 2005] et qui consiste à travailler dans un espace de recherche composé de métaheuristiques afin d’optimiser le choix de la métaheuristique à utiliser pour un problème donné. Il s’agit, donc, de mettre en confrontation un ensemble de métaheuristiques différentes pour en sélectionner la plus optimale.

Ministère de l'Enseignement Supérieur et de la recherche Scientifique

BADJI MOKHTAR-ANNABA UNIVERSITY
NIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار – عنابة

Faculté des sciences de l'ingénieur
Département d'informatique

THESE

Présentée en vue d'obtention du diplôme de ***DOCTORAT en Informatique***

Sécurisation évolutionnaire du transfert d'images

Option
Informatique

Par
SOUICI Ismahane

Directeur de Thèse
SERIDI Hamid Pr. Univ. 08 mai 1945 Guelma

Devant le jury

Président :	**KHADIR Tarek**	Pr. Univ. Badji Mokhtar Annaba
Examinateurs :		
	FARAH Nadir	Pr. Univ. Badji Mokhtar Annaba
	MEROUANI Hayet	MC. Univ. Badji Mokhtar Annaba
	KHOLADI Khiredine	Pr. Univ. Mentouri Constantine
	CHIKHI Salim	Pr. Univ. Mentouri Constantine

Année universitaire : 2012/2013

Remerciements

En premier lieu, je tiens à exprimer ma profonde gratitude à Monsieur SERIDI Hamid, Professeur à l'université 08 mai 1945 de Guelma, pour avoir dirigé ma recherche depuis le Magister et pour la confiance et l'intérêt qu'il m'a témoignés tout au long de l'élaboration de cette thèse. Je lui suis très reconnaissante pour le suivi régulier et formateur reçu durant ces quatre années de Doctorat, et bien avant, durant les trois années de Magister. Je le remercie pour la démarche scientifique rigoureuse et l'esprit d'auto-critique qu'il a su m'inculquer. J'espère avoir été à la hauteur de ces espérances.

J'adresse mes remerciements les plus sincères à Monsieur KHADIR Tarek, Professeur à l'université Badji Mokhtar d'Annaba, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Mes vifs remerciements vont aux membres du jury qui ont accepté de prendre de leur temps pour examiner mon travail : Monsieur FARAH Nadir, Professeur à l'université Badji Mokhtar d'Annaba, Madame MEROUANI Hayet, Maître de Conférences à l'université Badji Mokhtar d'Annaba, Monsieur KHOLLADI Khiredine, Professeur à l'université Mentouri de Constantine et Monsieur CHIKHI Salim, Professeur à l'Université Mentouri de Constantine.

Au cours de ces quatre années de recherche, j'ai eu à effectuer un stage de courte de durée en 2010, au laboratoire CReSTIC de l'université de Reims Champagne Ardenne, France, sous la direction de Monsieur AKDAG Herman, Professeur à l'université de Reims Champagne Ardenne, France. Qu'il trouve ici, l'expression de mes remerciements les plus sincères, pour son chaleureux accueil. Je tiens à remercier, aussi, Monsieur Cyril DE RUNZ, Maitre de Conférences à l'université de Reims Champagne Ardenne, France pour sa disponibilité, ainsi que pour ses discussions fructueuses que j'ai eues avec lui.

Merci à celles et ceux qui auront à lire tout ou une partie de ce manuscrit et qui y trouveront un intérêt quelconque.

Enfin, un grand MERCI à ma famille et, en particulier, mes parents pour le soutien et les encouragements qu'ils ont su m'apporter pour arriver au terme de cette thèse.
J'espère les honorer avec ce travail.

الملخص

في يومنا هذا أصبحت شبكات المعلوماتية معقدة و التنصتات غير القانونية عليها قائمة. لهذا أصبح من الضروري حماية المعلومات الحساسة المنقولة عبر هذه الشبكات, أين يعد علم التشفير واحد من الحلول الجد فعالة, الذي و على الرغم من كل التطورات التي سجلت بهذا المجال ماتزال بعض النقائص مسجلة خاصة ما يتعلق بحجم المفتاح المستعمل و مقاومة الهجمات المتطورة.

في هذا السياق ينصب موضوع بحثنا للدكتوراه حيث نسعى إلى تطوير خوارزمات تشفير باستعمال métaheuristiques التي تعد من الطرق التي توفر حلولا تقريبية و تفتح مجالات تصميمية لطرق حل مهمة لمشاكل التحسين.
هذه الأخيرة تنقسم إلى قسمين : metaheuristiques التي تستعمل مجموعة حلول métaheuristiques التي تستعمل حلا واحدا. وبما أن métaheuristiques توظف طابع العشوائية في معظم مراحل عملها و الذي يعد عنصرا مهم إستعماله في مجال التشفير لتعقيد مهمة المهاجمين, لهذا وقع إختيارنا على هذه الطرق لحل مشكل تشفير المعلومات النصية و الصور.

إذن وضمن الصنف الأول الذي يشمل métaheuristiques التي تستعمل مجموعة حلول قمنا بتطوير ثلاث خوارزمات تعتمد الطريقة التطورية المستوحاة من نظرية "دارون" و هذا حسب نوعين مختلفين من التشفير (التشفير المعتمد على الوضعيات و التشفير المعتمد على الظهور), بالإضافة إلى خوارزم رابع و الذي يستعمل métaheuristique مستوحاة من comportement الحقيقي للنمل.
بالنسبة للصنف الثاني والذي يضم métaheuristique التي تستعمل حلا واحدا, توصلنا إلى تكييف طريقة البحث الطابوه إستعمالها في حل المشكل لمدروس و هكذا قمنا بتطويرخوارزم تشفير طابوه خامس.

أخيرا قمنا بتجريب ومقارنة كل الخوارزمات المقترحة فيما بينها وبالنسبة لعدد من مراجع التشفير حيث أظهروا خصائص جيدة سواءا فيما يتعلق بدرجة الغموض أو مقاومة الهجمات الأكثر تطورا. قطة أخرى جد مهمة توصلنا إليها عبر عملنا هذا هي استحداث خوارزمات non déterministes , نقطة تساهم بدرجة كبيرة في زيادة قوة مثل هذه الخوارزمات.

Résumé

Aujourd'hui, les réseaux informatiques sont complexes et les écoutes illégales possibles. Il se pose donc un réel problème quant à la sécurité lors de la transmission de données. Pour des raisons éthiques, le transfert des données délicates ne peut se faire avec un tel risque et doit donc se protéger. La protection la plus adaptée pour ce type de communication réside dans la cryptographie. Cependant et malgré toutes ses évolutions et ses mises en œuvre, elle est toujours entravée par quelques défauts citant en particulier la taille des clés et la résistance contre les attaques avancées.

Ainsi, notre travail de thèse porte sur le développement de nouveaux algorithmes cryptographiques par exploitation des métaheuristiques constituant une partie importante des méthodes approchées et ouvrant des voies très intéressante en matière de conception de méthodes heuristiques pour l'optimisation. Ces dernières se scindent en deux catégories : les métaheuristiques à population de solutions et les métaheuristiques à une seule solution ou encore dites à trajectoire. Et vu que les métaheuristiques exploitent un aspect pseudo-parallèle durant différentes étapes de leurs schémas opératoires, chose qui est très intéressante à exploiter dans le domaine cryptographique pour compliquer de façon considérable la tâche des cryptanalystes, donc notre choix s'est porté sur l'utilisation de telles méthodes pour résoudre le problème de cryptage de données texte et images.

En effet, et pour la première catégorie de métaheuristiques à population, nous avons développé trois algorithmes de chiffrement exploitant les principes évolutionnaires inspirés de la théorie darwinienne, mais opérants suivant deux modes de chiffrement différents (chiffrement à base de positions et un chiffrement à base d'occurrences) ; et un quatrième algorithme utilisant une métaheuristique inspirée du comportement réel des fourmis (ACO).

Pour la deuxième catégorie qui est celle de métaheuristiques à trajectoire, nous avant arrivé à adapter la méthode de recherche taboue et l'exploiter pour résoudre notre problème étudié et ainsi développer un cinquième algorithme de cryptage tabou.

Les algorithmes proposés ont été évalués et comparés entre eux et avec les principaux algorithmes standards de cryptage où ils ont montré de bonnes caractéristiques relatives soit à leur degré confusionnel ou à leur résistibilité aux attaques les plus avancées. L'autre point crucial résultant de nos travaux de thèse était l'innovation d'algorithmes de chiffrement non déterministes, chose qui augmente considérablement leur robustesse.

Mots clés : Cryptage, attaques avancées, métaheuristiques, algorithmes évolutionnaires, recherche tabou, algorithme de colonies de fourmis.

Abstract

Today, computer networks are complex and the illegal wiretapping is possible. This raises a real problem of security when transmitting data. For ethical reasons, the transfer of delicate data can not be done with such a risk and must protect themselves. Protection best suited for this type of communication lies in cryptography. However, despite all its developments and implementations, it is still hampered by some shortcomings citing in particular the key size and resistance against advanced attacks.

Thus, the object of our thesis is the development of new cryptographic algorithms by exploitation of metaheuristics constituting a significant portion of the approximate methods and opening very interesting channels in the design of heuristic methods for optimization. The latter fall into two categories: metaheuristics of solutions population and metaheuristics of one solution or also tell trajectory metaheuristics. And since metaheuristics exploit a pseudo-parallel aspect during different stages of their operating patterns, something that is very interesting to use in the cryptographic domain to significantly complicate the task of the cryptanalysts, so our choice fell on the use of such methods to solve the problem of the encryption of text and images data.

Indeed, for the first class of metaheuristics of population, we developed three encryption algorithms exploiting the evolutionary principles inspired by the Darwinian theory but operating in two different encryption modes (positions based encryption and occurrences based encryption) and a fourth algorithm using a metaheuristic inspired by the ants real behavior (ACO).

For the second category which is trajectory metaheuristics, we arrived to adapt the method of tabu search and use it to solve the problem at hand. Thus, a fifth taboo encryption algorithm was developed.

The proposed algorithms were evaluated and compared among themselves and with principal encryption standard algorithms where they showed good characteristics for either their confusional degree or their resistibility against the most advanced attacks. The other crucial point arising from our thesis work was the innovation of non-deterministic encryption algorithms, something that greatly increases their robustness.

Table des matières

Chapitre III : CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

Chapitre IV : CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

Chapitre V : CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Dès que les hommes ont appris à communiquer, ils ont trouvé des moyens d'assurer la confidentialité d'une partie de leurs communications : l'origine de la cryptographie remonte sans doute aux origines de l'homme. En effet, le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire de les rendre inintelligibles sans une action spécifique.

Du bâton nommé « scytale » au Vie siècle avant JC, en passant par le carré de Polybe ou encore le code de César, on assista au développement plus ou moins ingénieux de techniques de chiffrement expérimentales dont la sécurité reposait essentiellement dans la confiance que leur accordaient leurs utilisateurs. Après la Première Guerre mondiale a lieu une première révolution technologique.

Mais ce n'est qu'à l'avènement de l'informatique et d'Internet que la cryptographie prend tout son sens. Les efforts conjoints d'IBM et de la NSA conduisent à l'élaboration du DES (Data Encryption Standard), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XXe siècle. À l'ère d'Internet, le nombre d'applications civiles de chiffrement (banques, télécommunications, cartes bleues…) explose. Le besoin d'apporter une sécurité accrue dans les transactions électroniques fait naître les notions de signature et authentification électroniques. La première technique de chiffrement à clef publique sûre (intimement liée à ces notions) apparaît : le RSA.

Donc, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information. C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à ramener et à modéliser le problème de cryptage comme un problème d'optimisation.

En effet, l'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solutions algorithmiques efficaces valables pour toutes les données.

Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles que les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens.

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques*. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, les chromosomes et les mécanismes d'intensification et de diversification pour la métaheuristique des algorithmes évolutionnaires), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Les métaheuristiques sont représentées essentiellement par les *méthodes de voisinage* comme le recuit simulé et la recherche tabou, et les *algorithmes évolutifs* comme les algorithmes génétiques et les stratégies d'évolution. Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi et hormis cette introduction et la conclusion générale qui reprennent les travaux de l'ensemble des chapitres et quelques perspectives majeures pour la poursuite de ce travail, le manuscrit est divisé en deux grandes parties.

Un état de l'art aussi complet que possible relatif soit au problème étudié qui est celui de cryptage, soit aux approches de résolutions exploitées qui sont les métaheuristiques, fera l'objet des deux premiers chapitres formant la première partie. En effet, le premier chapitre traite la cryptographie depuis sa première apparition jusqu'à nos jours en présentant un bref aperçu historique permettant de comprendre la distinction entre les trois disciplines (cryptologie, cryptographie et cryptanalyse), les fonctionnalités de base de la cryptographie (confidentialité, authentification, intégrité et la non-répudiation) et leurs différentes catégories (la cryptographie symétrique, asymétrique, hybride et quantique) tout en citant les fameux algorithmes appartenant aux principales catégories.
De son tour, le deuxième chapitre présente une introduction aux métaheuristiques tout en citant les concepts de base ainsi qu'une classification des métaheuristiques illustrée d'exemples d'algorithmes de chacune des classes.

Dans la deuxième partie de notre recherche, nous proposons un nouvel axe de recherche que représente l'application des métaheuristiques pour résoudre le problème de cryptage de données textes et images. Ainsi, les deux catégories de métaheuristiques ont été exploitées : métaheuristiques à population et les métaheuristiques à trajectoire.
Dans le cadre de la première, nous avons choisi d'exploiter deux métaheuristiques qui sont les algorithmes évolutionnaires (AEs) et les algorithmes de colonies de fourmis. Le principal avantage de ces méthodes heuristiques vient de leur capacité à traiter le problème de cryptage en ne possédant qu'un minimum d'informations sur celui-ci. Chose qui est primordiale dans ce problème pour augmenter la confusion des algorithmes développés.

Ainsi, le troisième chapitre résume l'application d'AEs où les différentes étapes partant du codage sont explicitées. En effet, pour un problème à résoudre, il est nécessaire d'adapter la représentation des individus aux objectifs recherchés. Cette adaptation permet de faire converger l'AE plus ou moins rapidement et de tenir compte naturellement des contraintes de la modélisation du problème. Dans le cas présent, l'espace de recherche que nous cherchons à optimiser est l'ensemble représenté par les solutions possibles du problème de cryptage et par une fonction d'évaluation de chaque solution.

Le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, les algorithmes ont été groupés en deux catégories distinctes suivant les modes de chiffrement utilisé implémentant différents codages : chiffrement à base de position ou chiffrement à base d'occurrences.

D'une manière globale, nous allons définir un individu de la population comme une donnée chiffrée possible. Cet individu doit pouvoir être évalué numériquement par la fonction d'évaluation proposée. Cette fonction de fitness calcule la qualité d'une donnée chiffrée en fonction du besoin de confusion. Nous montrerons aussi l'application de notre méthode sur quelques exemples d'images de différentes tailles. Une interprétation et une discussion des résultats obtenus seront abordées pour pouvoir, par la suite, comparer les nouveaux algorithmes proposés de cryptage évolutionnaire où l'algorithme opérant suivant le mode de chiffrement par occurrences a montré une efficacité en termes de temps de calcul, de pouvoir de confusion, de résistibilité aux attaques et de longueur de clé meilleure que celles des algorithmes opérants suivant le mode de chiffrement par positions.

Cette conclusion a été démontrée par l'application d'une deuxième méthode de résolution exploitant les algorithmes de colonies de fourmis illustrée sur un quatrième chapitre où les résultats obtenus été très satisfaisants.

Au niveau du cinquième chapitre, un autre algorithme de cryptage a été proposé s'inscrivant cette fois-ci, sous la deuxième catégorie de métaheuristiques dans le but d'explorer l'espace de recherche par exploitation de la notion de voisinage. Il s'agit d'un algorithme utilisant le principe d'une recherche tabou pour pouvoir choisir la configuration représentant la meilleure solution au problème de cryptage de données texte ou images.

Dans chacun des chapitres décrivant nos algorithmes de cryptage proposés, nous présentons des exemples et résultats expérimentaux sur des données tests. Nous avons aussi effectué des bilans en faisant ressortir les avantages et faiblesses des méthodes développées.

CHAPITRE I

CRYPTOGRAPHIE

I.1. Introduction

Depuis les temps historiques les plus reculés, l'homme a perçu le besoin de cacher, de dissimuler ou faire mystère des informations personnelles ou confidentielles, et cela bien avant l'ère informatique afin de les rendre inintelligibles à des lecteurs indésirables. C'est pourquoi, les codes ont existé.

Les origines de la cryptographie semblent remonter à plus de 4000 ans en Egypte. Plusieurs indications archéologiques tendent à montrer que les « écritures secrètes » sont en fait anciennes que l'invention de l'écriture elle-même. Polybius développa un système de codage des lettres de l'alphabet consistant à remplacer chaque lettre de l'alphabet par deux nombres, donnant la ligne et la colonne où se trouve cette lettre dans une matrice. Jules César utilisait une simple méthode de substitution de lettres pour communiquer secrètement avec ses généraux : c'est un chiffre par décalage,…etc.

C'est cependant au cours de la seconde guerre mondiale que la cryptographie s'inscrit véritablement comme élément central des stratégies militaires. Le cas le plus connu est certainement l'histoire entourant le décodage du code Enigma par les Polonais et les Britanniques. Une conjonction d'espionnage classique et d'efforts de mathématiciens polonais permet de déduire la clé utilisée et ainsi de décoder les messages encodés avec Enigma. On trouve apparemment bien moins de détails sur les efforts cryptographiques durant la guerre froide, probablement parce que ces informations sont encore « Top Secret ».

On en est maintenant à l'époque moderne où le champ d'application de la cryptologie s'est élargi et a trouvé un regain d'actualité avec toutes les applications nouvelles suscitées par l'utilisation de l'Internet. La révolution d'Internet et l'utilisation de plus en plus d'informations massives sous forme numérique facilitent les communications et rendent de ce fait plus fragiles les informations que l'on détient, c'est pourquoi il devient nécessaire de protéger le contenu de certains messages des inévitables curieux. En effet, les réseaux ouverts créent des brèches de sécurité et il est plus aisé à un adversaire d'accéder aux informations. Dans une communication à distance, des questions cruciales se posent et leurs réponses s'imposent ; comment être sur :

- que l'on parle à la bonne personne (authenticité),
- que nos propos ne sont pas altérés (intégrité),
- que la conversation n'est pas espionnée (confidentialité),
- de l'identité de l'émetteur (non répudiation) ?

Indéniablement, avec l'essor fulgurant des nouvelles technologies, le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements.

Alors que la cryptographie consiste à sécuriser les données, tandis que, la cryptanalyse est l'étude des informations cryptées afin d'en découvrir le secret. Grâce à la cryptanalyse, les militaires ont pu mener leurs guerres en découvrant les correspondances de leurs ennemis et en contrôlant les réseaux de communications mais d'autre part si elle est utilisée par une personne mal intentionnée il peut générer des dégâts onéreux aux entreprises ou aux sociétés.

I.2. Les Fondements de la Cryptographie

I.2.1. Terminologie

Comme toute science, la cryptographie possède son propre langage. Dans ce qui suit les mots clés de domaine cryptographique.

I.2.1.1. Cryptographie

Le terme cryptographie vient en effet des deux mots grecs : *Kruptus* qu'on peut traduire comme secret et *Graphein* pour écriture. Ainsi la cryptographie est l'art de dissimuler une information écrite en clair (plain text) en *cryptogramme* (cipher text) pour qu'elle soit incompréhensible que par son destinataire légitime par le biais d'une clé appelé « clé de chiffrement » (processus de *chiffrement*). Pour rendre l'information à nouveau intelligible par le biais d'une clé appelé « clé de déchiffrement » le processus inverse est appliqué (processus de *déchiffrement*).

On distingue généralement deux types de clefs :

- **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

Le schéma suivant illustre le processus cryptographique :

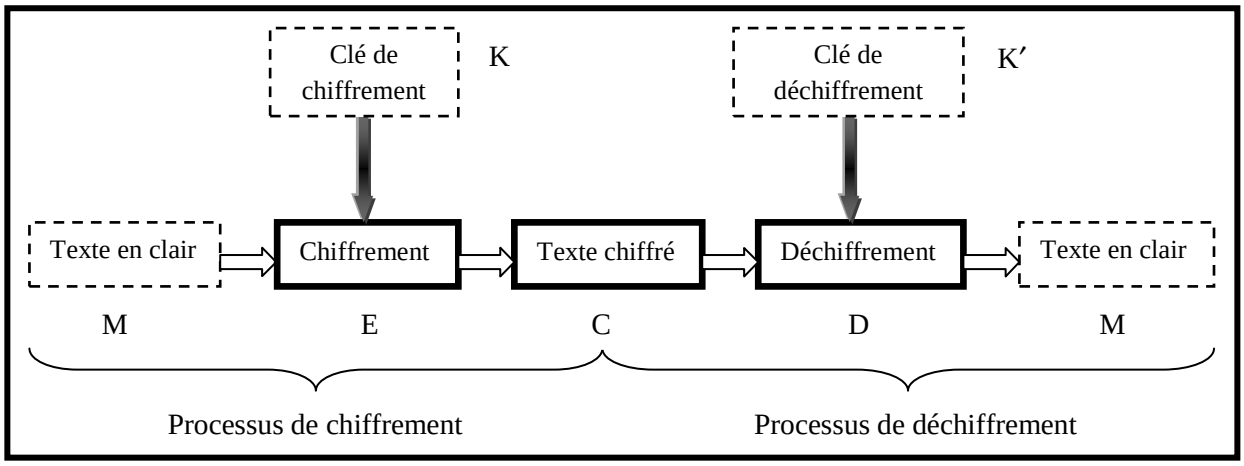


Figure I.1. *Processus cryptographique.*

I.2.1.2. Cryptanalyse

a. Définition

C'est l'art d'étude des crypto systèmes en cherchant leurs failles et leurs vulnérabilités afin de retrouver des messages clairs correspondant à des messages chiffrés sans avoir à connaître les clés utilisées dans le chiffrement. Lorsque tous les éléments de la méthode utilisée pour coder des messages sont repérés, on dit qu'on a **cassé** ou **brisé** le système cryptographique utilisé. Plus un système est difficile à briser, plus il est sûr.
La personne qui pratique la cryptanalyse est appelée : cryptanalyste. Il tente à décrypter le message chiffré pour découvrir son secret. On a distingué entre le verbe « décrypter » et « déchiffrer » puisque ce dernier est réservé pour le déchiffrement par le destinataire légitime.

b. Types de cryptanalyse

On distingue plusieurs types d'attaques :

- ***Attaque sur texte chiffré seul (ciphertext-only)*** : l'attaquant a seulement la possibilité d'intercepter un ou plusieurs messages chiffrés. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- ***Attaque à texte clair connu (known-plaintext attack*)** : se base sur la connaissance d'une partie du texte en clair pour déduire le reste du message. La tâche est de retrouver la clef utilisée pour chiffrer ce message.
- ***Attaque à texte clair choisi (chosen-plaintext attack)*** : se base sur la possibilité de choisir un texte clair et d'obtenir son chiffrement et en ayant la possibilité de générer les versions chiffrées de messages clair avec un algorithme considérer comme une **boîte noire** tel que les algorithmes à clé publique puisque l'algorithme est public.
- ***Attaque à texte chiffré choisi (chosen-ciphertext attack)*** : le cryptanalyste possède des messages chiffrés et essaye de les déchiffrer de son choix. Sa tâche est de retrouver la clef.

c. Familles d'attaques cryptanalytiques

Il existe plusieurs familles d'attaques cryptanalytiques, les plus connues sont les suivantes :

- **L'analyse fréquentielle** : examine les répétitions des lettres du message chiffré afin de trouver la clé. Cette technique est découverte par Al-Kindi au IXe siècle [www1] contre les chiffrements mono-alphabétiques. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est basée sur le fait que, dans chaque langue, certaines lettres ou combinaisons de lettres apparaissent avec une certaine fréquence.
- **L'attaque par dictionnaire** : le mot testé est pris dans une liste prédéfinis contenant les mots de passe les plus courants et aussi des variantes de ceux-ci. Ces listes sont généralement dans toutes les langues les plus utilisées, elles contiennent des mots existants ou des mots diminutifs (par exemple « powa » pour « power » ou « K7 » pour « cassette »). Elle est souvent couplée à l'attaque par force brute.
- **L'attaque par force brute** : s'appuie sur le cassage d'un mot de passe en testant tous les mots de passe possibles. C'est le seul moyen de récupérer la clé dans les algorithmes les plus modernes et encore inviolés comme AES.
- **La cryptanalyse linéaire** : c'est une attaque à texte clair inventée par le japonais Mitsuru Matsui [www1]. L'idée est de trouver des approximations linéaires entre les bits de sortie, les bits d'entrée et les bits de la clé. Si certaines de ces approximations apparaissent avec une probabilité suffisante, on a alors démontré que la correspondance

entre entrée et sortie n'est pas purement aléatoire. Le nombre de clés à envisager pour déchiffrer le message est restreint.

- **La cryptanalyse différentielle** : découverte par deux cryptologies israéliennes : Bihan et Shamir [www1]. Elle consiste à comparer les sorties de l'algorithme quand on lui met en entrée deux messages ayant une différence fixe. On étudie comme variant les sorties si les deux messages ne diffèrent que par un seul bit. Si en déplaçant ce bit à l'intérieur des messages, certains bits des sorties restent inchangés, on a alors trouvé une faille dans l'algorithme et celui-ci est attaquable.

I.2.1.3. Cryptologie

La cryptologie du grec kruptos « secret, caché » et logos « discours » qui embrasse à la fois la cryptographie et la cryptanalyse. Elle se partage entre la cryptographie, qui inclut la conception des mécanismes destinés à assurer les fonctions suivantes : confidentialité, intégrité, authentification et traçabilité (non répudiation), et la cryptanalyse dont le but est de déjouer les protections ainsi mises en place.

I.2.1.4. Fonction de hachage

Une fonction de hachage est une fonction mathématique qui à partir d'un message (d'une donnée) génère une autre chaîne, l'empreinte (généralement plus courte). Cette empreinte est très sensible au texte initial (une petite modification du texte provoque une grande modification d'empreinte).

I.2.1.5. Signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message. Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur.

I.2.1.6. Les certificats

Pour assurer l'intégrité des clés publiques, ces dernières sont publiées avec un certificat. Un certificat (ou certificat de clés publiques) est une structure de données qui est uniquement signée par une autorité certifiée.

CA, Certification Autorité, est une autorité en qui les utilisateurs peuvent faire confiance. Il contient une série de valeurs, comme le nom du certificat et son utilisation, des informations identifiants le propriétaire et la clé publique, la clé publique elle-même, la date d'expiration et le nom de l'organisme de certificats. Le CA utilise sa clé privée pour signer le certificat et assure ainsi une sécurité supplémentaire. Si le récepteur connaît la clé publique du CA, il peut vérifier que le certificat provient vraiment de l'autorité concernée et, ainsi, de s'assurer que le certificat contient des informations viables et une clé publique valide.

I.2.2. Fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur fiabilité et confidentialité au travers d'un canal peu sûr (téléphone, réseau informatique ou autre). Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions en plus de la fiabilité et la confidentialité. Il s'agit de garantir l'intégrité et l'authenticité des données échangées. Les postulats de sécurité associés à la cryptographie sont :

a. Intégrité

Vérifier l'intégrité des données consiste à déterminer si les données, ressources, traitements ou services n'ont pas été altérées durant la communication de manière fortuite ou intentionnelle.

b. Confidentialité

La confidentialité est le maintien du secret des informations. Elle consiste à rendre l'information discrète ou inintelligible à d'autres personnes que les seuls acteurs de la transaction. La confidentialité peut être vue comme la «protection des données contre une divulgation non autorisée» [Gher, 2004].

c. Authentification

L'authentification consiste à assurer l'identité d'un utilisateur c.à.d. de garantir à chacun de correspondant que son partenaire est bien celui qu'il croit être.
On distingue deux types d'authentification :

- ***Le contrôle d'accès :*** C'est l'opération permettant d'être certain de l'identité d'une personne utilisateur pour permettre l'accès à des ressources uniquement pour la personne autorisée (par exemple l'utilisation d'un mot de passe pour un disque dur).
- ***Authentification de l'origine des données :*** Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services, l'authentification et l'intégrité.

d. La non répudiation (traçabilité)

La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction ; l'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information.

I.2.3. Problèmes de la cryptographie

Quelque soit le cryptosystème utilisé pour le chiffrement, ceci reste toujours cassable un jour ou l'autre. La sécurité d'un cryptosystème repose en fait sur la complexité des algorithmes définis et sur les puissances de calcul disponibles pour une attaque. Ainsi la solution adoptée actuellement est de faire « *retarder le travail des cryptanalystes* », c.à.d. faire en sorte que la durée nécessaire pour déchiffrer un code soit supérieure à la durée de validité des données.

I.3. Algorithmes cryptographiques

Au cours d'un échange visant à communiquer de façon secrète deux protagonistes, appelé ici l'émetteur et le récepteur à travers un canal peu sûr, l'information à transmettre sera donc chiffrée par un procédé de chiffrement en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de en utilisant une clé de déchiffrement sans laquelle son procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique : *algorithme cryptographique* (ou *chiffre***)**. La sécurité des données chiffrées est entièrement dépendante de deux facteurs : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un *cryptosystème***.**

I.3.1. Le principe de Kerckhoffs

Pour briser un cryptosystème, un opposant cherche à obtenir deux éléments d'information :

1. Quel est le type de système de codage utilisé ? et,
2. Quelle est la clé d'encodage utilisée ?

Bien entendu, son travail est simplifié (mais certainement pas terminé) s'il connait le type de système utilisé. Avec le temps cette information finit par circuler. Cette hypothèse de travail est appelée le principe de Kerckhoffs. Ce principe consiste à affirmer que la sécurité d'un système de chiffrement ne devrait pas être fondée sur le secret de la procédure utilisée, mais essentiellement sur le secret de la clé.

Auguste Kerckhoffs écrit en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire », où il disait [Kerc, 1883] :

« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants:

1) le système doit être matériellement, sinon mathématiquement, indéchiffrable.
2) il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.
3) la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
4) il faut qu'il soit applicable à la correspondance télégraphique.
5) il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
6) enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer ».

Les points 2 et 3 sont les axiomes fondamentaux de la cryptographie suivant lesquels l'attaquant possède tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé spécifique pour le chiffrement. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ce secret sera découvert ou volé.

I.3.2. Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplait (P, C, K, E, D) satisfaisant les points suivants :

1) P est un ensemble fini de blocs de textes clairs possibles.
2) C est un ensemble fini de blocs de textes chiffrés possibles.
3) K est un ensemble fini de clefs possibles.
4) Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k (e_k (x)) = x$ pour tout texte clair $x \in P$.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$ qui doit être transmise préalablement loin des yeux d'Oscar (à travers un canal de communication sûr). Supposant qu'Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne : $x = x_1 x_2 \ldots x_n$ avec : $n \in Z$, $n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera : $y = y_1 y_2 \ldots y_n$.

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1 x_2 \ldots x_n$. Le procédé de communication est illustré sur la Figure I.2.

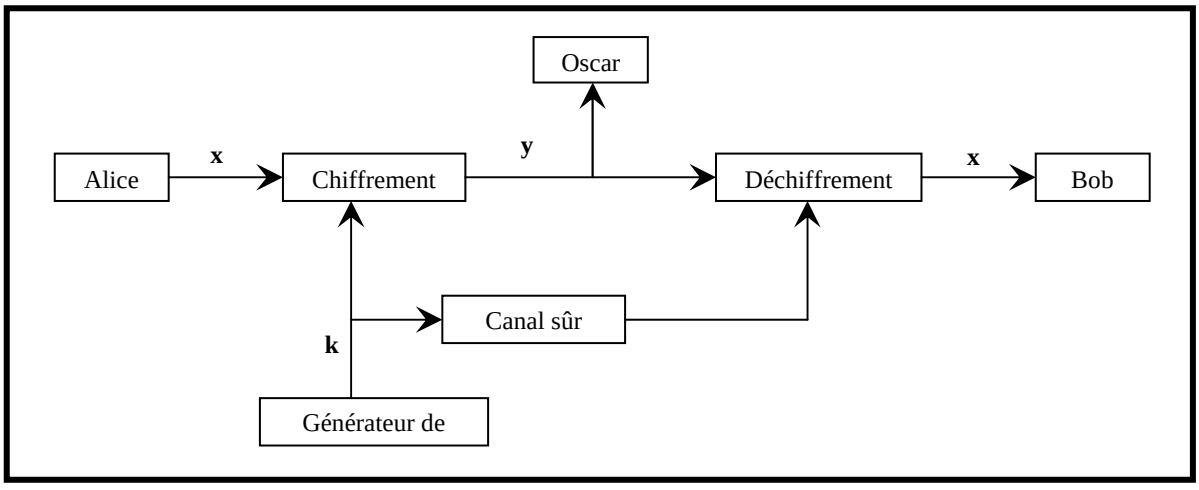


Figure I.2. *Le procédé de communication.*

I.3.3. Classes de cryptographie

Le schéma suivant illustre les différentes classes de la cryptographie :

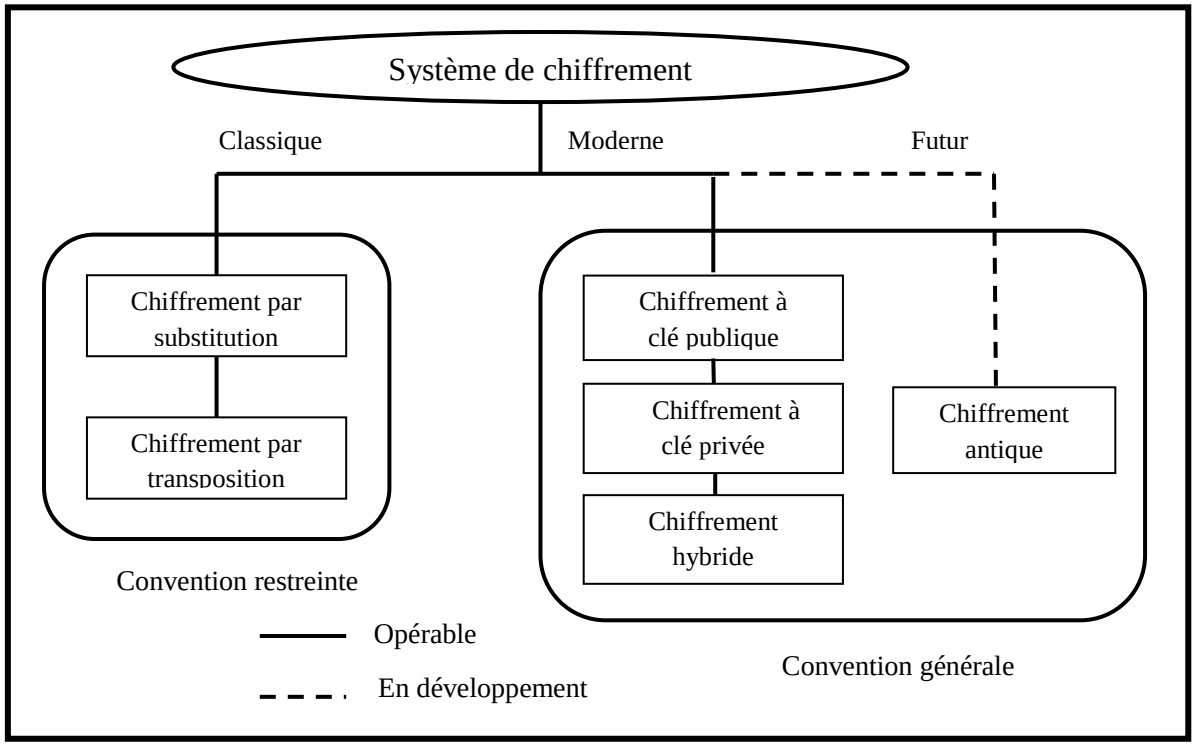


Figure I.3. *Les classes de la cryptographie*

I.3.3.1. La cryptographie classique

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble et ils sont trop simples pour offrir la moindre sécurité. Pour cacher la substance d'un texte, ils utilisent la substitution de caractères par d'autres ou les transposer dans des ordres différents. De ce fait, la confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide. On appelle généralement cette classe de méthodes : le chiffrement **à usage restreint**.

a. Cryptographie par substitution

La substitution signifie que chaque lettre (ou groupe de lettres) est substituée par une (ou groupe) lettre(s), chiffre(s) ou symbole(s). Le déchiffrement consiste à effectuer la substitution inverse. Selon la façon de substituer, on quatre catégories :

- **Substitution simple (mono-alphabétique)**

Le codage par substitution mono-alphabétique (ou encore les alphabets désordonnés) est le plus simple à imaginer. Chaque lettre dans le message clair est remplacée dans le message chiffré par une autre lettre différente unique pour toutes les occurrences de celle-ci.

Dans la littérature, plusieurs algorithmes ont été proposés, entre autres, nous citons : le chiffre de César, le chiffre Atbash, le carré de Polybe, etc.

- **Substitution poly-alphabétique**

Au lieu de remplacer une lettre par une même autre lettre dans tout le message comme dans la substitution simple, elle est remplacée périodiquement par différentes lettres. L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles.

- **Substitutions homophoniques**

Au lieu d'associer un seul caractère crypté à un caractère en clair, on dispose d'un ensemble de possibilités de substitution de caractères dans lequel on choisit aléatoirement. Par exemple : C=>S, K ; G=>G, J ; Q=>K ; S=>S, Z ; PH=>F ; …etc.

- **Substitution par polygrammes**

Au lieu de substituer des caractères, on substitue par exemple des digrammes : groupe de deux caractères. Pour se faire, deux moyen sont utilisés : soit par table (Chiffre de Playfair) ou par transformation mathématique (Chiffre de Hill).

b. Cryptographie par transposition

Elle consiste à permuter les lettres du message à chiffrer entre elles, afin de le rendre inintelligible. Plusieurs variations de transposition sont utilisées, parmi eux on trouve :

- **Transposition simple (à base matricielle)**

Elle consiste à écrire le texte en clair dans une matrice de ***n*** colonnes (une lettre dans chaque case), et ensuite de construire le texte chiffré en prenant les lettres à partir de cette matrice colonne par colonne. La clé dans ce cas est le nombre ***n***.

- **Transposition avec substitution simple**

L'idée dans ce cas est de combiner la transposition avec une substitution simple. Il s'agit ainsi de chiffrer le message clair par une méthode de substitution simple, et en suite d'en appliquer une transposition. Une autre astuce est souvent utilisée qui consiste à appliquer une fonction de permutation sur l'ordre d'arrangement des colonnes. On cite à titre d'exemple : le chiffre de DELASTELLE.

c. Conclusion

Les nouvelles techniques de communications (moyens de transports rapides, journaux, télégraphe, télégraphie sans fil) donne une nouvelle impulsion à la cryptologie. L'interception devient simple et le décryptement des informations devient vital. La cryptologie entre dans son ère moderne et les algorithmes classiques ne sont plus efficaces.

I.3.3.2. La cryptographie moderne

La cryptographie entre dans son ère moderne avec l'utilisation intensive des ordinateurs à partir des années septante. Vue la nécessité croissante de sécuriser les données dans tous les domaines (économique, industriel, informatique,...etc.) …. La cryptographie est appelée à devenir une technique de plus en plus fondamentale pour la protection des informations possédées et/ou échangées par des individus ou des organisations. Dans la cryptographie moderne, on utilise aussi des problèmes mathématiques que l'on ne sait pas (encore) résoudre, par exemple factoriser des grands nombres (chiffre RSA).

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ la *cryptographie à clef secrète,* ou encore appelée *symétrique,*
- ✓ la *cryptographie à clef publique,* dite également *asymétrique*.

a. La cryptographie symétrique

- **Principe**

Le cryptage à clé privée ou symétrique (ou encore dit conventionnel) utilise la même clé pour chiffrer et déchiffrer un message. Autrement dit, les clefs de chiffrement et de déchiffrement sont identiques ou on peut facilement calculer l'une à partir de l'autre. La connaissance de cette clef est cruciale pour la confidentialité des informations échangées. L'emploi d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Ces systèmes sont vulnérables parce qu'il est généralement possible de découvrir la clé à partir des messages codés. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1}. En excepte le système à masque jetable qui est hors porté de cette attaque.

Les algorithmes symétriques sont de deux types :

- ✓ Les algorithmes de *chiffrement en continu*, qui agissent sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé *chiffrement en flux* (Stream cipher).
- ✓ Les algorithmes de *chiffrement par blocs* (Bloc cipher), qui opèrent sur le texte en clair par groupes de bits appelés blocs.

Plusieurs algorithmes de chiffrement symétriques ont été définis. Nous présentons, ci-dessous, les plus connues de ces méthodes.

- **Quelques algorithmes de chiffrement symétrique**

A ce niveau, on va présenter, plus ou moins en détails, les plus fameux des algorithmes de chiffrement s'inscrivant sous ce mode.

1) Masque jetable (one-time pad)

Ce chiffre appelé aussi chiffre de Vernam ou encore le chiffrement parfait; est un algorithme de cryptographie inventé par Gilbert Vernam en 1917. Ce chiffrement est le seul qui soit théoriquement impossible à casser puisque la sécurité de ce système repose sur la génération complètement aléatoire de la clé [www2]. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront équiprobables. Malgré cela, il présente d'importantes difficultés de mise en œuvre pratique, il ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

Il consiste à combiner le message en clair avec une clé présentant les caractéristiques suivantes :

- choisir une clé K_M aussi longue que le texte à chiffrer.
- utiliser une clé constituée de caractères choisis aléatoirement.
- ne jamais utiliser 2 fois la même clé (d'où le nom de masque jetable).
- pour chiffrer un message faire le ou exclusif du message et de la clé : $C=M \oplus K_M$.
- pour déchiffrer un message l'opération est la même : $M= C \oplus K_M = M \oplus K_M \oplus K_M$.

2) DES (Data Encryption Standard)

Jusqu'aux années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence de besoins civils, le NBS (National Bureau of Standards) lança le 15 mai 1973 un appel d'offres dans le Federal Register (l'équivalent du Journal Officiel américain) [Stin, 1996] pour la création d'un système cryptographique qui doit :

- Reposer sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement.
- Etre facile à implémenter, logiciellment et matériellement et être très rapide aussi.
- Avoir un haut niveau de sécurité, lié uniquement à la clé et non à sa confidentialité.

Ce cryptosystème permet de chiffrer des messages de 64 bits avec une clef de 56 bits. De ce fait, c'est un système de chiffrement par blocs. Pour chiffrer un texte, il faut d'abord le découper en blocs de 64 bits puis appliquer le chiffrement sur chacun des blocs. Ainsi, les données en entrée de cet algorithme (texte clair et les données en sortie (texte chiffré) seront des bocs de 64 bits.
Sa clé est une chaîne binaire de 64 bits, mais en fait seuls 56 bits servent réellement à définir la clé. Les bits 8, 16, 24, 32, 40, 48, 56, 64 sont des bits de parité. Le $8^{ème}$ bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Ceci permet d'éviter les erreurs de transmission. Il y a donc pour DES 2^{56} clés possibles, soit environ 72 milliards possibilités. Malgré ça, c'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, alors qu'elle a été largement suffisante au moment de sa conception.

Nous donnons ci-dessous une idée simple et intuitive du fonctionnement du DES qui est un algorithme relativement simple puisqu'il combine des permutations et des substitutions. Il se déroule en quatre étapes [www3] :

1. **Préparation-Diversification de la clé :** le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K 16 sous-clésK_1,…, K_{16} à 48 bits.
   La figure ci-dessous montre comment obtenir à partir d'une clé de 64 bits 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES.

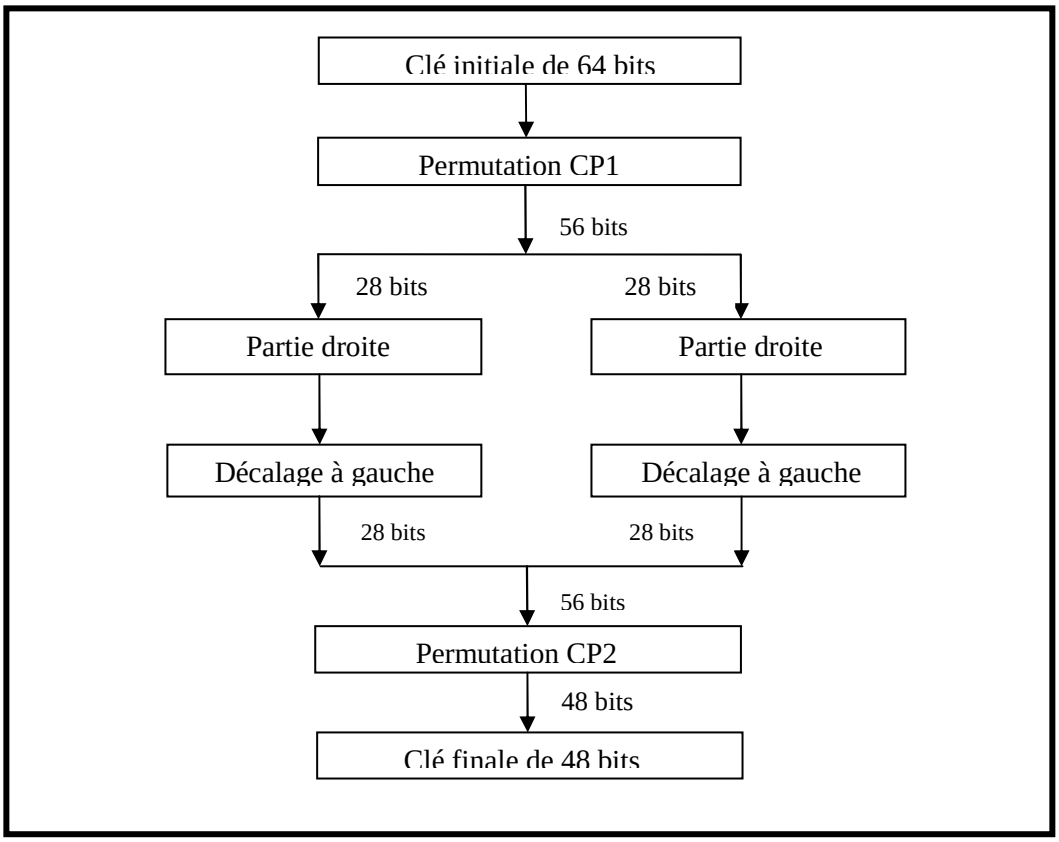


Figure I.4. *Génération des clés.*

La première étape consiste à faire une permutation noté CP1 dont la matrice est présentée ci-dessous, puis éliminer les bits de parité afin d'obtenir une clé d'une longueur de 56 bits. Elle est composée de deux matrice G_i et D_i chacune de 28 bits. Ces deux blocs subissent ensuite une rotation à gauche. Et enfin, ces derniers sont regroupé en un bloc de 56 bits ensuite une permutation CP2 a eu place pour fournir en sortie une clé de 48 bits. Des itérations de l'algorithme permettent de donner les 16 clés K_1, …, K_{16}.

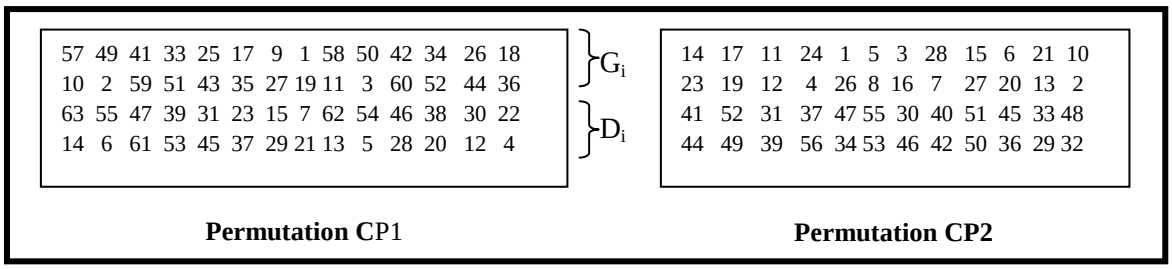


Figure I.5. *Permutation CP1 et CP2.*

2. **Permutation initiale :** pour chaque bloc de 64 bits x du texte, on calcule une permutation finie y=PI(x). y est représenté sous la forme : y = G_0 D_0 , G_{16} étant les 32 bits à gauche de y, D_0 les 32 bits à droite (voir Figure I.6). Quant à leurs significations, par exemple, la permutation initiale (IP) signifie que le $58^{ème}$ bit de la chaine à chiffrer, *x*, est le premier bit de IP(*x*),…

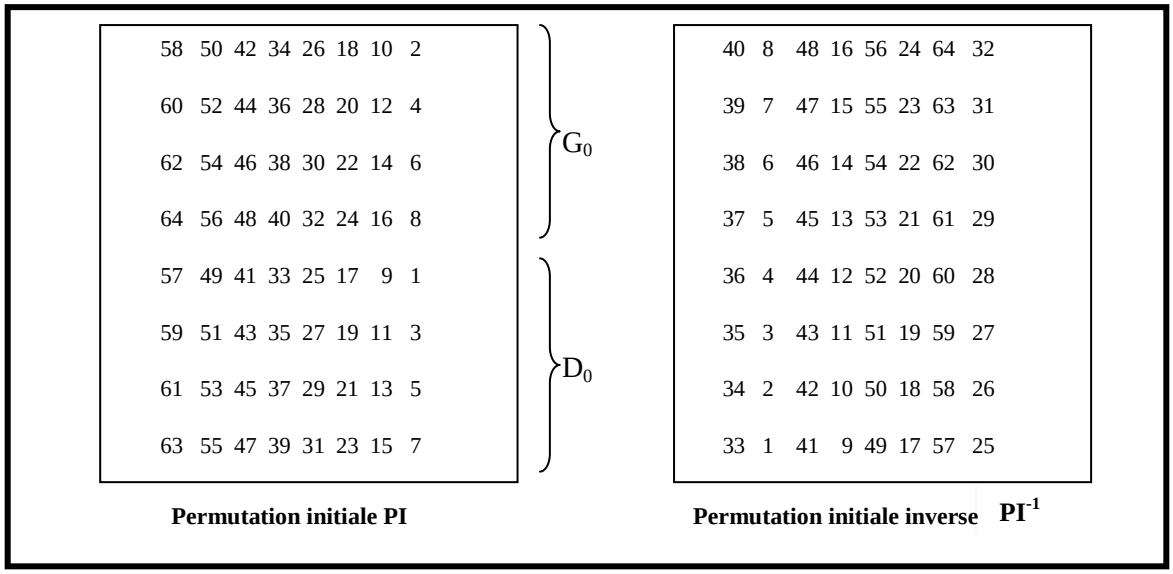


Figure I.6. *La permutation initiale et son inverse.*

3. **Itérations (Rondes) :** On applique 16 rondes d'une même fonction. A partir de $G_{i-1}D_{i-1}$ (pour i de 1 à 16), on calcule G_iD_i en posant :
   - $G_i = D_{i-1}$
   - $D_i = G_{i-1} \oplus f(D_{i-1},K_i)$

D'après la formule qui correspond au calcul de D_i, on constate que la fonction *f* utilise deux arguments ayant des tailles différentes : D_{i-1} de 32 bits et k_i de 48 bits. Ainsi, D_{i-1} sera étendu en 48 bits grâce à une matrice appelée table d'expansion (notée E), dont les 48 bits sont mélangés et 16 d'entre eux sont dupliqués (voire Figure I.7).

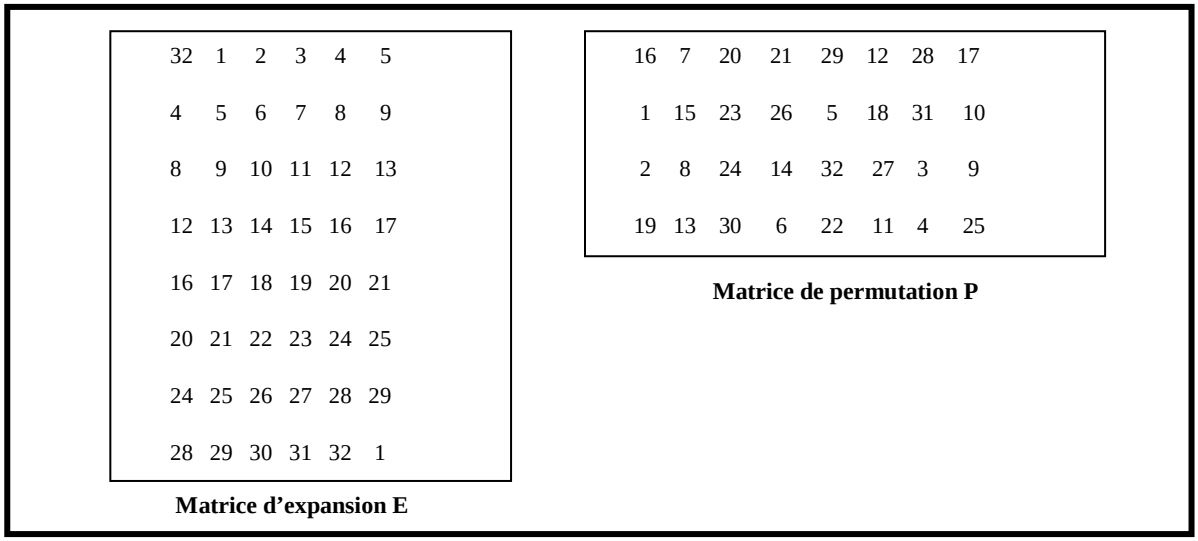


Figure I.7. *Matrice d'expansion E et de permutation P.*

L'opération qui suit consiste à calculer un « ou exclusif » entre le résultat obtenu jusqu'à maintenant, codé sur 48 bits, et G_{i-1} codé sur 32 bits. Donc, le résultat final de *f* doit être codé sur 32 bits au lieu de 48 bits. Pour cela, la chaine de $48 = 8 \times 6$ bits sera transformée en une chaîne de $32 = 8 \times 4$ bits en utilisant des dispositifs appelés *boites-S*. Elles sont au

nombre de huit, où chacune calcule un bloc de 4 bits à partir d'un bloc de 6 bits. Enfin, on applique une permutation à ce 32 bits pour obtenir la valeur finale de *f* (voir la Figure I.7).
L'ensemble de ces résultats en sortie de P est soumis à un « ou exclusif » avec le G_0 de départ (comme il est indiqué dans la Figure I.9) pour donner D_1, tandis que le D_0 initial donne G_1, et ainsi de suite pour les 16 itérations comme il est mentionné auparavant.
La succession d'opérations constituant la fonction ***f*** est représentée à travers la Figure I.8**.**

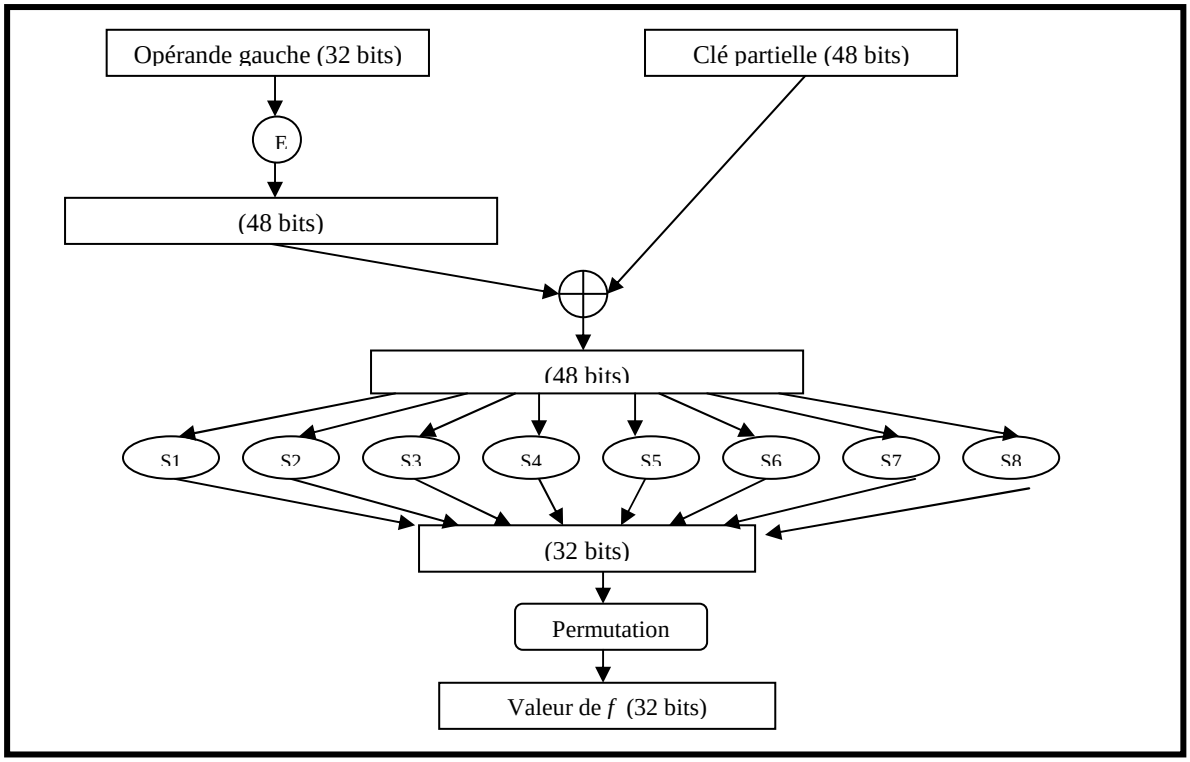


Figure I.8. *Schéma de la fonction f.*

4. **Permutation initiale inverse :** A la fin des itérations, les deux blocs G_{16} et D_{16} sont récoltés, puis soumis à la permutation initiale inverse : $Z = PI^{-1}(G_{16}\ D_{16})$. Le résultat en sortie est un texte codé de 64 bits.
   Cette description peut être résumée par le schéma général illustré par Figure I.9, où on a seulement représenté quelques-unes des 16 étapes.

Remarque : le déchiffrement suit le même algorithme avec la même clef *K*. Seules les sous-clés sont appliquées dans le sens inverse en commençant par la clé k_{16} jusqu'à la clé k_1.

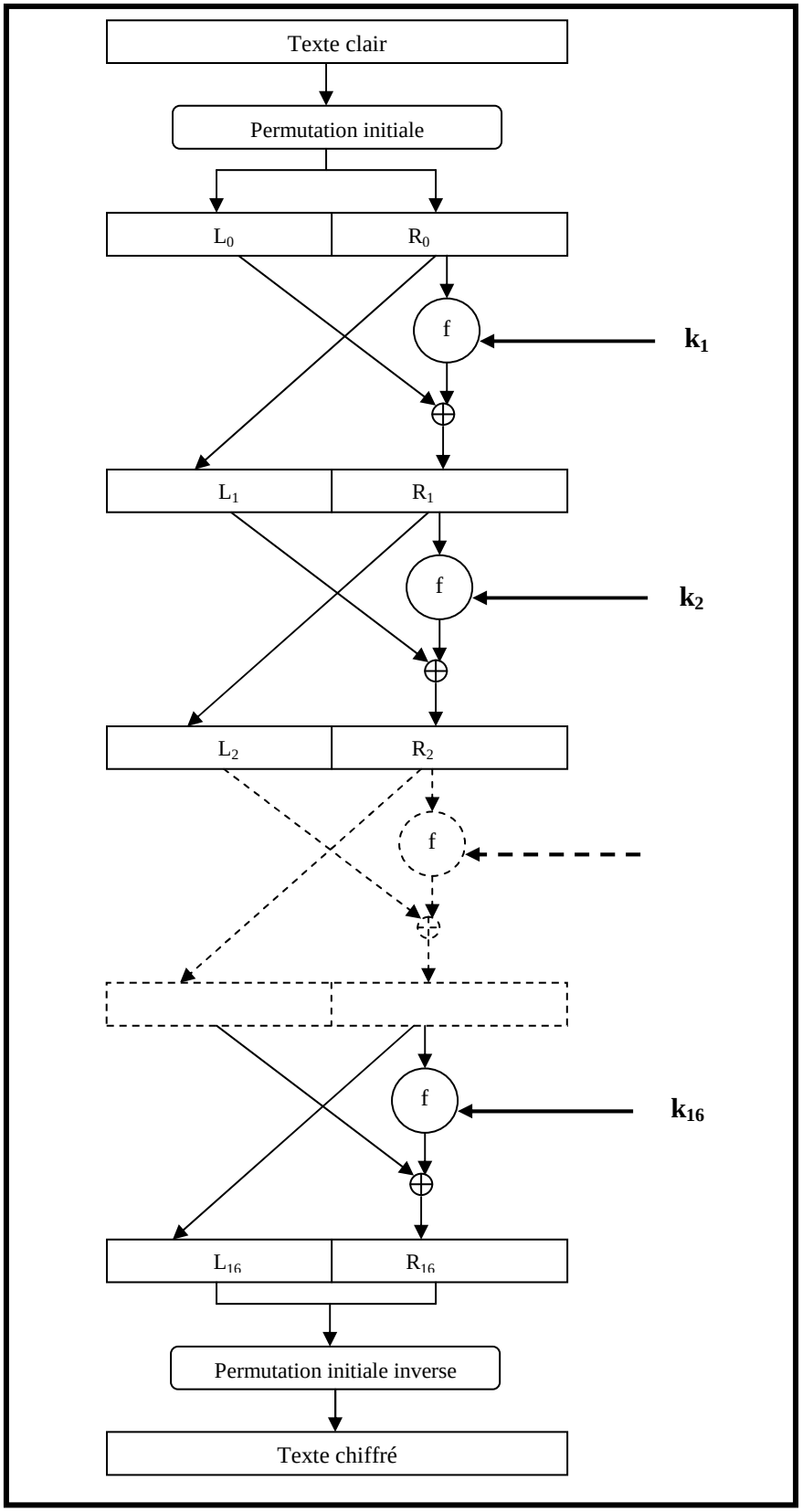


Figure I.9. *Schéma général de DES.*

Le DES a fait l'objet de très nombreuses attaques. On peut citer quelques une :

- **Cryptanalyse différentielle :** c'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires. Ainsi, un DES à 8 ou à 10 tours peut facilement être cassé, mais le DES complet à 16 tours est resté hors de portée de cette attaque.

- **Cryptanalyse linéaire :** c'est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Elle n'est utilisable que pour un DES restreint à quelques tours, mais le DES réel n'est pas menacé par cette attaque.

- **Recherche exhaustive :** les laboratoires RSA ont lancé en Janvier 1997 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES pour démontrer que la taille des clés DES a devenu insuffisante. Ils ont réussi le 17 Juin 1997. Bien qu'une telle recherche demande des moyens considérables, elle a montré que le DES n'offre plus aujourd'hui une grande sécurité.

3) Triple DES (3DES)

Pour palier l'insuffisance cryptographique observée du cryptosystème DES, dû à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. Triple DES est apparait comme une solution pour remédier les faiblesses de DES. Comme son nom l'indique, le principe du triple DES est d'effectuer 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Ce principe peut être formulé comme suit :

$$\textbf{Triple-DES}_{k1,\,k2} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}^{-1}_{k2} \;\textbf{O}\; \textbf{DES}_{k1}$$

On peut constater que le DES sera retrouvé comme cas particulier de la formule ci-dessus, lorsque $k_1 = k_2$. Le déchiffrement de son tour est formulé par :

$$\textbf{Triple-DES}^{-1}_{k1,\,k2} = \textbf{DES}^{-1}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}^{-1}_{k1}$$

Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc composée de 112 bits. Une autre variante à trois clés DES différentes peut être conçue. D'une façon plus formelle, son principe peut être donné par la formule suivante :

$$\textbf{Triple-DES}_{k1,\,k2,\,k3} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}_{k3}$$

Malgré cela, cette variante reste aussi fragile à une attaque de coût en 2^{112} s'appuyant sur l'un des deux messages intermédiaires.

4) AES (**Advanced Encryption Standard**)

Avec le temps et les progrès de l'informatique, les 2^{56} clés possibles du DES se voient incapables de fournir une barrière infranchissable dû à la faiblesse des clés de 56 bits. Désormais avec des moyens modestes, percer les messages chiffrés par DES en un temps raisonnable, ne pose aucun problème. De ce fait, l'AES a vu le jour. Il est issu d'un appel à candidatures international lancé en janvier 1997 et ayant reçu 15 propositions. Au bout de cette évaluation, ce fut le candidat Rijndael du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen [www4] qui a été choisi par le NIST en Octobre 2000 pour être l'algorithme AES et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. II a été déclaré vainqueur de la deuxième ronde dans laquelle

s'opposaient les 5 candidats finalistes (MARS, RC6, Rijndael, Serpent, TwoFish). L'AES a une taille longue de sa clé allant jusqu'à 256 bits qu'il le permet de résister toujours a la cryptanalyse. L'AES est devenu le nouveau standard du chiffrement symétrique comme le signifie cet acronyme. L'AES est libre d'utilisation comme l'est l'algorithme DES.

Techniquement, le chiffrement AES travaille avec des blocs de 128 bits seulement et la longueur des clés utilisées peut être de 128, 192 ou de 256 bits. Chaque bloc subit une séquence de transformations que nous résumons à travers les points suivants :

1. Addition de la clé secrète et du bloc en question avec un « ou exclusif ».
2. ByteSub : les 128 bits sont répartis en 16 blocs de 8 bits (16 octets), qui sont ensuite placés dans une matrice de 4×4. Chaque octet est transformé par une fonction non linéaire S (S-box) conçu pour résister à la cryptanalyse linéaire et différentielle.
3. ShiftRow: les lignes de cette matrice sont soumises à une rotation vers la droite où l'incrément pour la rotation varie selon le numéro de la ligne. La $2^{ème}$ ligne est décalée d'une colonne, la $3^{ème}$ ligne de 2 colonnes, et la $4^{ème}$ ligne de 3 colonnes.
4. MixColumn: chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne. Cela revient à multiplier la matrice 4×4 par une autre matrice 4×4.
5. AddRoundKey: une clé dite *de tour* est générée à partir de la clé secrète par un sous-algorithme dit *de cadencement*. Cette clé de tour est ajoutée par un « ou exclusif » au dernier bloc obtenu.

Ces différentes opérations, définissant un *tour*, sont répétées plusieurs fois. Cependant dans le dernier tour il n'y a pas d'opération MixColumn. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours. La figure suivante résume le principe de fonctionnement de cet algorithme de chiffrement.

Le déchiffrement consiste en appliquer les opérations inverses dans chacune des étapes (*InvSubBytes*, *InvShiftRows*, *InvMixColumns*). *AddRoundKey* (à cause du XOR) est son propre inverse. On réitère ce processus le nombre de tour-1. Pour le dernier tour on exclu l'opération *InvMixColumns* ; et comme dans le chiffrement pas d'opération *InvMixColumns* dans le dernier tour.

De nombreux travaux de cryptanalyse ont été publiés mais pour l'instant il n'a pas été cassé et la recherche exhaustive demeure la seule solution. En particulier, on cite :

- **Attaques sur des versions simplifiées :** *Niels Ferguson* et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours. En effet, cette dernière attaque repose sur le principe des clés apparentées. Dans une telle attaque, la clé demeure secrète mais l'attaquant peut spécifier des transformations sur la clé et chiffrer des textes à sa guise. Il peut donc légèrement modifier la clé et regarder comment la sortie de l'AES se comporte.

- **Attaques sur la version complète :** plusieurs chercheurs ont mis en évidence des possibilités d'attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL. Le XSL fait appel à une analyse heuristique dont la réussite n'est pas systématique. Elles sont impraticables car le XSL demande au moins 2^{87} opérations voire 2^{100} dans certains cas. Le principe est d'établir les équations (quadratiques / booléennes) qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer et l'AES est toujours considéré comme sûr.

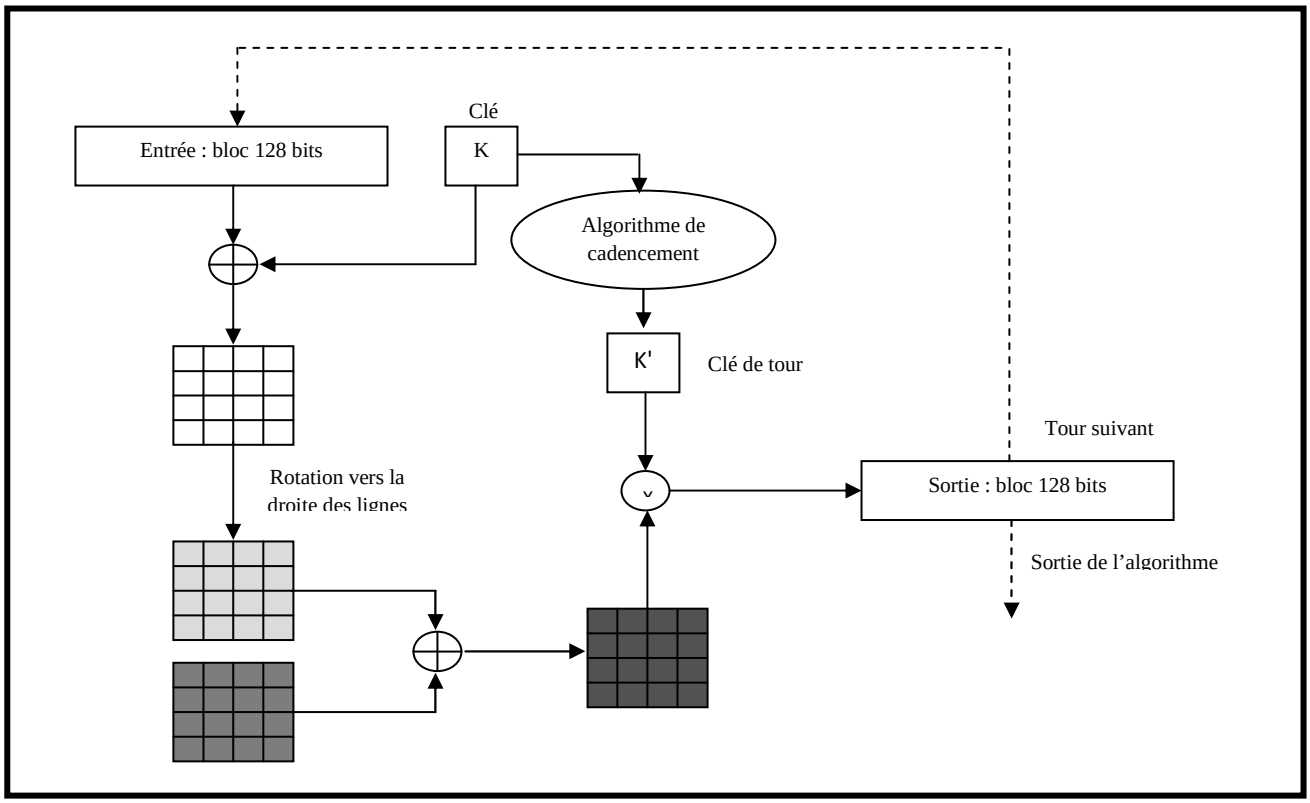


Figure I.10. *Schéma général de l'AES.*

5) IDEA (International Data Encryption Algorithm)

L'**IDEA** qui est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 rondes inventé en 1990 à Zürich par James L. Massey et Xuejia Lai. C'est la taille des clés qui apporte une grande sécurité au système (que l'on n'a pas, à l'heure actuelle, officiellement réussi à "casser").

Chaque bloc est divisé en 4 sous-blocs de 16 bits : X_1, X_2, X_3 et X_4. Ces quatre sous-blocs deviennent les entrées de la première ronde de l'algorithme.
À chaque ronde, la séquence d'évènements est la suivante (voir Figure I.11) :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé ;
5. combinez par OU exclusif les résultats des étapes (1) et (3) ;
6. combinez par OU exclusif les résultats des étapes (2) et (4) ;
7. multipliez le résultat de l'étape (5) avec la cinquième sous-clé ;
8. additionnez les résultats des étapes (6) et (7) ;
9. multipliez le résultat de l'étape (8) par la sixième sous-clé ;
10. additionnez les résultats des étapes (7) et (9) ;
11. combinez par OU exclusif les résultats des étapes (1) et (9) ;
12. combinez par OU exclusif les résultats des étapes (3) et (9) ;
13. combinez par OU exclusif les résultats des étapes (2) et (10) ;
14. combinez par OU exclusif les résultats des étapes (4) et (10).

La sortie de la ronde est constituée des 4 sous-blocs produits par les étapes (11), (13), (12) et (14). Changez les deux blocs intérieurs (sauf lors de la dernière ronde) et cela donne l'entrée de la ronde suivante.
Après la huitième ronde, il y a une transformation finale :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé.

Enfin les 4 sous-blocs sont réassemblés pour former le texte chiffré.

Chaque ronde utilise 6 sous-clés K_i^r de 16 bits : K_1^1, ..., K_6^1, …, K_1^8, …, K_6^8. La transformation finale utilise 4 sous-clés : $K_1^9, K_2^9, K_3^9, K_4^9$; donc un total de 52 sous-clés est utilisé, les premiers 8 sous-clés sont extraites directement à partir de clé, des groupes de 8 clés sont extraits par rotation à gauche de 25 bits de la clé K, ce qui donne 6 rotations en total.
Concernant le processus de déchiffrement on utilise celui de chiffrement avec un seul changement dans la génération des clés. En fait, on utilise K pour générer les sous clés K_i^r ; à partir de ces derniers, d'autres clés $K_i^{'r}$ sont obtenues dans le Tableau I.1 [Omar, 2006] ; ensuite on utilise $K_i^{'r}$ à la place des K_i^r dans l'algorithme de chiffrement IDEA. Dans le Tableau I.1, $-K_i$ dénote l'opposé modulo 2^{16} de K_i. K_i^{-1} dénote l'inverse multiplicative de $K_i \bmod(2^{16}+1)$, se trouvant aussi dans {0, 1, …,216-1}.

Ronde r	$K_1^{'(r)}$	$K_2^{'(r)}$	$K_3^{'(r)}$	$K_4^{'(r)}$	$K_5^{'(r)}$	$K_6^{'(r)}$
r =1	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
2 ≤ r ≤ 8	$(K_1^{10-r})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
r= 9	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	--	--

Tableau I.1. *Les sous clés de déchiffrement $K_i^{'r}$ générées à partir des sous clés K_i^r.*

La méthode de génération des sous-clés de l'IDEA est toujours régulière et donc pourrait être une faiblesse à l'algorithme. Cependant, il est considéré comme étant hautement sécuritaire. En effet, pour résoudre IDEA avec l'attaque en force brute, il faudrait effectuer 2^{128}, donc 10^{38} opérations [www5].

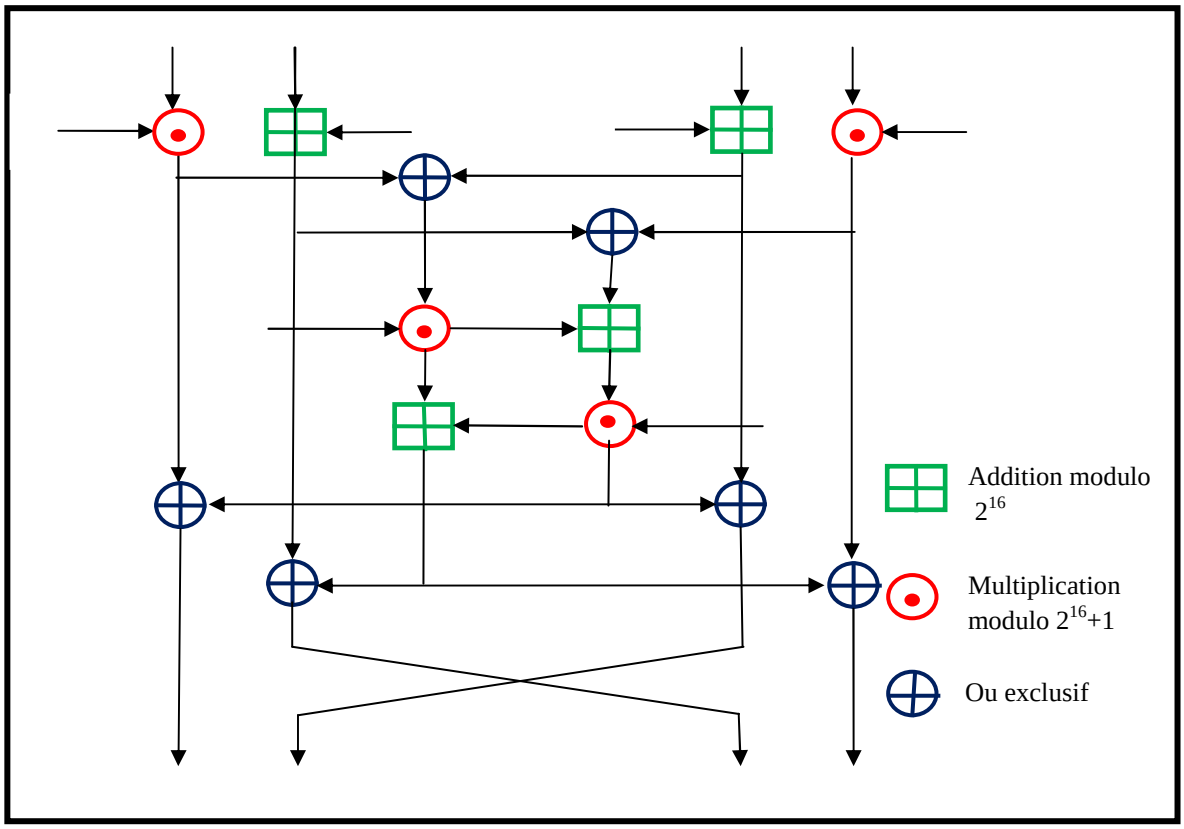


Figure I.11. *Schéma général de l' IDEA.*

- **Conclusion**

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. AES est immunisé contre les attaques et il reste néanmoins le meilleur choix dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause. D'autre part, IDEA est considéré par les spécialistes comme l'un des meilleurs cryptosystèmes à clé privée. Il est utilisé par le PGP pour le chiffrement de données.

b. La cryptographie asymétrique

- **Principe**

La *cryptographie asymétrique* ou encore dite *à clé publique* utilise deux clés différentes pour chaque utilisateur : une est privée et n'est connue que de l'utilisateur et qui est sensé d'être en mesure de faire signature ou déchiffrement. L'autre est publique diffusée en général dans un annuaire et donc accessible par quiconque afin de permettre aux interlocuteurs de mettre en œuvre les opérations réciproques (vérification de signature ou chiffrement de message). Les clés, publique et privée sont mathématiquement liées par l'algorithme de cryptage de telle manière qu'un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de *fonction à sens unique avec trappe*. Sachons qu'une fonction est appelée à *sens unique* si elle est

aisément calculée, mais extrêmement difficile de déduire la fonction inverse. Et elle sera dite *à trappe*, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la *trappe*. L'utilité d'utilisation d'une telle fonction réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas assez facile puisqu'elles s'appuient généralement sur des problèmes mathématiques réputés difficiles.

Ce cryptage présente l'avantage de permettre le placement de signature numérique dans le message et ainsi permettre l'authentification de l'émetteur grâce à la fonction de hachage. Le principal avantage consiste à résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé puisque la clé privée n'est connue que par l'utilisateur. Bien que plus lent que la plupart des cryptages à clé privée il reste toujours préférable pour 3 raisons :

- ✓ plus évolutif pour les systèmes possédant des millions d'utilisateurs.
- ✓ authentification plus flexible.
- ✓ supporte les signatures numériques.

Les systèmes asymétriques les plus connus sont [Mene, 1996] : RSA basé sur la difficulté de la factorisation des grands entiers, El Gamal basé sur la difficulté de résoudre le problème du logarithme discret dans un corps fini et les systèmes sur les courbes elliptiques basés sur la difficulté de certains calculs sur les courbes elliptiques.

- **Quelques algorithmes asymétriques**

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivants :

1) RSA

La méthode de cryptographie RSA a été inventée en 1977. Elle titre son nom des noms de ses trois inventeurs: *R. Rivest, A. Shamir* et *L. Adleman* [www6]. Le RSA est le premier et est encore le système cryptographique à clé publique le plus utilisé de nos jours et toujours considéré comme sûr. Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers ; à l'heure actuelle, il est pratiquement impossible de les reconstituer en un temps raisonnable. Ainsi, la sécurité de RSA semble satisfaisante malgré qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser.

On peut résumer le fonctionnement de ce cryptosystème dans les étapes suivantes :

- **Fonction d'encodage E (publique) :** la clé publique k utilisée pour l'encodage comporte deux entiers: k = (e,n). L'opération de chiffrement se fait au moyen de l'élévation à la puissance **e** modulo n : $E_k(M) = M^e \bmod n$.
- **Fonction de décodage D (privée) :** la clé secrète k′ utilisée pour le déchiffrement est aussi un couple d'entiers : k′(d,n). Pour reconstituer le message initial, la fonction inverse d'encodage est appelée, elle est ainsi : $D_{k'}(M) = M^d \bmod n$.
- **Détermination des clés :**
  - **Détermination de n :** pour calculer n on doit initialement choisir deux entiers premiers *p* et *q* très grands et leurs valeurs sont secrètes connus que part l'utilisateur. Le choix de *p* et *q* affecte grandement le niveau de sécurité de RSA. Pour cela, il faut évidemment se prémunir contre les algorithmes de factorisation dont la complexité dépend essentiellement de la taille du plus petit facteur premier de *n* [Zimm, 2005], donc si possible choisir *p* et *q* de même taille.

- **Détermination de e :** pour calculer e, on calcule premièrement un entier z = (p-1)*(q-1) puis tout simplement choisir un entier e premier avec z.
- **Détermination de d :** pour calculer d, on procède ainsi : e*d≡1[z].

La sécurité de RSA est basée sur l'hypothèse que la fonction E est à sens unique, ce qui rend impossible à décrypter un texte chiffré. Mais à l'aide de la trappe qui est la factorisation n = p*q, il est possible d'en trouver d et donc la clé privée.

Depuis son apparition, plusieurs attaques ont été découvertes contre le RSA. Et même si aucune de ces attaques n'est réellement destructive, elles démontrent toutefois qu'il faut implémenter RSA avec beaucoup de précautions. La fameuse attaque est :

- **Recherche exhaustive :** comme la fonction de chiffrement RSA est déterministe, si l'ensemble des messages possibles est connu et de petite taille, il sera facile de décrypter par une recherche exhaustive [Ster, 2004]. Pour éviter cette attaque, il est indispensable de randomiser les messages avant chiffrement.

2) Chiffrement d'ElGamal

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste c.à.d. un même message M peut avoir plusieurs chiffres différents [Lafo, 2006]. Il est basé sur la difficulté du problème des logarithmes discrets c.à.d. la difficulté de trouver l'unique *a* noté $\log_\alpha ß$ / $0 \leq a \leq p-2$ tel que : $\alpha^a \equiv ß \pmod p$ où *p* est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$. Cependant, et pour éviter les attaques connues, *p* doit être convenablement choisi, et $p-1$ doit avoir un grand facteur premier.

D'une manière formelle, l'algorithme ElGamal peut être résumé comme suit :

Soit *p* un nombre premier tel que le problème du logarithme discret dans Z_p soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et $K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod p\}$
Les valeurs *p*, α et ß sont publiques, et *a* est secret.
Pour k = (*p*, α, *a*, ß), et pour un $k \in Z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$\mathbf{e_k(x, k) = (y_1, y_2)}$$

Où

$$\mathbf{y_1 = \alpha^k \bmod p}$$

Et

$$\mathbf{y_2 = x\, ß^k \bmod p}$$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit comme suit :

$$\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$$

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par $ß^k$, en produisant y_2 ;
- ✓ la valeur α^k est également transmise en tant que partie du texte chiffré ;
- ✓ Bob, qui connaît l'exposant secret *a*, peut calculer $ß^k$ à partir de α^k. Il peut alors « enlever le masque » en divisant y_2 par $ß^k$ et obtenir le texte clair x.

Une attaque possible à ce cryptosystème est celle dite ***man in the middle***. Son principe dans le cas d'ElGamal fonctionnant avec le mode *Diffie-hellman* est illustré sur la figure I.12.

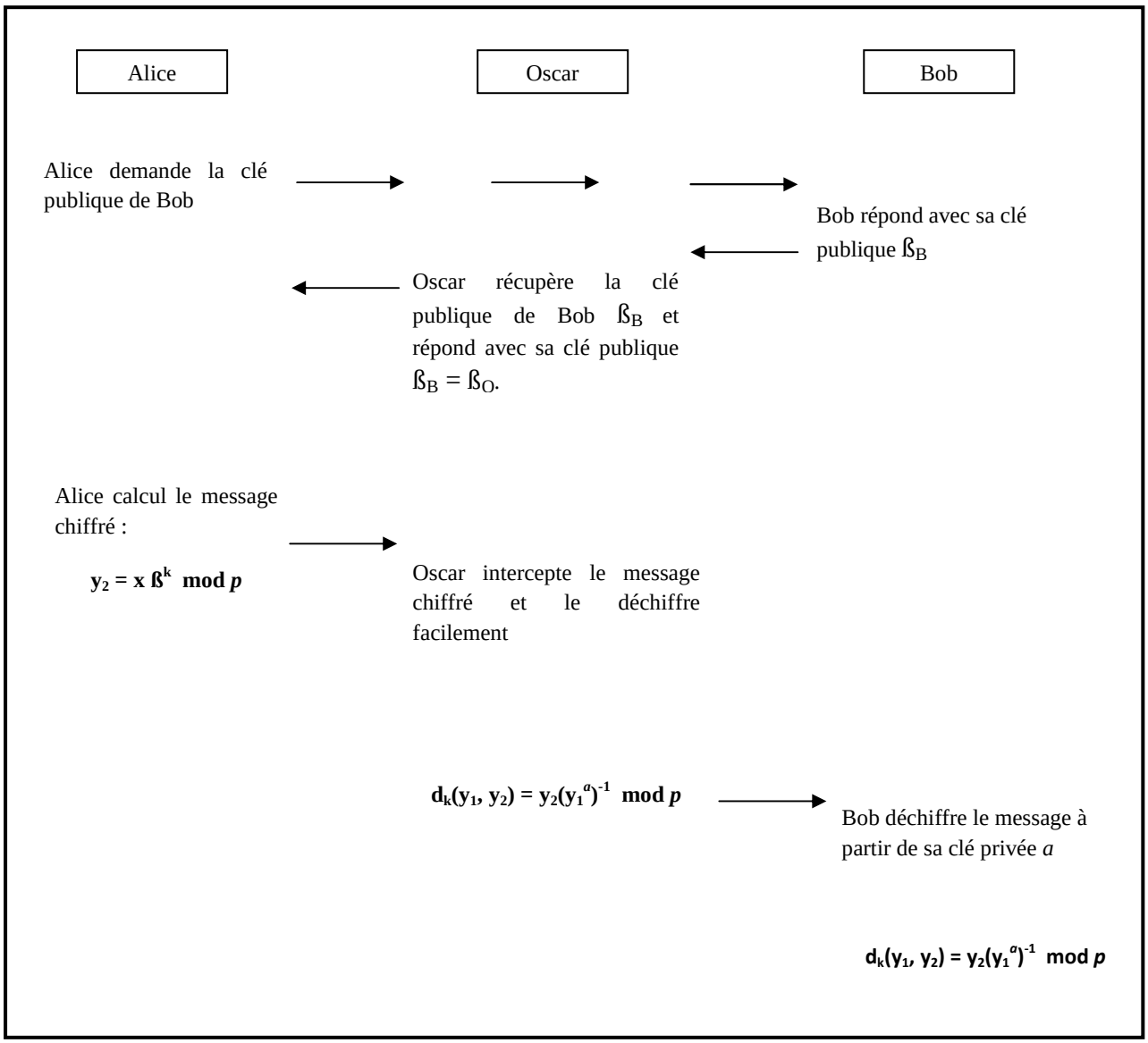


Figure I.12. *Principe de l'attaque « man in the middle ».*

- **Conclusion**

Les calculs faits en 1995 ont ouvert un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent des chiffres plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique serait capable de remettre en cause ce cryptosystème.

De même, casser l'algorithme ElGamal est aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Et pour assurer une bonne sécurisation de cet algorithme, Zimmermann en 2005 [Zimm, 2005] a recommandé l'utilisation des clés d'au moins 1024 bits.

c. Comparaison entre les cryptosystèmes symétriques et asymétriques

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage.

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ?
À partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que l'importance de la taille de clé n'est légitime que dans le cas de la clé secrète, puisque les seules attaques possibles sont les attaques exhaustives [www7]. Mais, dans le cas de la clé publique, la taille de la clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

Méthode	Avantages	Inconvénients
À clefs Secrètes	▪ Rapidité de calcul en général (dépend de la taille de la clé). ▪ Adaptée au cryptage de flux de données.	▪ Moins sécurisé (DES). ▪ Problème de communication de clefs entre émetteur et récepteur. ▪ Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clefs Publiques	▪ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre. ▪ Permet la signature électronique. ▪ Un couple de clés publique/privée suffisant pour 'n' correspondants.	▪ Lente. ▪ Problèmes de gestion de clefs publiques.

Tableau I.2. *Comparaison entre les méthodes de chiffrement symétriques et asymétriques.*

d. La cryptographie hybride

- **Principe**

La cryptographie asymétrique est beaucoup plus lente que la cryptographie symétrique qui brille par sa rapidité. En revanche, cette dernière souffre d'une grave lacune ; assurer une transmission secrète de la clé. Pour palier ce défaut et cumuler les avantages des deux méthodes on a fait recourt à la cryptographie hybride. On code tout d'abord les données avec une clé privée dite *clé de session*, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Comme la clé est courte, on utilise l'algorithme asymétrique puisqu'il prend peu de temps. En revanche, chiffrer l'ensemble du message avec un algorithme asymétrique serait plus lourd. Il suffit ensuite d'envoyer le message chiffré avec une clé privée et accompagné de cette dernière chiffrée avec une clé publique. Le destinataire procède inversement, il commence à déchiffrer la clé symétrique avec sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, via un déchiffrement symétrique pour retrouver le message original. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

- **Quelques algorithme hybride**

1) PGP (**Pretty Good Privacy**)

PGP est inventé par *Philip Zimmermann*, qui dise que tout individu a droit à la confidentialité, notamment les organisations des droits de l'homme dans des pays soumis à la dictature. Il l'a mis à disposition gratuitement sur Internet. Cela lui a valu de sérieux ennuis avec la justice américaine, car les logiciels de cryptage sont considérés comme du matériel de guerre et sont interdits à l'exportation.

Il est souvent utilisé pour garantir l'authentification et le contrôle d'intégrité des fichiers et du courrier électronique. Mais avant de crypter un texte avec PGP, les données doivent tout d'abord compressées dont le but de réduire le temps de transmission, ainsi d'économiser l'espace disque et, surtout, de renforcer la sécurité cryptographique puisque les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement alors que la compression réduit ces modèles dans le texte en clair.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes qui sont [www8] :

✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
✓ PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de déchiffrement se fait également en deux étapes, qui sont [www8] :

✓ PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
✓ PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

Depuis 1978, la recherche universitaire civile a intensément attaqué la cryptographie à clé publique, sans pour autant réussir à la remettre en cause. Mais cela ne fournisse aucune garantie totale sur la sécurité de cette manière de chiffrer, car une attaque menée par le gouvernement, par exemple, qui ne se manque pas de ressources très développées ou même en utilisant quelques nouvelles percées mathématiques classées top-secret, peut tenir à bout ces cryptosystèmes conventionnels utilisés dans PGP.

Tout de même, l'optimisme semble justifié. Les algorithmes de clé publique, les algorithmes de contraction de message, et les chiffres par blocs utilisés dans PGP ont été conçus par les meilleurs cryptographes du monde [Loid, 2005]. Les chiffres de PGP ont subi des analyses de sécurité approfondies et des examens méticuleux de la part des meilleurs cryptographes dans le monde non classé top secret. De plus, et même si les chiffres par blocs utilisés dans PGP possèdent quelques faiblesses, la compression du texte clair utilisée avant le chiffrement réduit de façon considérable ces faiblesses.

2) GPG (**GNU Privacy Guard**)

GPG est la version GNU de PGP. En raison qu'il est sous licence GPL (General Public License) il permet la fourniture du code-source, la libre modification et la libre redistribution de ce code-source. Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

- **Conclusion**

Aujourd'hui, de nombreux gouvernements ont restreint l'usage du PGP, qui a été largement diffusé par son développeur, en pensant qu'un cryptage trop fiable ferait le jeu des terroristes et des trafiquants. Toutefois, il y'a pas mal d'applications qui utilisent encore ce système de chiffrement, telles que les paiements en ligne qui se font grâce au procédé SSL

fonctionnant selon le principe du PGP. De sa part, GPG, qui est un cryptosystème utilisant des algorithmes de chiffrement à clé publiques (DSA, RSA et ElGamal), est largement utilisé dans les communications par messagerie, c.-à-d. pour chiffrer des mails, ainsi que pour signer des données.

I.3.3.3. Cryptographie quantique

a. Principe

La cryptographie à base d'algorithmes aura toujours des faiblesses. Même si la cryptanalyse bloque devant un algorithme, la force brute pourra toujours décrypter n'importe quel code si on lui donne assez de temps. Même avec le plus solide des chiffrages, le contenu du message peut être subtilisé et dupliqué. Les clés, quant à elles, peuvent être volées en chemin ou présenter des faiblesses qui les rendent prévisibles.

Si maintenant on base notre cryptographie, non pas sur un algorithme mathématique, mais sur des lois de la physique quantique, il n'y plus de force brute qui peut briser un code ici. Le cryptage ne se trouve pas dans des formules, mais dans des photons, ainsi tout le monde peut accéder à des formules, mais pas tout le monde peut accéder aux photons sans que le message devienne inutile. De ce fait, l'information n'est plus sécurisée par des subterfuges mathématiques, mais plutôt par des lois fondamentales de physique. Au de-là, la cryptographie quantique a vu le jour.

La figure I.13 [Nave, 2002] présente un exemple relatif à la possibilité soit qu'un photon traverse le filtre ou pas, selon l'orientation de sa polarisation. Sachons qu'un filtre permet de distinguer entre les photons polarisés horizontalement (0°) et verticalement (90°) ; un autre entre les photons polarisés en diagonale (45°, 135°).

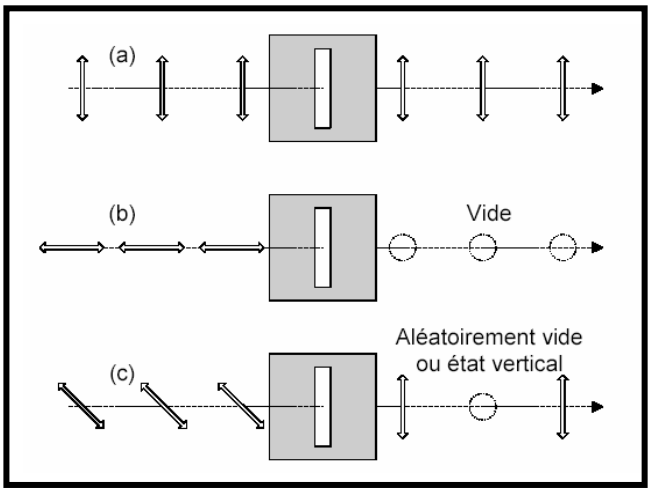


Figure I.13. *Photon unique traversant un filtre ne laissant passer que la lumière polarisée verticalement :*
(a) *Les états polarisés verticalement traversent le filtre sans être absorbés*
(b) *Les états polarisés horizontalement sont tous absorbés*
(c)*Les états polarisés diagonalement sont aléatoirement absorbés ou transmis*

Pour mieux comprendre ce type de cryptographie, on va l'aborder à partir d'un exemple dont le protocole est bien : le protocole BB84. Avant de commencer, on doit mentionner qu'il existe deux modes de polarisations possibles [www9] :

- **Mode 1 :** "0" est codé par un photon d'axe de polarisation 0° et "1" par un photon de polarisation 90°.
- **Mode 2 :** "0" est codé par un photon d'axe de polarisation 45° et "1" par un photon de polarisation 135°.

Pour plus de clarté, ces deux modes peuvent être schématisés comme suit :

Figure I.14. *Les deux modes de polarisation.*

Cet exemple se déroule en plusieurs étapes que l'on va présenter ci-dessous. On a Alice l'expéditrice, Bob le destinataire et Oscar l'éventuel espion [www9]:

1. Alice génère aléatoirement un bit selon un mode (mode 1 ou mode 2) choisi lui aussi aléatoirement, ce que la physique quantique permet, et transmet le photon obtenu à Bob. Elle répète cette opération autant de fois que nécessaire.
2. Pour chaque photon reçu, Bob choisit aléatoirement un mode, c'est-à-dire un polariseur, et note si le photon traverse le polariseur.
3. Ensuite une phase de réconciliation : Alice et Bob communiquent entre eux et pour chaque photon ils comparent si leurs modes coïncident, si c'est le cas, ils ont le même bit, qu'ils conservent, sinon ils ne conservent pas le bit en question. Vue le nombre de photons inutiles, il est donc nécessaire de prévoir un excédent suffisant de photons pour obtenir une clé de longueur donnée. Comme Bob a une chance sur deux de choisir le bon mode, en moyenne on observe ***N*** erreurs pour ***2N*** photons envoyés.
4. Enfin, Alice et Bob contrôlent la sûreté de la clé : parmi les bits conservés ils en choisissent un certain nombre qu'ils comparent publiquement, s'ils ont été espionnés ils obtiendront en moyenne 25% de bits différents (Comme Oscar ne peut pas cloner les photons, il est dans l'obligation de les intercepter, de les mesurer et de renvoyer à Bob un photon similaire à celui qu'il a mesuré. Cependant il a une chance sur deux de choisir le mauvais mode et donc de renvoyer un photon différent de celui qu'Alice avait envoyé à Bob. Puis comme Bob a lui aussi une chance sur deux d'avoir le bon mode, en cas d'espionnage pour chaque photon on a une chance sur 4 (25%) de détecter l'intrusion). Dans ce cas, ils n'utiliseront pas la clé. S'ils n'ont pas de différences, alors ils peuvent conserver la clé pour l'utiliser ultérieurement.

Alice et Bob décident alors de sacrifier une partie de leur clé commune et les comparent publiquement par le canal radio (comme exemple de canal). Dans le Tableau I.3 ils ont quatre bits différents c.à.d. N bits parmi 2N. Ainsi la clé obtenue : 0011.

Photon envoyé par Alice								
Mode choisi par Bob	Mode 1	Mode 2	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								
Après réconciliation								

Tableau I.3. *Exemple sans Oscar.*

Photon envoyé par Alice								
Mode choisi par Oscar	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 2	Mode 2
Résultat mesuré et renvoyée à Bob par Oscar								
Mode choisi par Bob	Mode 1	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								

Tableau I.4. *Exemple avec Oscar.*

Plusieurs cas de figure se présentent à nous dans le Tableau I.4 [Camp, 2010] :

1. Oscar et Bob ont tous les deux le bon mode, alors Oscar mesure bien la bonne polarisation qu'il renvoie à Bob et puis de même pour Bob. Le bit est valable pour la clé et Oscar est passé inaperçue. Exemple : premier photon envoyé.
2. Oscar a le bon mode, mais pas Bob, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon. Oscar passe encore inaperçue, mais sa bonne mesure lui est inutile. Exemple : le deuxième photon envoyé.
3. Oscar a le mauvais mode, mais Bob a le bon mode, alors d'après ce que l'on a vu, Bob a une chance sur deux d'obtenir une mesure compatible avec ce qu'à envoyer Alice. Donc Oscar a une chance sur deux d'être détecté, de plus sa mesure est inutile. Exemple : sixième photon envoyé (avec détection) et dernier photon envoyé (sans détection).
4. Oscar et Bob ont le mauvais mode, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon, Oscar passe donc inaperçue. Exemple : cinquième photon envoyé.

Une fois la réconciliation terminée il ne reste que les bits de la possibilité 1 et 3 qui sont équiprobables. Pour la première possibilité Oscar passe inaperçue, et pour la troisième il a une chance sur deux de se faire détecter. On retrouve bien le 25% ($\frac{1}{2}\frac{1}{2} = \frac{1}{4}$) vu ci-dessus.

b. Conclusion

Le MIT (Massachussets Institute of Technology) a récemment classé la cryptographie quantique parmi les «10 inventions qui changeront le monde». Si on voit peu les avantages à l'heure actuelle, les possibilités pour le futur sont impressionnantes. Le centre canadien IRCA est un leader mondial en recherche quantique, comptant parmi ses membres des chercheurs montréalais au front de cette révolution du monde des télécommunications. Avec le développement du calcul et de la cryptographie quantique, on peut espérer qu'Alice et Bob pourront un jour communiquer en toute quiétude. Entre temps, considérons d'un œil sceptique les soi-disant algorithmes « incassables » qui n'attendent que le prochain superordinateur pour céder comme une coquille d'œuf [Camp, 2010].

I.3.3.4. Algorithme de chiffrement évolutionniste OTL

Omary Fouzia a développé en 2006 un nouvel algorithme de chiffrement en le simulant à un problème d'optimisation dont la résolution est par les algorithmes génétiques. Le but de cet algorithme est de modifier au maximum les fréquences d'apparition des caractères dans le message à chiffrer et d'établir le plus de désordre dans leurs positions. Ces caractères appartiennent à l'ensemble des 256 caractères du code ASCII. L'application de l'algorithme est précédée d'une phase de brouillage du texte initial (M_0) qui peut combiner plusieurs méthodes simples comme les substitutions, les permutations, le chiffrement affiné, etc..., pour obtenir un texte initialement chiffré (M_0').

Pour ce faire, le codage adopté pour représenter les individus, qui sont les différents messages, est résumé à travers la figure ci-dessous :

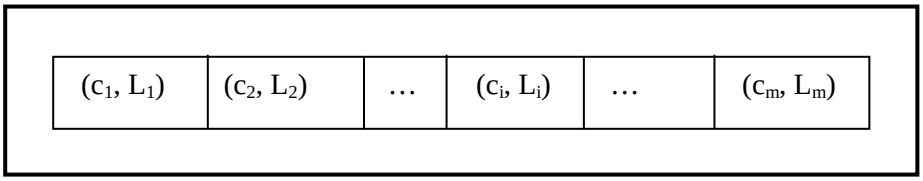


Figure I.15. *Codage des individus.*

Avec :

c_i : un caractère appartenant au message.
L_i : la liste des positions du caractère c_i.
m : le nombre des différents caractères du message.
$L_i \cap L_j = \emptyset, \ \forall i, j \in [1, m]$

L'algorithme OTL cherche à changer itérativement la répartition des listes L_i sur les différents caractères du message (sans modifier le contenu des listes) de telle manière que la différence entre le cardinal de la nouvelle liste affectée à chaque caractère c_i et le cardinal de la liste L_i d'origine soit maximale [Omar, 2006]. Cette variation, en terme de répartition, est assurée grâce aux opérateurs génétiques choisis (MPX pour le croisement avec un taux compris entre 60% et 100%, et une simple permutation pour la mutation avec un taux compris entre 0.1% et 5%). Et pour juger la pertinence des individus construits, la fonction

d'évaluation utilisée est celle donnée ci-dessous. Les meilleurs individus sont ensuite sélectionnés par le biais de la méthode classique de la roulette, en vue de se reproduire.

$$F(X_j) = \sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right|$$

Le processus évolutionnaire est répété jusqu'à la satisfaction d'un critère d'arrêt exprimé à l'aide de la fonction F. Cette dernière est bornée car : $0 \le F(X) \le 2 * l$ (l est la taille du message) pour tout individu X. En fait [Omar, 2006] :

$$\sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right| \le \sum_{i=1}^{m} (card(L_{j_i}) + card(L_i)) \le 2 * l$$

L'opération de déchiffrement, quand à elle, se fait en deux étapes :

1) Tout d'abord, le texte chiffré est représenté suivant le codage proposé en une suite de listes de positions, et c'est grâce à la clé génétique que les caractères vont retrouver leurs listes de position correspondantes dans le message en clair. En effet, la clé, qui peut être d'un usage symétrique ou asymétrique, est une permutation de {1,2,…, m}. Comme résultat, nous obtenons le message M_0'.
2) La deuxième étape, correspond au déchiffrement du message M_0' pour obtenir M_0.

I.4. Conclusion

Dans ce chapitre, nous avons présenté les différentes catégories de cryptographie depuis sa première apparition jusqu'à nos jours. Un état de l'art aussi riche que possible sur les plus célèbres algorithmes de chiffrement, ainsi les fameuses ruses des cryptanalystes ont été exposées. D'après cette étude, un système cryptographique est considéré comme sûr si personne n'a encore mis en défaut sa sécurité. Nous avons vu que ni la cryptographie classique ni symétrique n'a pu s'assurer ce besoin dû aux leurs inconvénients. Pour résoudre cela, les cryptographes ont cherché à déplacer la difficulté ; plutôt que d'utiliser de simples substitutions et de faire reposer la sécurité sur le nombre de clés possibles, ils ont essayé de faire reposer la sécurité sur des difficultés calculatoires. Cela a donné naissance aux algorithmes de cryptographie asymétrique. Cependant il s'avère que l'augmentation constante de la puissance de calcul de nos machines nécessite d'augmenter constamment la difficulté de nos algorithmes. Il se peut en effet qu'une nouvelle technologie anéantisse toute la difficulté d'un algorithme. La cryptographie hybride est aussi un sujet d'attaque puisqu'elle n'est qu'une combinaison de la cryptographie symétrique et asymétrique. Une nouvelle tendance basée sur la cryptographie quantique a aussi été développée. Cette cryptographie est sûre, néanmoins elle est basée sur des principes beaucoup plus théorique et lourds.

Enfin, à partir de la richesse et de la variété des modèles mathématiques exploités dans le domaine de la cryptographie, de nouvelles approches cryptographiques peuvent être proposées par exploitation de méthodes approchées. Le précédent chapitre présentera de telles méthodes de résolution de problèmes : les métaheuristiques.

CHAPITRE II

METAHEURISTIQUES

II.1. Introduction

Un très grand nombre de méthodes existent en RO et en IA pour résoudre différentes sortes de problèmes tels que les problèmes d'optimisation combinatoire. D'une manière très générale, les méthodes de résolution suivent quatre approches différentes pour la recherche d'une solution [JinK, 1999] : l'approche de construction, l'approche de relaxation, l'approche de voisinage et l'approche d'évolution. Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative : par exemple, la méthode de Lin et Kernighan qui resta longtemps le champion des algorithmes pour le problème du voyageur de commerce [LinS, 1973].

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques* [Reev, 1993], [Aart, 1997]. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [Lapo, 1996], [Osma, 1996]. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi, les métaheuristiques sont conçues pour résoudre des problèmes d'optimisation complexes où d'autres méthodes d'optimisation ne sont pas efficaces. Ces méthodes ont fini par être reconnues comme l'une des approches les plus pratiques pour résoudre des problèmes nombreux et complexes, et ceci est particulièrement vrai pour les divers problèmes du monde réel qui sont de nature combinatoire. L'avantage pratique de métaheuristiques réside dans leur efficacité et leur application générale. Dans la littérature et au début des recherches, des heuristiques spécialisées ont été généralement développées pour résoudre des problèmes complexes d'optimisation combinatoire.
D'autre part, avec l'émergence de stratégies des solutions plus générales, y compris les métaheuristiques telles que la recherche taboue, algorithmes génétiques, recuit simulé, le principal défi est devenu l'adaptation des méta-heuristiques à un problème particulier ou une catégorie de problème. Cela nécessite généralement beaucoup moins de travail que de développer une heuristique spécialisée pour une application spécifique, ce qui rend les métaheuristiques un choix attrayant pour la mise en œuvre dans les logiciels à usage général. En outre, une bonne mise en œuvre de métaheuristique est susceptible de fournir des solutions quasi optimales en temps de calcul raisonnables.

Ainsi, une métaheuristique est définie de manière similaire à une heuristique (qui est une méthode conçue pour un problème d'optimisation donné et qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème), mais à un niveau d'abstraction plus élevé (d'après E. Taillard).

II.2. Principes

Les métaheuristiques sont apparues dans les années 80. Ce sont des méthodes d'optimisation, de type stochastique, conçus pour les problèmes difficiles. Des progrès importants ont été réalisés avec l'apparition de nouvelles générations de méthodes approchées puissantes et générales, souvent appelées métaheuristiques [Reev, 1993] [Aart, 1997].

Le terme « métaheuristique » a été initialement utilisé par F. Glover pour distinguer la méthode tabou des heuristiques spécifiques [Glov, 1986]. Notons que ce terme est également utilisé par J-L. Laurière dans son système de résolution Alice [Laur, 1978].

L'origine du mot méta heuristique nous aide à comprendre sa signification. En grec: « Méta » signifie « au-delà », ou « au niveau supérieur », et « Heuristique» veut dire « Trouver ». Ainsi, l'interprétation de ces mots peut signifier que l'on effectue des recherches à un très haut niveau et que la procédure algorithmique regroupe plusieurs heuristiques.

D'après Glover, metaheuristique "se réfère à une stratégie maître qui guide et modifie d'autres heuristiques pour produire des solutions au-delà de ceux qui sont normalement générés dans une quête d'optimalité locale" [Glov, 1997].

Selon [Glov, 1997] « métaheuristiques dans leurs formes modernes sont basées sur une variété d'interprétations de ce qui constitue une recherche intelligente », où le terme de recherche intelligente a été mis en évidence par Pearl [Poire, 1984] (en ce qui concerne heuristiques dans un contexte de l'intelligence artificielle) et [vobs, 1993] (en ce qui concerne le contexte de la recherche opérationnelle). Dans ce sens, on peut aussi considérer la définition suivante: « Une métaheuristique est un processus à générations itératives qui guide

une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter les espaces de recherche en utilisant des stratégies lesarning pour structurer l'information afin de trouver des solutions quasi-optimales » [Osma, 1996]

Pour résumer, la définition suivante semble être la plus appropriée : « Une métaheuristique est un processus maître itératif qui guide et modifie les opérations d'heuristiques subordonnées pour produire efficacement des solutions de haute qualité. Il peut manipuler une solution unique complète (ou incomplète) ou un ensemble de solutions à chaque itération. Les heuristiques subordonnées peuvent être des procédures de niveau élevé (ou faible), ou une recherche locale simple, ou tout simplement une méthode de construction. La famille de métaheuristiques comprend, mais n'est pas limitée à, des procédures de mémoire d'adaptation, de recherche tabou, des systèmes de fourmis, avides de recherche randomisée d'adaptation, de recherche à voisinage variable, des méthodes évolutionnaires, des algorithmes génétiques, de recherche de points, des réseaux neuronaux, de recuit simulé, et leurs hybrides. " [VobS, 1999]

Le but des métaheuristiques est de minimiser ou de maximiser une, ou plusieurs fonctions objectives. Comme exemple, on peut demander de trouver le plus court chemin entre un point A et un point B, sachant que des obstacles sont présents donc il s'agit d'un problème de minimisation ; ou si on veut trouver comment effectuer le plus grand nombre de tâches en un temps limité alors c'est la maximisation du travail à faire.

L'évolution des métaheuristiques se fait de manière itérative. Ceci a l'avantage de permettre l'arrêt de l'algorithme quand on le souhaite, et de récupérer la meilleure solution trouvée jusqu'à présent sans être obligé d'attendre la fin de l'exécution. Un autre point important des métaheuristiques est qu'elles font évoluer des solutions en les améliorant à chaque itération.

Donc, les métaheuristiques sont généralement des algorithmes itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction. Les itérations successives permettent d'évoluer d'une solution peu satisfaisante à la solution la plus adaptée. Les métaheuristiques se comportent, donc, comme des algorithmes de recherche tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution.

Les métaheuristiques sont souvent inspirées de systèmes naturels, qu'ils soient pris en physique -cas du recuit simulé-, en biologie de l'évolution -cas des algorithmes génétiques- ou encore en éthologie -cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires.
Les métaheuristiques désignées par M sont, souvent, des algorithmes utilisant un échantillonnage probabiliste. Elles tentent de trouver l'optimum global (G) d'un problème d'optimisation difficile (avec des discontinuités (D), par exemple), sans être piégées par les optima locaux (L). Ceci est indiqué à la figure suivante :

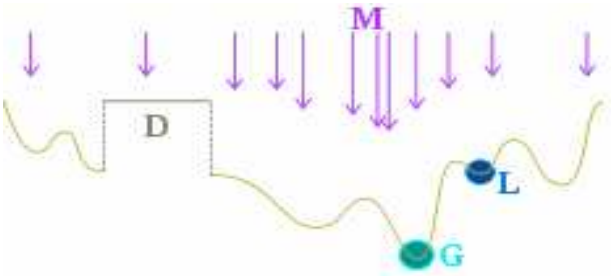


Figure II.1. *Principe général des métaheuristiques.*

II.3. Organisation d'une métaheuristique

D'une manière générale, les métaheuristiques s'articulent autour des principales notions suivantes :

II.3.1. Le Voisinage

Le voisinage d'une solution est un sous-ensemble de solutions qu'il est possible d'atteindre par une série de transformations données.
Formellement parlant [JinK, 1999], soit X l'ensemble des configurations admissibles d'un problème, on appelle *voisinage* toute application $N : X \to 2^X$. On appelle *mécanisme d'exploration* du voisinage toute procédure qui précise comment la recherche passe d'une configuration $s \in X$ à une configuration $s' \in N(s)$. Une configuration *s* est un *optimum local* par rapport au voisinage *N* si $f(s) \leq f(s')$ pour toute configuration $s' \in N(s)$.

Une méthode typique de voisinage débute avec une configuration initiale, et réalise ensuite un processus itératif qui consiste à remplacer la configuration courante par l'un de ses voisins en tenant compte de la fonction de coût. Ce processus s'arrête et retourne la meilleure configuration trouvée quand la condition d'arrêt est réalisée.

Un des avantages de cette stratégie de recherche réside précisément dans la possibilité de contrôler le temps de calcul : la qualité de la solution trouvée tend à s'améliorer progressivement au cours du temps et l'utilisateur est libre d'arrêter l'exécution au moment qu'il aura choisi. Dans la littérature, plusieurs voisinages ainsi que plusieurs stratégies de parcours de ces voisinages ont été définies [Gold, 1989], [Hert, 2003].

II.3.2. Diversification, intensification et apprentissage

La *diversification* (ou *exploration*, synonyme utilisé presque indifféremment dans la littérature des algorithmes évolutionnaires) désigne les processus visant à récolter de l'information sur le problème optimisé. L'*intensification* (ou *exploitation*) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Autrement dit, l'intensification insiste sur la capacité d'examiner en profondeur par une méthode des zones de recherche particulières alors que la diversification met en avant la capacité de découvrir des zones de recherche prometteuses.

La *mémoire* est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi les optima locaux.

Les métaheuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage, ou en mêlant ces notions de façon plus

étroites. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

Les notions d'intensification et de diversifications sont prépondérantes dans la conception des métaheuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherches. En effet, l'application systématique du seul principe d'exploitation ne permet pas une recherche efficace. En effet, l'exploitation conduit à confiner la recherche dans une zone limitée qui finit par s'épuiser. Le cas de l'amélioration itérative rapidement piégée dans un optimum local illustre cruellement ce phénomène. Une autre illustration souvent évoquée est fournie par le problème de convergence prématurée des algorithmes génétiques : du fait de la sélection, la population finit par n'être constituée que d'individus tous similaires. L'une des préoccupations majeures dans les algorithmes génétiques consiste d'ailleurs à préserver le plus longtemps possible une diversité suffisante dans la population. Face à ce type de difficulté, la solution consiste à diriger la poursuite de la recherche vers de nouvelles zones, *i.e.*, à recourir à l'exploration. Les deux notions ne sont, donc, pas contradictoires mais complémentaires, et il existe de nombreuses stratégies mêlant à la fois l'un et l'autre des aspects.

II.4. Méthodes générales et méthodes spécifiques

Contrairement aux algorithmes traditionnels dédiés à un problème spécifique, les métaheuristiques constituent des mécanismes très généraux qui peuvent être adaptés pour traiter de nombreux problèmes différents. Pour expliquer l'efficacité d'un algorithme spécifique, on invoque souvent le fait qu'il utilise des connaissances spécifiques du problème. Pour expliquer l'efficacité d'une métaheuristique, deux types d'arguments opposés sont généralement avancés.

Selon certains, des mécanismes généraux suffisamment puissants ont par eux mêmes la faculté de mener efficacement la recherche sans disposer d'information spécifique du problème considéré (contexte de boîte noire) : c'était notamment le point de vue classique porté sur les algorithmes génétiques [Gold, 1989]. Malheureusement, de même qu'il ne peut pas exister de stratégie avantageuse dans un jeu de hasard comme la roulette, il existe également des limitations théoriques fondamentales qui ruinent les espoirs d'une méthode aveugle dans le cas le plus général. En fait, le théorème « *No Free Lunch (NFL)* » [Wolp, 1997] montre que pour les problèmes de type boîte noire, toutes les méthodes sont équivalentes et font aussi bien, ou plutôt aussi mal, que l'énumération aléatoire.

Selon le point de vue opposé, la puissance d'une métaheuristique est d'abord liée à son aptitude à intégrer des connaissances spécifiques du problème. La connaissance du problème la plus fondamentale réside dans le codage du problème et dans le choix de la fonction de voisinage. Plusieurs auteurs insistent sur l'importance du codage du problème, voir par exemple [Radc, 1995]. Dans le cas du voyageur de commerce, plusieurs types de codages différents ont été proposés et conduisent à des performances très variées [Mich, 1992]. En général, il n'existe pas de codage universellement efficace. Un « bon » codage doit permettre de restreindre l'espace de recherche et d'intégrer des connaissances du problème.

Les métaheuristiques tentent également d'améliorer leur efficacité en incorporant des connaissances supplémentaires dans leurs opérateurs. Plus les opérateurs d'une méthode utilisent des connaissances spécifiques, plus la méthode dispose de moyens potentiels pour conduire efficacement la recherche. En contrepartie, l'intégration de ces connaissances spécifiques (en supposant que ces connaissances soient disponibles) nécessite un effort pour spécialiser ou adapter la méthode. La méthode tabou vise à incorporer le plus possible de connaissances du problème pour atteindre le maximum d'efficacité : l'utilisateur doit notamment définir de façon pertinente le type de caractéristique figurant dans la liste tabou.

De leur coté, les algorithmes génétiques se sont éloignés du modèle standard pour intégrer également des connaissances du problème : codage non binaire, opérateurs spécifiques. Au contraire, le recuit simulé est parfois présenté comme une méthode facile à adapter à un problème et qui ne tente pas d'exploiter de connaissances spécifiques.

En général, une méthode offrant des possibilités d'intégrer des connaissances du problème a plus de chance de produire de bons résultats, mais demande un effort d'adaptation et de spécialisation. Au contraire, une méthode très générale qui prétend n'intégrer aucune connaissance propre ne peut pas être compétitive.

II.5. Classification des métaheuristiques

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents.
Dans la littérature, plusieurs classifications de métaheuristiques ont été proposées. Nous présentons à ce niveau un aperçu des principales classifications.

II.5.1. Les métaheuristiques inspirées et non inspirées d'un phénomène naturel

Une manière intuitive de classifier les métaheuristiques consiste à séparer celles qui sont inspirées d'un phénomène naturel, de celles qui ne le sont pas.

Les algorithmes génétiques ou les algorithmes par colonies de fourmi entrent clairement dans la première catégorie, tandis que la méthode de descente, ou la recherche Tabou, vont dans la seconde.

II.5.2 Les métaheuristiques avec fonction objective statique ou dynamique

Les métaheuristiques peuvent être aussi classées selon la façon dont ils utilisent la fonction objective. Certains algorithmes conservent la fonction objective donnée dans la représentation de problème telle qu'elle est (comme la recherche locale guidée (GLS)) et la modifie lors de la recherche. L'idée de cette approche est d'éviter les minimums locaux en modifiant l'espace de recherche. En conséquence, lors de la recherche la fonction objective est altérée en essayant d'intégrer des informations intégrées au cours du processus de recherche.

II.5.3. Les métaheuristiques avec une ou plusieurs structures de voisinage

La plupart des métaheuristiques travaillent sur une seule structure de voisinage. En d'autres termes, la topologie du paysage de fitness ne change pas en cours de l'algorithme. D'autres métaheuristiques telles que la recherche par voisinage variable utilisent un ensemble de structures de voisinage qui donne la possibilité de diversifier la recherche en échangeant entre les structures de voisinage qui ont des formes différentes.

II.5.4. Les métaheuristiques avec et sans mémoire

Les métaheuristiques utilisent l'historique de leur recherche pour guider l'optimisation aux itérations suivantes. Dans le cas le plus simple, elles se limitent à considérer l'état de la recherche à une itération donnée pour déterminer la prochaine itération : il s'agit alors d'un processus de décision markovien, et on parlera de méthode *sans mémoire*. C'est le cas de la plupart des méthodes de recherche locale (recherche à voisinage variable, recherche locale

itérée, recherche locale stochastique, recherche locale guidée), algorithme d'estimation de distribution, recuit simulé, GRASP.

Beaucoup de métaheuristiques utilisent une mémoire plus évoluée, que ce soit sur le court terme (solutions visitées récemment, par exemple) ou sur le long terme (mémorisation d'un ensemble de paramètres synthétiques décrivant la recherche).

II.5.5. Les métaheuristiques implicite, explicite, directe

En considérant les métaheuristiques comme des méthodes itératives utilisant un échantillonnage de la fonction objective comme base d'apprentissage (définition plus particulièrement adaptée aux métaheuristiques à populations) apparaît le problème du choix de l'échantillonnage.

Dans la très grande majorité des cas, cet échantillonnage se fait sur une base aléatoire, et peut donc être décrit via une distribution de probabilités. Il existe alors trois classes de métaheuristiques, selon l'approche utilisée pour manipuler cette distribution.

La première classe est celle des méthodes *implicites*, où la distribution de probabilité n'est pas connue *a priori*. C'est le cas par exemple des algorithmes génétiques, où le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. L'évolution différentielle, Scatter search, algorithmes à essaim de particules, stratégies d'évolution et algorithmes de colonie de fourmis sont des méthodes implicites.

Par opposition, on peut donc classer les méthodes *explicites*, qui utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution.

Dans cette classification, le recuit simulé occupe une place particulière, puisqu'on peut considérer qu'il échantillonne la fonction objective en utilisant directement celle-ci comme distribution de probabilité (les meilleures solutions ayant une probabilité plus grande d'être tirées). Il n'est donc ni explicite ni implicite, mais plutôt « direct » (exemple : le recuit simulé) [JinK, 1999].

II.5.6. Les métaheuristiques évolutionnaires et non évolutionnaires

On trouve parfois une classification présentant les algorithmes d'optimisations stochastiques comme étant « évolutionnaires » (ou « évolutionnistes ») ou non. L'algorithme sera considéré comme faisant partie de la classe des algorithmes évolutionnaires s'il manipule une population *via* des *opérateurs*, selon un algorithme général donné.

Cette façon de présenter les métaheuristiques dispose d'une nomenclature adaptée : on parlera d'opérateurs pour toute action modifiant l'état d'une ou plusieurs solutions. Un opérateur construisant une nouvelle solution sera dénommé *générateur*, alors qu'un opérateur modifiant une solution existante sera appelé *mutateur*.

Dans cette optique, la structure générale des algorithmes évolutionnaires enchaîne des étapes de *sélection*, de *reproduction* (ou *croisement*), de *mutation* et enfin de *remplacement*. Chaque étape utilise des opérateurs plus ou moins spécifiques.

Quelques algorithmes évolutionnaires : l'algorithme génétique, la programmation génétique, l'algorithme d'évolution différentielle, les stratégies évolutionnaires et l'algorithme d'estimation de distribution [JinK, 1999].

II.5.7. Les métaheuristiques à base de population et les métaheuristiques à trajectoire

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

Les plus connues dans cette classe sont le recuit simulé, la recherche tabou, la recherche à voisinage variable, la méthode GRASP. L'autre approche utilise la notion de population. La métaheuristique manipule un ensemble de solutions en parallèle, à chaque itération. On peut citer les algorithmes génétiques, l'optimisation par essaims particulaires et les algorithmes de colonies de fourmis.

La frontière est parfois floue entre ces deux classes. On peut ainsi considérer qu'un recuit simulé où la température baisse par paliers, a une structure à population. En effet, dans ce cas on manipule un ensemble de points à chaque palier, il s'agit simplement d'une méthode d'échantillonnage particulière.

II.5.7.1. Les métaheuristiques à trajectoire (à solution unique)

Ici, nous résumons les plus connues des métaheuristiques à trajectoire.

a. La méthode de descente

Le principe de la méthode de descente (dite aussi *basic local search*) consiste, à partir d'une solution *S*, à choisir une solution *S'* dans un voisinage de *S* telle que *S' améliore* la recherche (généralement telle que $f(S') < f(S)$). On peut décider soit d'examiner toutes les solutions du voisinage *N* et prendre la meilleure de toutes ces solutions (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.
La méthode de descente est la méthode la plus élémentaire de recherche locale. On peut la formaliser comme suit :

Algorithme II.1 *descente_simple* (solution initiale *S*)

Début
  Répéter
    Choisir *S'* dans N(*S*)
    Si $f(S') < f(S)$ alors $S \leftarrow S'$
  Jusqu'à ce que $f(S') \geq f(S)$, $\forall S' \in$ **S**
Fin

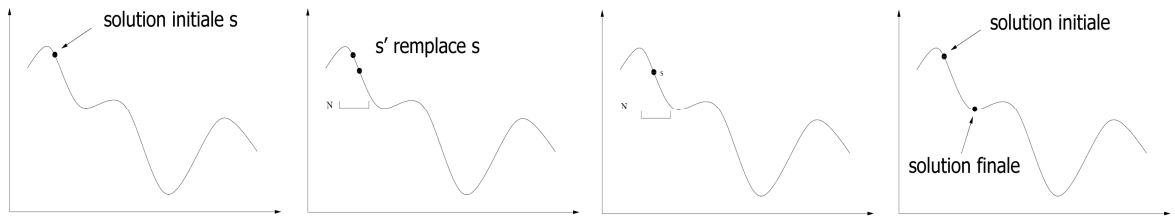


Figure II.2. *Evolution d'une solution dans la méthode de descente.*

On peut varier cette méthode en choisissant à chaque fois la solution S' dans N(S) qui améliore le plus la valeur de f. C'est la méthode de plus grande descente [Cost, 2006].

b. Recherche aléatoire (Random Search : RS)

C'est la plus simple des méthodes stochastiques. Cette méthode consiste à tirer à chaque itération une solution au hasard. La fonction objective *f* est évaluée en ce point. La nouvelle valeur est comparée à la précédente. Si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continu. Sinon on repart du point précédent et on recommence le procédé, jusqu'à ce que les conditions d'arrêt soient atteintes. L'algorithme II.2 [Luke, 2010] présente un pseudo code de la recherche aléatoire dans le cas d'un problème de minimisation.

Algorithme II.2 *Recherche aléatoire*

Début
$S_0 \leftarrow$ solution aléatoire;
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$;
Répéter
S ← solution aléatoire ;
Si ($f(S) < f_{min}$) *Alors*
$f_{min} \leftarrow f(S)$;
$x_{min} \leftarrow S$;
Fin Si
Jusqu'à conditions d'arrêt satisfaites.
Fin

c. Hill Climbing (HC)

On part d'une solution si possible "bonne" (par exemple donnée par une heuristique) et on balaie l'ensemble des voisins de cette solution ; s'il n'existe pas de voisin meilleur que notre solution, cela veut dire qu'un optimum local a été trouvé et on arrête la recherche. Sinon, on choisit le meilleur des voisins et on recommence. Une autre implémentation consiste non pas à passer au meilleur des voisins à chaque étape, mais au premier meilleur voisin trouvé. La convergence vers un optimum local pouvant être très lente, on peut éventuellement fixer un nombre de boucles maximum, si on veut limiter le temps d'exécution [Sea, 2010].

Cette méthode a l'inconvénient de rester bloquée dans un optimum local : une fois un optimum local trouvé, on s'arrête, même si ce n'est pas l'optimum global. Selon le paysage des solutions, l'optimum local peut être très bon ou très mauvais par rapport à l'optimum global. Si la solution de départ est donnée par une heuristique déterministe, l'algorithme sera déterministe. Si elle est tirée au hasard, l'algorithme devient non déterministe et donc plusieurs exécutions différentes sur la même instance pourront donner des solutions différentes et de qualités différentes [Sea, 2010].

La notion de voisinage est primordiale. Si les voisins sont très nombreux, on a de fortes chances de trouver l'optimum global, mais visiter un voisinage peut être très long: on visitera une grande partie de l'espace des solutions. Si le voisinage est très restreint, on risque fort de rester bloqué dans un optimum local de "mauvaise qualité". Le choix de la notion de voisinage est un compromis entre efficacité et qualité.

Le pseudo-code de l'algorithme Hill Climbing est celui présenté par l'algorithme II.3 [Sea, 2010]:

Algorithme II.3 *HILL CLIMBING*

Début
$S_0 \leftarrow$ Solution aléatoire
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$
Répéter
Engendrer un N-échantillon $S_i(x)$ voisinage de S(x) *ET* calculer
$f(S(x)) = \min[f(S_i(x))]$ avec $1 \leq i \leq N$

Si ($f(S) < f_{min}$) *then*
$x_{min} \leftarrow S$
Sinon
Sortir de Répéter
Fin Si
Jusqu'à condition d'arrêt satisfaite.
Fin.

d. Recuit Simulé

Le recuit simulé a été introduit et mis au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985. Il a été repris sous différentes formes par Lawrence Davis en 1987 [Lawr, 1987].
Cet algorithme a été dérivé de l'algorithme de Metropolis, développé par les scientifiques du projet ex-Manhattan : Nicholas Metropolis, Arianna et Marshall Rosenbluth, Augusta et Edward Teller en 1953. Ce dernier permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. Partant d'une solution donnée, en la modifiant, on en obtient une seconde. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

La description des phénomènes physiques et quantiques liés au processus de recuit s'appuie sur la statistique de Boltzmann. Pour qu'un métal ait une qualité optimale, il faut que son état d'énergie soit minimal. Pour améliorer la qualité d'un métal, on le chauffe, puis on le refroidit par paliers en attendant à chaque fois que l'état d'énergie soit stabilisé. Au niveau de l'atome, cela signifie qu'à une température chaude, les atomes bougent beaucoup, et en se

refroidissant, ils trouvent leur place optimale. Plus la température descend, moins les atomes bougent et le métal devient de plus en plus résistant.

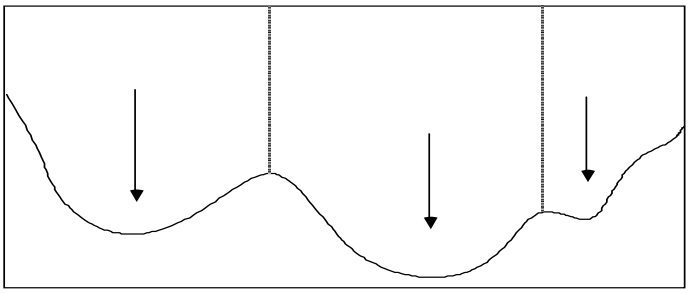


Figure II.3. *Un paysage d'énergie. Suivant la configuration initiale, indiquée par les flèches, la dynamique aboutit à température nulle dans l'un quelconque des minima relatifs, séparés par les barrières indiquées en pointillés. A température élevée, les processus probabilistes permettent au système de sauter les barrières séparant les vallées.*

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. À cette solution correspond une énergie initiale $E = E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T = T_0$ élevée est également choisie. Ce choix est alors totalement arbitraire et va dépendre de la loi de décroissance utilisée.

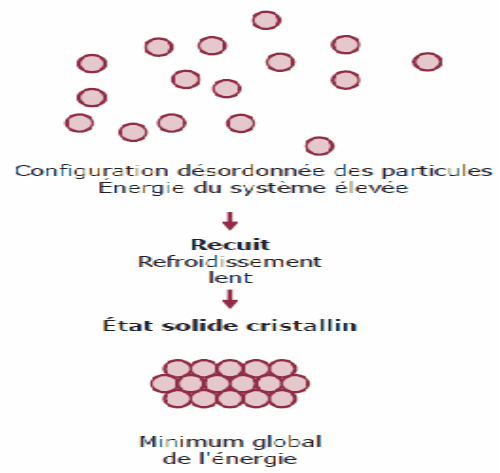


Figure II.4. *Transposition de procédé recuit à la résolution d'un problème d'optimisation.*

À chaque itération de l'algorithme une modification élémentaire de la solution est effectuée. Cette modification entraîne une variation Δ_E de l'énergie du système (toujours calculée à partir du critère que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta_E}{T}}$. Ce choix de l'exponentielle pour la probabilité s'appelle règle de Metropolis.

On itère ensuite selon ce procédé en gardant la température constante.

Algorithme II.4 *Recuit simulé*

```
Début
    Initialisation du temps
    Tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint] Faire
        Initialisation de la température à Tmax
        Tant que [la température est supérieure à 0] et [le temps limite n'est pas atteint] Faire
            Incrémentation du temps
            Modification de la solution courante
            Si [la nouvelle solution est meilleure que l'ancienne] Alors
                la sauvegarder
            Sinon
                Si [le système est stable depuis un nombre défini d'itérations] Alors
                    Diminuer la température
                Fin si
            Fin si
        Fin tant que
    Fin tant que
Fin
```

L'algorithme varie de Hill-Climbing (algorithme 3) dans sa décision du moment de remplacer S, la solution candidate d'origine, avec R, son enfant nouveau peaufiné. Plus précisément, si R est meilleur que S, nous allons toujours remplacer S par R comme d'habitude. Mais si R est pire que S, on peut toujours remplacer S par R avec une certaine probabilité P(t, R, S):

$$P(t, R, S) = e^{\frac{\text{Quality}(R) - \text{Quality}(S)}{t}}$$

Où t> 0. Cette équation est intéressante à deux égards. Notez que la fraction est négative car R est pire que S. Premièrement, si R est bien pire que S, la fraction est plus grande, et donc la probabilité est proche de 0. Si R est très proche de S, la probabilité est proche de 1. Ainsi, si R n'est pas bien pire que S, nous allons toujours choisir R avec une probabilité raisonnable.

Deuxièmement, nous avons un paramètre ajustable t. Si t est proche de 0, la fraction est de nouveau un grand nombre, et donc la probabilité est proche de 0. Si t est élevé, la probabilité est proche de 1. L'idée est d'abord de mettre t comme un grand nombre, ce qui provoque l'algorithme de se déplacer à chaque solution nouvellement créée indépendamment de sa qualité. Nous faisons une marche aléatoire dans l'espace. Puis t diminue lentement, éventuellement à 0, à quel point l'algorithme ne fait rien de plus que de simples Hill-Climbing.

e. Recherche Tabou (Tabu Search : TS)

La recherche avec tabou, ou simplement dite recherche tabou, est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans un article paru en 1986 [Glov, 1986], mais reprenant de nombreuses idées proposées antérieurement dès les années 60 dans les deux articles simplement intitulés « Tabu chearch » [Glov, 1989] proposant la plus part des principes de cette recherche telle qu'elle est décrite actuellement. Maintenant, elle est d'usage très classique en domaine d'optimisation combinatoire pour résoudre les problèmes NP-durs.

L'idée principale de la recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui optimise la fonction objective. Le voisinage choisi est exploré de manière déterministe, il est interdit de reprendre des solutions récemment visitées.

L'un des principes fondamentaux de cette recherche qui la caractérise des autres méta-heuristiques est la construction d'un historique de la recherche itérative ou, ce qui est équivalent, au fait de doter la recherche de mémoire [Glov, 1997]. La recherche tabou examine un échantillonnage de solutions *N(S)* et retient la meilleure solution *S* des solutions obtenues.

A chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche tabou qu'elle est une méthode agressive [www10].

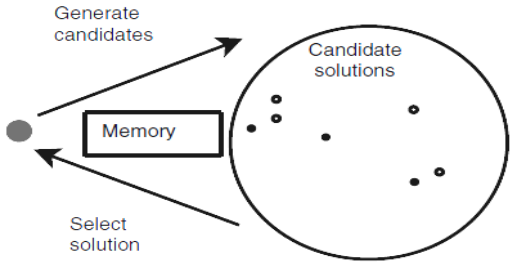
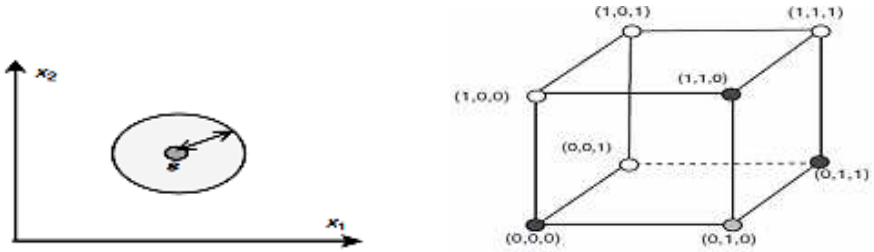


Figure II.5. *Principe de base d'une méta-heuristique à mémoire.*

La recherche tabou est essentiellement axée sur une exploration non triviale de l'ensemble des solutions en utilisant la notion de voisinage. Formellement pour toute solution *s* de *S*, un ensemble de $N(s) \in S$ qu'on appelle ensemble des solutions voisines de *s*.



(a) *(b)*

Figure II.6. *Les types de solutions du voisinage.*
(a) *Le voisinage pour un problème à variables continus : Le cercle symbolise les voisins de la solution s,*
(b) *Le voisinage pour un problème à variables discrètes : Les nœuds du cube sont les solutions et leurs voisins adjacents.*

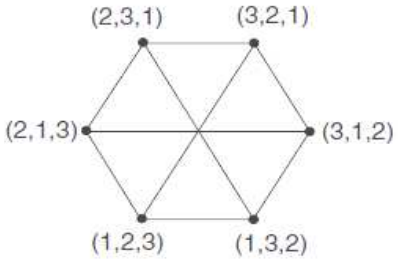


Figure II.7. *Voisinage d'une permutation de taille égale à 3.*
Les voisins de la solution (2.3.1) sont : (3.2.1), (2.1.3), et (1.3.2).

Vue d'un côté pratique l'ensemble N(s) des solutions voisines de s n'est autre que l'ensemble des modifications que l'on peut apporter à cette dernière. On appelle *mouvement* une modification apportée à une solution.

Dans la littérature et dans le cas de la recherche tabou, l'application d'un mouvement *m* à une solution *s* est notée ; d'où l'expression suivante du voisinage :

$$N(s) = \{s' / s' = s \oplus m, m \in M\}$$

La figure II.8 [Kamm, 2006] illustre l'ensemble des mouvements possibles dans le cas d'une solution composée de quatre éléments.

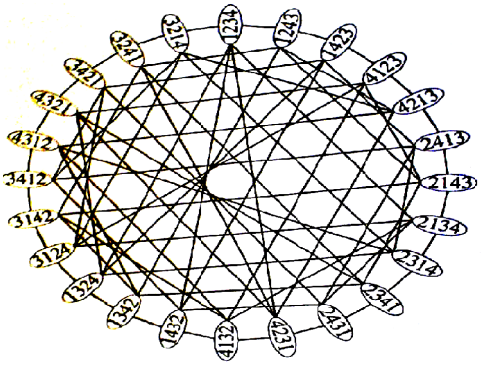


Figure II.8. *Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments.*

La recherche tabou est une méthode itérative qui explore un espace de solutions. Comme toute méthode itérative, il est intéressant d'éviter de revisiter des solutions pour essayer de converger vers la solution optimale le plus rapidement possible et ceci n'est possible qu'à l'aide d'une mémoire caractérisée par une taille définie par l'utilisateur et le problème à résoudre. Cependant, l'utilisation d'une mémoire peut s'avérer dans bien des cas peu efficace, voire mauvaise. Outre le fait de mettre cette idée en pratique est difficile, cela suppose que l'on doit mémoriser chaque solution visitée et de tester à chaque itération et pour chaque solution admissible si cette dernière a été déjà examinée. Les *tables de hachage* permettent de faire cela de manière efficace, mais ceci n'empêche pas la croissance linéaire de la taille de la mémoire avec le nombre des itérations effectuées.

Une table de hachage est définie par un tableau *T* de *J* entiers, avec *J* choisi relativement grand et assez voisin de la capacité de la machine utilisée tout en prenant en considération la taille du problème.

Le but de la table de hachage est :

- ✓ d'interdire le retour aux solutions obtenues pendant les *t* dernières opérations ;
- ✓ ainsi éliminer les cycles de longueurs égales ou inférieures à *t*.

Toutefois, le fait d'interdire un grand nombre de mouvements aura pour conséquence de rendre l'obtention de bons résultats très délicate faute de mouvements disponibles. Si ce nombre diminue, cela augmentera les chances d'une meilleure exploration aux alentours des optimums locaux et d'obtenir ainsi de meilleures solutions. Il ne faut cependant pas trop diminuer ce nombre, car, dans ce cas, il devient très probable de se trouver prisonnier d'un ensemble très restreint de solutions et de les visiter tout le temps de la recherche.

Pour palier à ce compromis du choix de nombre des mouvements à interdire et de bénéficier simultanément des avantages d'un petit nombre, qui permet une visite détaillée du voisinage d'une même solution, et d'un grand nombre qui permet de franchir le voisinage de différentes solutions, ce nombre doit varier au cours du processus itératif. Pour ce faire, plusieurs méthodes existent. Ce nombre peut être tiré au hasard à partir d'un intervalle donné à chaque itération ou après un nombre d'itérations, comme il peut aussi croître ou décroître en fonction des résultats obtenus au cours de la recherche [Tail, 1991], [Tail, 1995] et [Tail, 1999].

Autrement dit, une liste tabou avec trop d'éléments peut devenir très restrictive. En effet, il a été observé que trop de contraintes tabou forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération. Par contre une liste tabou contenant trop peu d'éléments peu s'avérer inutile et mener à des mouvements cycliques [Ayas, 2004].

En plus, dans beaucoup de problèmes, l'interdiction de revisiter des solutions mènerait à des incohérences comme la déconnection de la solution courante de la solution optimale ou bien le blocage de la recherche itérative par cause d'absence de solutions voisines non visitées[Kamm, 2006].

La figure suivante illustre ces derniers cas incohérents :

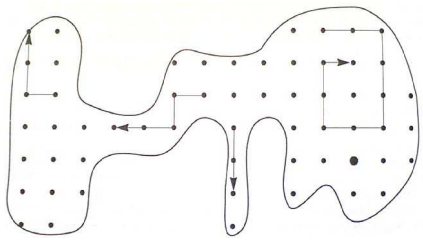


Figure II.9. *Déconnections et blocages dans la recherche tabou.*

Il est à signaler qu'une liste tabou peut être soit statique ou dynamique. Pour la première, la taille de la liste est fixée, en générale au début. Elle dépend de la taille du problème à résoudre et du voisinage choisi. Pour la deuxième, la taille doit changer durant le processus de recherche sans utiliser aucune information sur la mémoire de recherche.

Algorithme II.5 *Recherche Tabou*

Début

Φ Fonction de coût
Variables locales : S solution courante, Liste Taboue L, Meilleure solution M, itération courante K, nombre d'itérations N
Paramétrages : taille de liste taboue, critère d'aspiration
Choix d'une solution initiale S_0
S← S_0
M← S
K ←0
Tant que K< N *faire*
K← K+1
Mise à jour de L
Génération des candidats E par opération de voisinage
C ← Best (E)
Si Φ (S) < Φ(M) *ou* C n'est pas taboue *ou* C vérifie l'aspiration *alors*
S ← C
Sinon
E ← E \ C
Fin tant que
Retourner S

Fin.

f. GRASP

À proprement parler, GRASP (pour *Greedy Randomized Adaptive Search Procedure*) est une méthode hybride, car elle cherche à combiner les avantages des heuristiques gloutonnes, de la recherche aléatoire et des méthodes de voisinage. Un algorithme GRASP répète un processus composé de deux étapes : la construction d'une solution suivie par une descente pour améliorer la solution construite. Durant l'étape de construction, une solution est itérativement construite : chaque itération ajoute un élément dans la solution partielle courante. Pour déterminer l'élément qui sera ajouté, on utilise une liste des meilleurs candidats obtenus avec une fonction gloutonne et on prend *au hasard* un élément dans cette liste. La liste des meilleurs candidats est dynamiquement mise à jour après chaque itération de construction. Cette étape de construction continue jusqu'à ce qu'une solution complète soit obtenue [JinK, 1999].

À partir de cette solution, une descente est appliquée pour améliorer la solution. Une procédure GRASP répète ces deux étapes et retourne à la fin la meilleure solution trouvée. Les deux paramètres de cette méthode sont donc la longueur de la liste de candidats et le nombre d'itérations autorisées.

Algorithme II.6 *GRASP*

```
Début
    x, x*, f* : type Données
    Début
    f* ← infini
    Pour i = 1 à nombre d'itérations faire
        x ← Construction Aléatoire Gloutonne ()
        x ← Recherche Locale(x)
        Si f(x) < f* alors
            x* ← x
            f* ←f(x)
        Fin Si
  Fin Pour
Fin.
```

II.5.7.2. Les métaheuristiques à population

Dans cette section, nous présentons un sommaire des plus connues des métaheuristiques à population.

a. Les algorithmes évolutionnaires

Les *algorithmes évolutionnistes* ou *algorithmes évolutionnaires* (*evolutionary algorithms* ou encore dits *evolutionary computation* en anglais), sont une famille d'algorithmes s'inspirant de la théorie de l'évolution pour résoudre des problèmes divers. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation, elles sont en cela des métaheuristiques, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. On les classe également parmi les méthodes d'intelligence calculatoire.

Selon la génétique et la théorie de l'évolution [www11] :

- ✓ Un enfant hérite son patrimoine génétique pour moitié de sa mère et pour moitié de son père (reproduction sexuée).
- ✓ Les enfants ne sont pas identiques aux parents car des altérations des gènes peuvent se produire (mutations).
- ✓ Parmi les mutations, certaines peuvent être favorables et d'autres défavorables.
- ✓ Il naît beaucoup de descendants : mais seuls les individus les mieux adaptés pourront survivre et transmettre leurs gènes à leur tour à leur descendance.

Dans ce modèle, on observe que [www11] :

- ✓ Le hasard joue un rôle moteur pour produire de nouveaux individus différents de leurs parents.
- ✓ La sélection naturelle effectue le tri entre les variations favorables et les autres.

Ainsi, basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin, ces algorithmes présentent des qualités intéressantes pour la résolution des problèmes d'optimisation. Les individus ou chromosomes d'un algorithme évolutionnaires sont des codages des solutions possibles du problème. Comme dans la nature, ces individus forment une population qui va évoluer dans le temps selon des lois de sélection qui vont favoriser les mieux adaptés à se croiser entre eux en produisant des populations meilleures. L'évolution des individus d'une population à une autre se fait à l'aide de la reproduction. Les individus parents vont se reproduire pour donner des individus enfants qui seront plus performants après avoir subis des opérations génétiques de croisement et de mutation.

Et comme ces reproductions se font avec une part de hasard par analogie à la nature, donc, les parents candidats à la reproduction sont choisis d'une manière probabiliste proportionnelle à leurs aptitudes et l'étape de reproduction est choisie d'une façon totalement aléatoire.

Finalement, passant d'une génération à une autre, les individus forment une progéniture plus performante qui s'approche au mieux de la solution optimale [Kamm, 2006].

La figure suivante résume le principe de résolution de problèmes par algorithmes évolutionnaires.

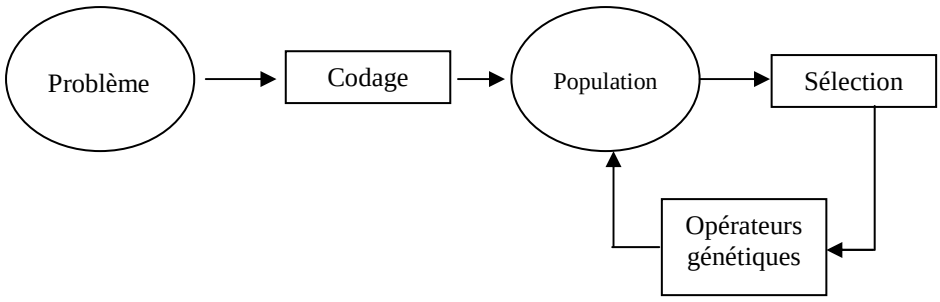


Figure II.10 . *Principe des algorithmes évolutionnaires.*

Ainsi, un algorithme évolutionnaire est construit autour des notions suivantes :

- **Le codage :**

La première étape de la résolution d'un problème à l'aide d'un algorithme génétique est la modélisation appropriée des solutions. Cette modélisation est appelée *codage* et permet de représenter les solutions sous forme de chromosomes.

Le codage se base sur deux notions importantes : le génotype et le phénotype.

- ✓ **Génotype** : représente l'ensemble des valeurs des gènes d'un chromosome.
- ✓ **Phénotype** : c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype.

S'il y a passage immédiat du phénotype au génotype, le codage est dit direct, sinon il est dit indirect et une procédure de passage est indispensable. Le codage d'une solution doit décrire toutes les données du problème et respecter toutes ses contraintes.

- **Population initiale :**

L'espace de recherche étant infini, il est très difficile de répartir plus ou moins équitablement la population initiale sur l'espace de recherche, de sorte que cet espace soit au maximum parcouru. Un certain nombre d'individus doivent être générés aléatoirement s'il n'existe aucune autre méthode d'initialisation.

- **Evaluation de la qualité d'une solution :**

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance (*fitness*) de mesurer cette efficacité pour permettre à l'algorithme de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution. Autrement dit, la fonction de performance *f*, doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre.

- **La sélection :**

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.
Plusieurs modes de sélection ont été définis tels que : sélection par roulette (wheel), sélection par rang, sélection steady-state, élitisme, sélection uniforme, etc.

- **Opérateurs génétiques :**

1) Croisement : C'est un opérateur génétique qui permet à deux chromosomes parents de produire deux chromosomes enfants où de nouvelles séquences de gènes pour les chromosomes enfants sont crées à partir d'une base de configuration des séquences héritées des chromosomes parents. Cet opérateur se produit selon une probabilité *Pc* fixée par l'utilisateur selon le problème à optimiser.

Il existe plusieurs manières d'effectuer un croisement soit par cross-over où l'on a besoin de deux parents qui génèrent à la fin du croisement deux enfants, soit par copie où l'on n'a besoin que d'un seul parent qui nous donnera à la fin du croisement un enfant qui est le parent lui-même (sa copie).

Dans la littérature, il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter [Mesg, 1999]. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que :

- ✓ le croisement à un point.
- ✓ le croisement multipoints.
- ✓ le croisement uniforme.

2) Mutation : permet de changer (permuter) un gène d'un chromosome par un autre d'une manière aléatoire, ce qui est à première vue très ressemblant avec un cross-over. La différence est que le cross-over essaie de converger vers une solution qui lui parait la meilleure (s'intéresse à la qualité), mais que la mutation permet la diversité. En quelque sorte, la mutation sert à éviter une convergence prématurée de l'algorithme. Par exemple, lors de la

recherche d'une solution optimum, la mutation sert à éviter la convergence vers un optimum local.
Cet opérateur est aussi appliqué avec une probabilité P_m. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche aléatoire et conserver le principe de sélection et d'évolution.

Dans la littérature plusieurs opérateurs de mutation ont été définis, tels que : la transposition de deux allèles consécutifs, la transposition de deux allèles quelconques, l'inversion d'allèles, etc.

Les algorithmes évolutionnaires se scindent en grandes familles, dont les principales sont :

1) Programmation évolutionnaire

La programmation évolutionnaire développée par *L.J. Fogel* [Foge, 2003], se base sur l'évolution d'une population d'automates à états finis (Figure II.11) pour résoudre des problèmes de prédiction. Ce modèle évolutionniste accentue l'utilisation de la mutation et n'utilise pas dans sa version originale la recombinaison des individus par croisement [Renn, 2000]. La table de transition des automates est modifiée grâce à des mutations aléatoires uniformes dans l'alphabet discret correspondant. Chaque automate de la population parente génère un enfant par mutation, et les meilleures solutions entre les parents et les enfants sont sélectionnées pour survivre, sachons que l'évaluation de la performance des individus correspond au nombre de symboles prédits correctement.

Par la suite, la programmation évolutionnaire a été développée et son domaine a été élargi par *D.B. Fogel*, afin qu'il puisse travailler dans l'espace réel, où la sélection déterministe est remplacée par un tournoi stochastique.

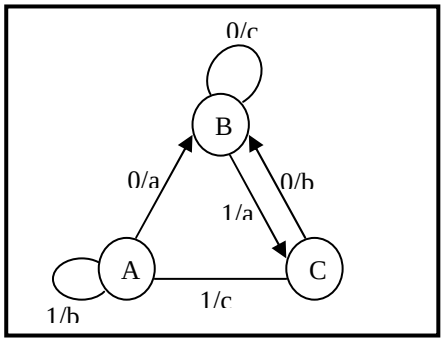


Figure II.11. *Exemple d'un automate à états finis ayant trois états différents S={A,B,C}, un alphabet d'entrée I={0,1}, et un alphabet de sortie O={a,b,c}. Chaque arrête entre deux états indique une transition possible, et la fonction de transition $\delta : S \times I \rightarrow S \times O$ est spécifiée par les labels au niveau des arrêtes ayant la forme i / o, signifiant que $\delta (s_i , i) = (s_j , o)$.*

2) Stratégies d'évolution

Les stratégies d'évolution sont dédiées à l'optimisation des problèmes continus dans l'espace de vecteurs des réels [Magn, 2001]. Les premiers efforts pour la mise en place des stratégies d'évolution ont eu lieu en 1973 à l'université de Berlin par Rechenberg [Rech, 2003] au cours de la résolution d'un problème aérodynamique. C'est avec ces méthodes évolutionnistes que la notion d'auto-adaptativité pour la mutation permettant de contrôler cette fonction de mutation, a été apparue. Une mise en œuvre de ce principe consiste à augmenter l'intensité de la mutation lorsque la proportion de descendants de bonne qualité, c'est à dire le nombre de mutation à succès, dépasse 20% de la population totale [Renn,

2000]. Elle est diminuée dans le cas opposé. Une interprétation possible de cette règle est la suivante : si la proportion de mutation réussie est élevée, l'espace de recherche exploré est restreint autour d'un optimum local, il faut donc diversifier la population en augmentant le taux de mutation, ce qui revient à ajuster la variance de la mutation au cours du temps par le processus d'évolution. C'est ce que les méthodes évolutionnistes auto-adaptatives visent de le faire : automatiser le réglage des paramètres de l'algorithme évolutionniste pour remédier aux méthodes empiriques du type « essais-erreurs » qui sont employées dans la pratique. De plus, ces approches utilisent un opérateur de sélection de type déterministe: les solutions dont le *fitness* est mauvais sont éliminées de la population. En outre, dans le modèle originel, les populations des parents et de leurs descendants sont généralement de taille différente.

3) Algorithmes génétiques

Les algorithmes génétiques (AGs) sont probablement les algorithmes les plus connus et les plus utilisés dans le calcul évolutionnaire. Ils ont été développé dans les années soixante par *Holland* qui les a appliqué à l'optimisation paramétrique pour la première fois en 1975 [Holl, 1975], en posant, ainsi, les fondements de cette technique d'application. Cependant, cette technique n'a pas été appliquée sur des problèmes réels de grande taille, à cause des machines calculatoires, de l'époque, et qui n'ont pas été suffisamment puissantes. Ce n'est que vers la fin des années quatre vingt, précisément, avec l'apparition de l'ouvrage de référence écrit par *Goldberg* [Gold, 1989], que les algorithmes génétiques ont été connus dans la communauté scientifique. Dans ce domaine, d'autres travaux peuvent faire la référence aussi, tel que celui de Michalewicz [Mich, 1996]. Leur particularité est qu'ils sont fondés sur le *Néo-Darwinisme*, c'est-à-dire l'union de la théorie de l'évolution et de la génétique moderne. Ainsi, les variables sont généralement codées en binaire, par analogie avec les quatre lettres de l'alphabet génétique d'ADN, sous forme de gènes dans un chromosome. Ensuite, des opérateurs génétiques, à savoir le croisement et la mutation, sont appliqués à ces chromosomes.

Les AGs sont utilisés pour retrouver une solution résolvant un problème donné, et ce sans avoir de connaissance à priori sur l'espace de recherche. Seul un critère de qualité est nécessaire pour évaluer les différentes solutions en quantifiant, ainsi, leur capacité à résoudre le problème donné. Donc, le but scientifique et technologique visé par ces algorithmes est de pouvoir traiter des problèmes d'optimisation globaux, grâce à la *généralité* avec laquelle on représente l'espace de recherche qui peut contenir des booléens (système actif ou non), des entiers (nombre de composants à optimiser), des réels (intensités associées aux composants réglables), ou des fonctions discrétisées (optimisation de forme), et grâce aussi à la *robustesse* de la convergence [Dupa, 2004].

4) Programmation génétique

L'idée de faire évoluer des programmes date des années cinquante où *Friedberg*, en 1958, a fait plusieurs tentatives pour avoir des ordinateurs auto-programmables en utilisant ce qui est de la mutation actuellement. Donc, et à partir d'une population constituée de programmes aléatoires dont il modifie leurs contenus stochastiquement, il essaye de les améliorer pour aboutir à des résultats satisfaisants. Plutard, *Smith* (1980) travaillant sur les systèmes classifieurs d'apprentissage, a introduit de petits programmes dans les règles qu'il cherche à les faire évoluer [Magn, 2001]. Il convient de noter que, la première utilisation des structures arborescentes dans un algorithme génétique a été suggérée par *Cramer* en 1985 dans le but de faire évoluer des sous-programmes séquentiels d'un langage algorithmique simple.

Toutefois, c'est grâce à *John Koza* (1992) [Koza, 1992] que cette présentation a été adoptée pour définir la programmation génétique comme un nouvel algorithme évolutionnaire, en étendant, ainsi, le modèle d'apprentissage des AGs à l'espace des programmes. Donc, son objectif initial était de faire évoluer des sous-programmes du langage LISP (Figure II.12), et c'est d'ailleurs grâce à son ouvrage que l'utilisation de la programmation génétique s'est étendue à la résolution de nombreux types de problèmes où les solutions peuvent être représentées par des structures arborescentes dont les feuilles sont constituées de symboles terminaux (variables, constantes,…), et les nœuds internes de symboles fonctionnels (opérateurs).

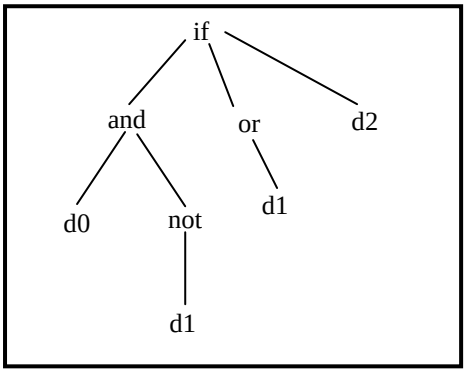


Figure II.12. *Exemple d'une solution Programmation génétique en LISP : {d0, d1, d2} est un ensemble d'instructions constituant les terminaux, et {if, and, or} sont des expressions LISP constituant les nœuds.*

5) Evolution différentielle

L'algorithme à évolution différentielle est inspiré des algorithmes génétiques et de l'évolution des stratégies combinées à une technique de recherche géométrique. Toutefois, l'évolution différentielle s'écarte encore un peu plus des principes de la génétique en abandonnant le codage génotype-phénotype et en faisant varier directement les paramètres réels proches du phénotype. L'évolution différentielle est donc valide pour tout type de fonction objective à valeurs réelles et peut tenir compte des propriétés mathématiques générales de ces fonctions. En revanche, tout comme l'algorithme génétique, l'évolution différentielle utilise une succession de générations définies par des opérations de mutation et de recombinaison de réels et donc possède des notions d'enfants et de parents [Laye, 2010].

L'évolution différentielle a été conçue comme une méthode de recherche parallèle, directe, et stochastique. Chaque solution est encodée avec un vecteur n-dimensionnel basé sur des nombres à virgule flottante. La taille de la population (m) ne change pas au cours du processus d'optimisation (minimisation ou maximisation). A chaque génération, de nouveaux vecteurs sont générés par la combinaison des vecteurs choisis au hasard de la population actuelle. La nouveauté de cet algorithme réside dans l'opérateur de mutation. Contrairement à la mutation des algorithmes génétiques, l'évolution différentielle utilise un concept de mutation auto référentiel se limitant à des combinaisons de mutations déjà présentes dans la population. L'opérateur de mutation consiste à générer de nouveaux vecteurs par le calcul de la différence pondérée de deux (ou quatre) autres vecteurs, selon la formule suivante [Laye, 2010] :

$$\overrightarrow{v_i} = \overrightarrow{a_{r1}} + F \times (\overrightarrow{a_{r2}} - \overrightarrow{a_{r3}})$$

Où i = 1,2,..., m, et les indices aléatoires r1, r2, r3 $\in$ [1,2,..., m] sont mutuellement différents et distincts de l'indice i. Cet opérateur est utilisé en liaison avec l'opérateur de croisement. Ce

dernier est introduit en vue d'augmenter la diversité de la population. Il combine un vecteur muté obtenu par l'opération de mutation avec l'un des vecteurs de la population. Finalement, l'opérateur de sélection compare seulement la valeur de la fonction objective des deux vecteurs en compétition et le meilleur individu est sélectionné pour la population de la prochaine génération.

Un algorithme d'évolution différentielle peut être résumé comme suit [Laye, 2010] :

Algorithme II.7 *Algorithme à évolution différentielle*

Début
Génération aléatoire de la population de vecteurs
Évaluer chaque solution
Répéter
Appliquer la mutation différentielle
Exécuter un croisement différentiel
Évaluer la nouvelle solution
Appliquer la sélection différentielle
Jusqu'à (le nombre de génération est atteint)
Fin.

6) La recherche par dispersion (Scatter Search)

La recherche par dispersion est une méthode d'optimisation relativement ancienne. Cette approche évolutionnaire a pour origine les stratégies de création de règles de décision composées et de contraintes de remplacement. Les études récentes ont démontré les avantages pratiques de cette approche pour résoudre divers problèmes d'optimisation. La recherche par dispersion contraste avec d'autres procédures évolutionnaires, telles que les algorithmes génétiques, en utilisant des conceptions stratégiques là où d'autres approches utilisent l'aléatoire [Jour, 2003].

La recherche par dispersion opère sur un ensemble de solutions appelé l'*ensemble de référence*, en les combinant pour en créer des nouvelles. À la différence d'une "population" dans les algorithmes génétiques, l'ensemble de référence de solutions dans la recherche par dispersion est relativement petit.

La recherche par dispersion comprend les cinq méthodes suivantes [Jour, 2003] :

- ✓ Une méthode de génération de diversification pour produire une collection de solutions diverses, en utilisant une solution d'essai arbitraire (ou solution initiale) comme entrée.
- ✓ Une méthode d'amélioration pour transformer une solution initiale en une ou plusieurs solutions d'essai améliorées.
- ✓ Une méthode de mise à jour de l'ensemble de référence pour construire et maintenir un ensemble de référence comprenant les meilleures solutions trouvées, organisé pour fournir l'accès efficace par d'autres parties de la méthode. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité.
- ✓ Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ces solutions comme une base pour créer des solutions combinées.

- ✓ Une méthode de combinaison de solutions pour transformer un sous-ensemble donné de solutions produit par la méthode de génération d'un sous-ensemble en un ou plusieurs vecteurs combinés de solutions.

7) Algorithmes à Estimation de Distribution (Estimation of Distribution Algorithms)

Récemment, une nouvelle classe d'algorithmes a fait son apparition : les algorithmes à estimation de distribution (EDA). Les stratégies évolutionnaires mettent en œuvre des opérateurs de mutation et de croisement mais il est difficile pour un utilisateur inexpérimenté de choisir l'opérateur approprié à son problème. Les algorithmes à estimation de distribution reprennent les principes des algorithmes à population mais utilisent des modèles probabilistes à la place d'opérateurs de mutation et de croisement pour construire de nouveaux individus.

Le modèle de fonctionnement de l'algorithme est le suivant [Jour, 2003] :

1. Génération de la population initiale.
2. Sélection des individus prometteurs.
3. Estimation de la distribution de ces individus (construction d'un modèle probabiliste).
4. Génération de nouvelles solutions à partir du modèle probabiliste.
5. Retour en 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Les EDA peuvent être appliqués aussi bien sur un domaine discret que sur un domaine continu. Il existe de nombreux algorithmes à estimation de distribution qui peuvent être classés selon le modèle utilisé pour la construction des nouveaux individus [Jour, 2003]: compact GA (produit de distribution de Bernouilli), Population Based Incremental Learning (règle de Hebian), Univariate Marginal Distribution Algorithm, extended compact GA (produit de distribution marginale) et Bayesian Optimization Algorithm (réseau bayesien).

8) Les algorithmes bactériologiques

Les algorithmes bactériologiques basés sur les algorithmes génétiques et qui sont donc assez similaires [Baud, 2005a], sont apparus récemment en tant que métaheuristique. Ils ont été développés par l'équipe Triskell [www12] à Rennes, notamment grâce à la thèse de Benoit Baudry [Baud, 2005b]. Lors du développement d'un générateur de cas de tests utilisant la technique de la mutation-based testing [Baud, 2005c], en utilisant les algorithmes génétiques, l'équipe s'est aperçue que l'utilisation d'algorithmes génétiques n'était pas bien adaptée. En effet, les nouveaux cas de tests n'étaient créés que par l'opérateur de mutation, le croisement ne faisant que récrire des cas déjà existants. L'équipe a donc eu l'idée d'utiliser la reproduction des bactéries, qui se multiplient en se clonant et des mutations s'opèrent. Les bactéries ayant le meilleur patrimoine génétique sont celles qui survivent le mieux dans leur environnement.

Les différences entre algorithmes génétiques et bactériologiques se situent au niveau du croisement des individus et au niveau de la sélection des individus. Dans l'algorithme bactériologiques l'opérateur de croisement a disparu, et pour ne pas perdre les bactéries les mieux adaptées, les meilleures sont sauvegardées lors de la sélection, à chaque itération. Il est possible d'en sauvegarder un certain nombre, mais aussi de sauvegarder les X meilleures. Les premières itérations entraînent alors la sauvegarde de toutes les bactéries.

Un algorithme bactériologique se déroule de manière très similaire à un algorithme génétique. Il opère suivant les étapes suivantes :

- ✓ initialisation du temps,
- ✓ création de la population initiale,
- ✓ tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint], faire:
  - incrémentation du temps,
  - mutations aléatoires à la population (Toutes les bactéries mutent),
  - évaluation de l'adaptation de chaque bactérie,
  - sauvegarde des meilleures bactéries.

Actuellement, la distinction entre ces approches est de plus en plus floue. Le génotype, qui est le codage d'un individu, étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, ...). Donc, la différence entre ces quatre catégories est essentiellement d'ordre historique.

b. Algorithmes de colonies de fourmis

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis, à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [Dene, 1983], [Dene, 1989]. Les fourmis trouvent le plus court chemin entre leur nid et une source de nourriture. La résolution de ce problème, qui est assez complexe en soi, fait appel à une certaine organisation et à un travail collectif. Des fourmis peuvent résoudre ce problème collectivement en se basant sur un moyen de communication particulier : « la phéromone ».

1) Les fourmis réelles

Les fourmis réelles sont aveugles et cherchent de la nourriture en se déplaçant de façon quasi aléatoire. Tout au long de leur déplacement, elles laissent derrière elles une substance chimique appelée phéromone. Cette substance a pour but de guider les fourmis vers leur objectif et possède la propriété de s'évaporer au cours du temps. Une fois cet objectif atteint (dans ce cas, la nourriture trouvée), les fourmis rentrent au nid, en rebroussant chemin, grâce à leur trace de phéromone. Celle-ci s'en trouve renforcée. Plus une trace de phéromone est concentrée, plus elle va attirer les fourmis. Au fil du temps, le plus court chemin, du nid vers la nourriture émergera, grâce au renforcement de la trace de phéromone (figure II.13). D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs congénères.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis. Ainsi, une colonie est capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [Goss, 1989] [Beck, 1992], sans que les individus aient une vision globale du trajet.

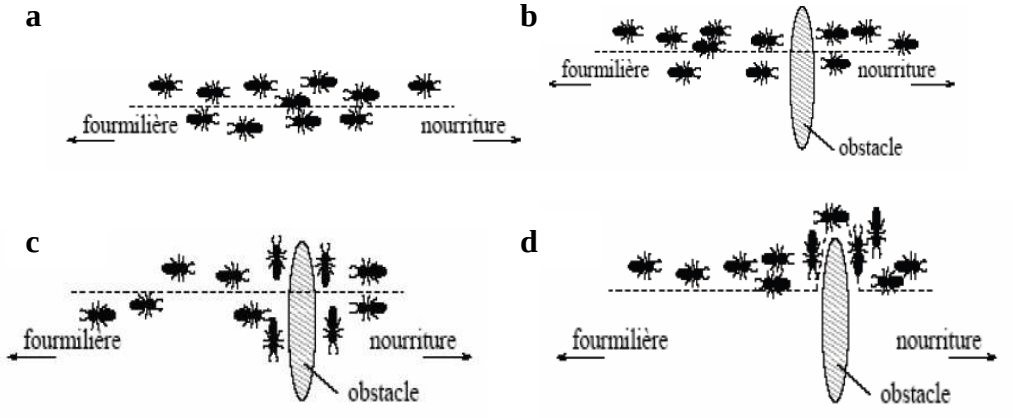


Figure II.13. (a) *Les fourmis suivent un chemin entre la fourmilière et la nourriture,*
(b) *Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité.*
(c) *La phéromone s'évapore sur le chemin le plus long.*
(d) *Toutes les fourmis choisissent le chemin le plus court.*

2) Les algorithmes de colonies de fourmis

Le premier algorithme à base de fourmis a été proposé par Dorigo en 1992 [Dori, 1992]. La métaheuristique d'optimisation par colonies de fourmis (ACO, Ant Colony Optimization) qui est une approche bio-inspirée [Dori, 1999], [Dori, 2004] est une généralisation des premiers algorithmes à base de fourmis.

Ce paradigme consiste à modéliser le problème à résoudre en une recherche d'un meilleur chemin dans un graphe et à utiliser des fourmis artificielles pour rechercher les "bons" chemins dans ce graphe. Le comportement de ces fourmis artificielles est inspiré des fourmis réelles : *(i)* les fourmis déposent de la phéromone pour marquer les chemins prometteurs, *(ii)* elles se déplacent dans le graphe de construction en choisissant leur chemin selon une probabilité dépendant des traces de phéromones précédemment déposées, et *(iii)* la quantité de phéromone déposée sur les chemins décroît à chaque cycle de l'algorithme afin de simuler le phénomène d'évaporation de la phéromone observé dans la nature.

Une formalisation plus précise existe [Dori, 2003]. Elle passe par une représentation du problème, un comportement de base des fourmis et une organisation générale de la métaheuristique. Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes, notamment la définition des pistes de phéromone en tant que mémoire adaptative, la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une recherche locale.

c. Algorithmes à essaim de particules (Particle Swarm Optimiser)

Ces algorithmes sont inspirés des essaims d'insectes [Cler, 2004] (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés *particules* et la population est appelée *essaim*.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou socio-

métriquement (position dans l'essaim de l'individu). Les nouvelles vitesses et direction de la particule seront définies en fonction de trois tendances : la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues.

II.6. Quelle métaheuristique à utiliser ?

Le premier problème pratique qui se pose à un utilisateur confronté à une application concrète est d'effectuer un choix parmi les différentes métaheuristiques disponibles. Ce choix est d'autant plus difficile qu'il n'existe pas de comparaison générale et fiable des différentes métaheuristiques. Cependant, il est possible de caractériser les métaheuristiques selon quelques critères généraux, ce qui pourrait faciliter ce choix. Le tableau de synthèse ci-dessous met en relation cinq critères avec six métaheuristiques parmi les plus représentatives. Les indications sont présentées à titre purement indicatif et correspondent aux travaux de [JinK, 1999] et aux résultats publiés. Il doit être clair que le choix d'une métaheuristique appropriée ne constitue qu'une condition nécessaire. La qualité des solutions trouvées par une méthode peut être très variable selon l'implémentation réalisée.

	AI	**RS**	**Tabou**	**AG**	**ACO**	**AH**
Simplicité	0	-	-	-	- -	- -
Facilité d'adatp.	0	0	-	-	-	-
Connaissance	0	0	+	+	+	+
Qualité	0	+	++	+	+	++ (+++)
Rapidité	0	-	-	- -	- -	- -

Tableau II.1. *Comparaison générale des principales métaheuristiques.*

Dans ce tableau, les six métaheuristiques comparées sont les suivantes :

- ✓ AI : amélioration itérative (descente) avec relance,
- ✓ RS : recuit simulé,
- ✓ tabou : méthode tabou,
- ✓ AG : algorithme génétique adapté,
- ✓ ACO : optimisation par colonies de fourmis,
- ✓ AH : algorithme hybride.

Les critères de comparaisons retenus sont les suivants :

- ✓ simplicité de la métaheuristique, *i.e.* simplicité de la méthode elle-même,
- ✓ facilité d'adaptation au problème,
- ✓ possibilité d'intégrer des connaissances spécifiques du problème,
- ✓ qualité correspond à la meilleure qualité qu'il est possible d'obtenir par une exécution prolongée,
- ✓ rapidité, *i.e.*, le temps de calcul nécessaire pour trouver une telle solution (sur une machine séquentielle).

La méthode d'amélioration itérative est utilisée comme point de référence pour l'ensemble des méthodes : les signes -, 0, + indiquent des performances respectivement inférieures, égales, supérieures à celles obtenues par l'amélioration itérative.

II.7. Conclusion

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires. Elles ont révélé leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand nombre de problèmes d'optimisation classiques et d'applications réelles de grande taille. C'est pourquoi l'étude de ces méthodes reste toujours en plein développement.

Si on peut constater la grande efficacité des métaheuristiques pour de nombreuses classes de problèmes, il existe en revanche peu de résultats permettant de comprendre la raison de cette efficacité. Nous possédons bien des comparaisons entre métaheuristiques sur différentes classes de problèmes mais nous ne savons généralement ni expliquer le fonctionnement d'une métaheuristique, ni prévoir son efficacité sur une instance donnée.

Dans les suivants chapitres, nous présenterons nos travaux de recherche qui se résument en l’application des métaheuristiques pour résoudre le problème de cryptage de données texte et images.

CHAPITRE III

CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

III.1. Introduction

Dans la littérature, les méthodes heuristiques sont réparties en deux classes : les algorithmes spécifiques à un problème donné, qui utilisent des connaissances du domaine [Talb, 1999], et les algorithmes généraux applicables à une grande variété de problèmes : les métaheuristiques [Dréo, 2003]. Dans ce chapitre, notre intérêt va se porter sur la deuxième classe d'algorithmes. Pour résoudre des problèmes multi-objectifs ou mono-objectifs et déterminer des solutions optimales, plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les plus connues de ces adaptations peuvent être trouvées dans [Coll, 2002].

Etant donné la complexité du problème traité qui celui de cryptage de données texte ou images, nous avons choisi d'utiliser les métaheuristiques pour le résoudre. La première méta-heuristique testée est un algorithme évolutionnaire (AE).

Le principal avantage des AEs vient de leur capacité à traiter le problème en ne possédant qu'un minimum d'informations sur celui-ci et en ne laissant aucun détail sur les calculs intermédiaires menant aux résultats. Ce dernier point convient parfaitement au domaine de chiffrement de données pour compliquer voir même pénaliser toutes tentatives de cryptanalyse.

Ainsi, le problème considéré qui est celui de chiffrement de données, peut être résolu par une procédure d'optimisation par AEs qui à partir d'un ensemble de solutions initiales, ou population de N individus, elle consiste à faire évoluer cette population en utilisant des opérateurs de sélection, de croisement et de mutation. A chaque itération de l'algorithme, une nouvelle population de solutions ou d'individus est générée. Tout d'abord, un ensemble d'individus est sélectionné pour générer la population suivante. Ces individus sont ensuite croisés pour créer de nouveaux individus et compléter la nouvelle population. Certains de ces nouveaux individus peuvent subir une mutation. Le critère d'arrêt de l'algorithme dans notre cas est un nombre d'itérations sans amélioration de la meilleure solution trouvée.

Algorithme 1 Structure générale de l'algorithme génétique

1: $\mathcal{P}_{courant} \leftarrow$ Initialiser une population de N individus
2: Evaluer chaque individu de $\mathcal{P}_{courant}$
3: $S_{best} \leftarrow$ Le meilleur individu $S \in \mathcal{P}_{courant}$
4: $I \leftarrow 0$
5: **Tant que** $I < \#ite$ **faire**
6: $\quad \mathcal{P}_{enfant} \leftarrow \emptyset$
7: $\quad$ **Pour** $j = 0$ à $j = N/2$ **faire**
8: $\quad\quad (P_1, P_2) \leftarrow$ Sélectionner deux individus parents de $\mathcal{P}_{courant}$
9: $\quad\quad (E_1, E_2) \leftarrow$ Croiser les deux parents (P_1, P_2) pour obtenir deux individus enfants
10: $\quad\quad \mathcal{P}_{enfant} \leftarrow \mathcal{P}_{enfant} \cup \{E_1, E_2\}$
11: $\quad$ **Fin pour**
12: $\quad$ Muter aléatoirement des individus de la population $\mathcal{P}_{enfant}$
13: $\quad \mathcal{P}_{courant} \leftarrow \mathcal{P}_{enfant}$
14: $\quad$ Evaluer chaque individu de $\mathcal{P}_{courant}$
15: $\quad$ **Si** il existe un individu $S \in \mathcal{P}_{courant}$ meilleur que S_{best} **alors**
16: $\quad\quad S_{best} \leftarrow S$
17: $\quad\quad I \leftarrow 0$
18: $\quad$ **Fin si**
19: $\quad I \leftarrow I + 1$
20: **Fin Tant que**

Algorithme III.1. *Structure générale d'un AE.*

Les éléments importants d'un algorithme évolutionnaire sont le codage et l'évaluation d'un individu (étapes 2 et 14), l'initialisation d'une population (étape 1), la sélection (étape 8), le croisement (étape 9) et la mutation des individus (étape 12). Ces éléments sont décrits dans les pages qui suivent, dans le cadre de l'adaptation que nous en avons faite au problème de cryptage.

Dans ce chapitre, nous présentons des nouveaux algorithmes de cryptage de données texte et images basés sur les AEs et opérant suivant deux modes : un chiffrement à base de positions et un chiffrement à base d'occurrences [Soui, 2008a], [Soui, 2008b] [Soui, 2009], [Sema, 2009a], [Sema, 2009b], [Soui, 2010a], [Soui, 2010b], [Soui, 2011a] et [Soui, 2011b]. Ce chapitre est scindé en cinq sections où la première présente une introduction au chapitre. La deuxième section démontre l'adaptabilité d'exploitation des AEs pour résoudre le problème de cryptage en énumérant les principaux points de motivation ; suivie de la troisième section consacrée à une formulation du problème de cryptage en tant que problème d'optimisation. Les hypothèses sur lesquelles repose cette approche sont exposées aussi dans cette section. Ensuite, une description détaillée des algorithmes développés est traitée dans la quatrième section. Ainsi, elle englobe la présentation d'algorithmes de cryptage développés, le réglage des paramètres de la métaheuristique, ainsi que les résultats. La cinquième section présente une étude comparative des algorithmes développés, d'une part, et des méthodes de référence, d'autre part. Le chapitre sera terminé par une conclusion.

Les données réelles de test que nous avons choisies pour illustrer les différents algorithmes, sont présentées sur la figure III.1. Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26 GHz.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

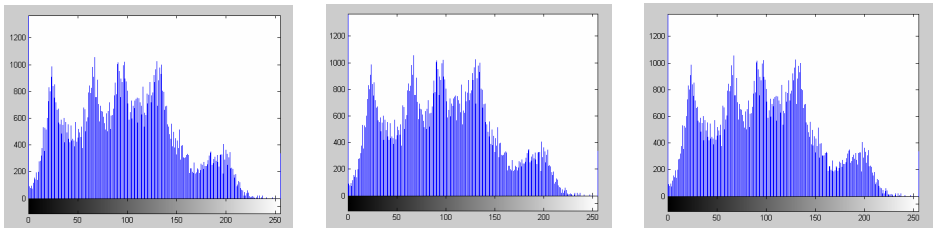
(a)

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

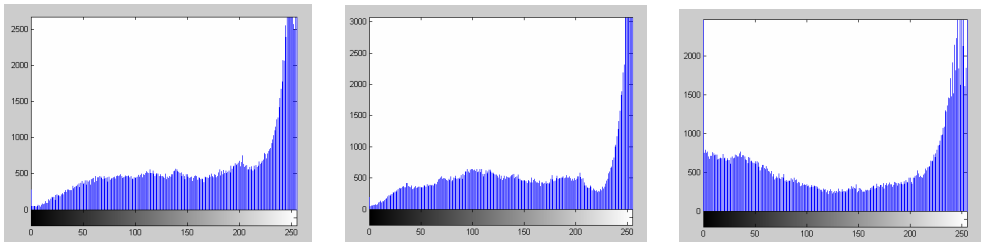
(b)

(c)

(d)



(e)



(f)

Figure III.1. *Données test.* ***(a)*** *Texte 1 (Taille = 1084),* ***(b)*** *Texte 2 (Taille = 1151),*
(c) *Image Lena,* ***(d)*** *Image Logo,* ***(e)*** *Histogrammes de l'image Lena,* ***(f)*** *Histogrammes de l'image Logo.*

III.2. Motivations

Les algorithmes évolutionnaires ont montré leur efficacité dans la résolution de nombreux problèmes et notamment dans les problèmes d'optimisation. La modélisation de l'espace de recherche que nous avons adoptée nous permet de tirer parti des propriétés des AEs en optimisation dans notre problème.

Notre approche se distingue des autres approches cryptographiques construites autour de principes mathématiques complexes (problèmes réputés difficiles), dans la mesure où nous modélisons le problème de cryptage à un niveau proche d'un paysage de qualité (fitness landscape) : l'AE traite directement avec les points de l'espace de recherche, les données.

Les algorithmes que nous avons mis au point bénéficient des avantages de la stratégie de recherche évolutionnaire surtout du bon comportement des AEs en ce qui concerne la résolution du dilemme d'exploration versus exploitation [Holl, 1975] : il décide quelles solutions intermédiaires proposées sont à garder en priorité et quelles sont les autres à éliminer.

III.3. Formulation du problème de chiffrement

Dans cette section, nous montrons que le chiffrement de données peut se ramener à un problème d'optimisation. D'où l'adaptabilité d'utiliser une métaheuristique.

Le chiffrement d'une donnée *D* (texte ou image) qui est une suite de *n* éléments (caractères ou pixels) *e* utilisant un attribut d'égalité *E*, génère une donnée chiffrée *D´* qui est une suite de *n* éléments *e´*, telle que :

1- $D = \{e_i\}, i \in [1,n]$

2- $D' = \{e_i'\}, i \in [1,n]$

3- $E(e_i, e_i') = Faux, \forall i \in [1,n]$

Nous faisons observer que l'unicité de chiffrement n'est pas garantie par ces trois conditions. Pour réduire le problème de la non unicité de la solution, le problème de chiffrement est régularisé par une contrainte d'optimisation d'une fonction *F*, caractérisant la qualité d'un bon chiffrement. Donc, une quatrième condition est ajoutée aux trois premières :

4- $F(D^*) = Max_{D' \in C(D)} F(D')$

Où *F* est une fonction mesurant le degré de confusion des données, D^* est la donnée chiffrée optimale ou plutôt la meilleure donnée chiffrée trouvée, *C(D)* est l'ensemble des données chiffrées possibles de *D*.

Il est clair que la condition 4 ne résout pas entièrement le problème d'unicité de chiffrement. Il demeure des cas où plusieurs chiffrements peuvent avoir la même valeur optimale. Toutefois, ce problème peut être réglé expérimentalement en fixant un nombre maximal d'itérations du processus de chiffrement.

La détermination d'un niveau de confusion mesurant le degré de dissimilarité entre la donnée originale et la donnée chiffrée correspondante rend le cryptage de données assimilable à un problème d'optimisation. D'où notre approche de cryptage au travers des méthodes heuristiques et des techniques destinées à résoudre ce type de problèmes.

Cette reformulation du problème de cryptage de données en un problème d'optimisation, nous conduit à la section suivante, où nous allons présenter les différents algorithmes proposés.

III.4. Algorithmes proposés

En cryptage de données, les AEs ont été récemment appliqués à travers le travail de Omary Fouzia [Omar, 2007]. C'est d'ailleurs l'unique méta-heuristique qui a été utilisée pour résoudre ce problème. Une étude comparative entre ce travail et notre proposition d'AEs de cryptage sera ensuite présentée.

Dans cette approche proposée, nous nous intéressons au cryptage des données texte et images. Le schéma général du processus de sécurisation proposé est celui illustré par le synoptique de la figure III.2.

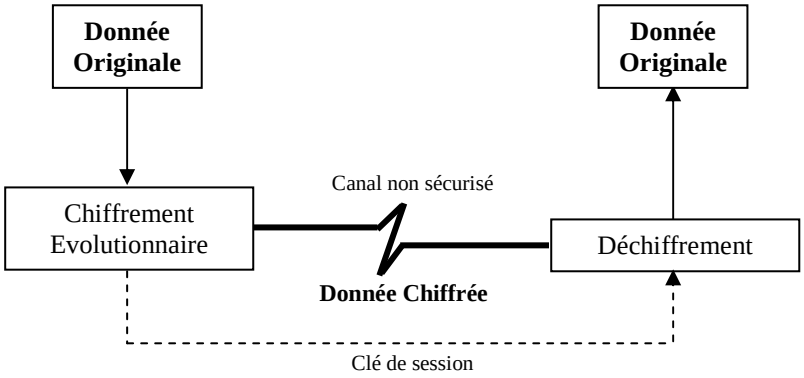


Figure III.2. *Schéma général du processus de sécurisation proposé.*

Dans ce cas et comme la première étape de la conception d'un AE est de coder (représenter) les solutions sous forme de chromosomes qui sont des chaînes de gènes sachons que cette étape dépend principalement de l'application envisagée et du but recherché [Prab, 1998], alors, nous avons proposé deux modes de chiffrements différents : chiffrement à base de position et chiffrement à base d'occurrences. Ainsi, trois algorithmes différents en résultent, un premier algorithme et un deuxième représentant une variante plus sécurisée du premier pour le cas de manipulation des images représentant des données spéciales pour le premier mode et un troisième algorithme pour le second mode.

Donc, ce qui fait la différence entre les trois algorithmes proposés est le codage utilisé et le paramétrage employé. Toutefois, le schéma global du processus de chiffrement ou de déchiffrement (illustré en détail dans les sections qui suivent) est le même. C'est celui donné ci-dessous :

Phase de chiffrement

Début

1) Définition d'un codage du problème,

Répéter

2) création de la population initiale,

3) sélection parmi les parents (individus de la population courante) ceux qui vont avoir des enfants,

4) application des opérateurs de variation (croisement / mutation) aux parents sélectionnés (génération des enfants),

5) évaluation des performances des individus,

6) sélection, parmi les parents et les enfants, de ceux qui vont survivre à la génération suivante.

Jusqu'à la satisfaction d'un critère d'arrêt.

7) Calcul de la clé de session correspondante à la solution produite.

Fin

Phase de déchiffrement

Début
Si La clé de déchiffrement est la bonne clé calculée **Alors**
1) Calcul du chromosome codant la donnée originale
2) Génération de la donnée originale à partir du chromosome calculé
Fin si
Fin

III.4.1. Chiffrement à base de positions

Dans cette section nous présentons deux nouveaux algorithmes de chiffrement symétrique s'inscrivant sous ce premier mode proposé où le deuxième est une variante du premier. Nous les avons appelé *PosESecL1* et *PosESecL2* pour *Position based Evolutionary Secure Level 1* (niveau 1 de sécurité évolutionnaire basé positions) et *Position based Evolutionary Secure Level 2* (niveau 2 de sécurité évolutionnaire basé positions), respectivement.

III.4.1.1. Description de PosESecL1

Avant de décrire les étapes du processus évolutionnaire en allant de la génération de la population initiale jusqu'à l'obtention du résultat final, il faut définir le codage adéquat des individus.

a. Codage

Le codage dépend étroitement du problème à résoudre. En effet sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide.

Dans le cas de manipulation d'une donnée *D* qui soit une suite de *n* éléments e_i, le codage adopté sera celui décrit à travers la figure III.3. Il consiste à transformer la donnée en un code particulier représenté par un chromosome qui est un ensemble de gènes représentant chacun le codage d'un élément e_i. Donc, en considérant le codage d'un certain élément, nous cherchons à permuter sa position (son emplacement) avec un autre élément de telle sorte que la différence entre les codages des deux éléments soit maximale. Il s'agit, ainsi, d'un problème d'optimisation où le but de l'algorithme proposé est de désordonner les positions des éléments suivant les contraintes du codage utilisé.

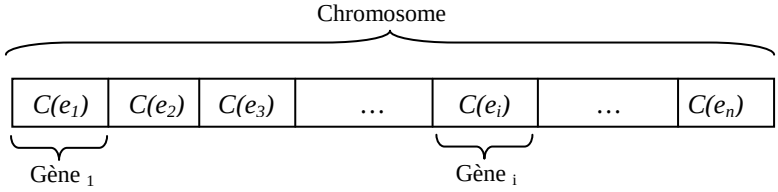


Figure III.3. *Codage adopté des données texte / images sous PosESecL1.*

Où $C(e_i)$ représente le codage de l'élément *i*.

Dans le cas des données texte, $C(e_i)$ représente la codification des caractères constituant la donnée. Etant donné qu'une donnée texte soit une suite de caractères appartenant à la liste des 1393 caractères affichables en Unicode et couvrant plusieurs sous-ensembles de caractères : Latin de base, Latin étendu-A, latin étendu-B, lettres de modification d'espace, Grec, Cyrillique, Hébreu, Hébreu étendu, Arabe, Arabe étendu, ponctuation générale, etc. Donc, $C(e_i)$ représente le codage Unicode du caractère e_i. La représentation que nous avons utilisée est le codage hexadécimal malgré que le codage entier soit tout à fait adapté.

Dans le cas des données images, et vu que l'espace de représentation choisis est le RVB (Rouge Vert Bleu), donc, nous avons utilisé un codage entier malgré que le codage binaire semble tout à fait adapté. Toutefois, cette représentation semble peu appropriée, dans notre cas, car elle nécessite deux opérations supplémentaires, le codage et le décodage qui n'apportent aucun plus par rapport au codage choisi.

Ainsi, les chromosomes sont des chaînes de *n* gènes ayant comme structure celle représentée à travers la figure III.4, où *n* représente la taille de l'image à chiffrer en terme de pixels. Donc, en considérant les trois composantes R_i, V_i et B_i relatives au $i^{ème}$ pixel de l'image, nous cherchons à permuter sa position (son emplacement) avec un autre pixel, ayant comme rang j et représenté par les composantes R_j, V_j et B_j, de telle manière que les différences entre les composantes (R_i,R_j), (V_i,V_j) et (B_i,B_j) soient maximales.

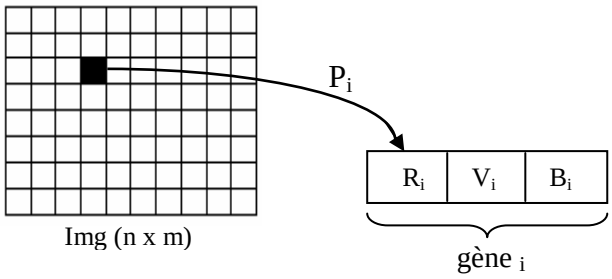


Figure III.4. *Représentation d'un pixel P_i de l'image Img ($n \times m$) sous la forme d'un gène.*

b. Création de la population initiale

La grande partie des versions des algorithmes évolutionnaires garde la même taille de la population des solutions potentielles, appelées aussi *individus*. La population initiale peut être choisie de manière aléatoire, donnée par l'utilisateur ou une partie est créée et l'autre est donnée (pour introduire dès le début de bonnes solutions construites, par exemple, à partir de certaines heuristiques). Cependant d'autres mécanismes d'initialisation peuvent être utilisés suivant l'application [Bhanu, et al., 1995].

Dans notre cas, les individus formant la population initiale sont obtenus par application de perturbations aléatoires sur le chromosome initial représentant le codage d'une donnée soumise au chiffrement.

c. Reproduction

Le but de notre algorithme PosESecL1 est de désordonner les positions des éléments d'une donnée originale de façon évolutionnaire ne laissant aucune trace des calculs intermédiaires menant au résultat final qui est le chiffré de la donnée initiale, sans modifier les valeurs des éléments. Donc, nous devons être très prudent lors de la manipulation des chromosomes durant les différentes étapes du processus évolutionnaire ; surtout, lors de l'application des opérateurs génétiques.

Ci-dessous, nous décrivons les opérateurs de reproduction exploités par le PosESecL1.

- **Croisement**

L'opérateur de croisement a pour but d'enrichir la diversité de la population, en manipulant la structure des chromosomes. Classiquement, les croisements sont réalisés à partir de deux parents et génèrent deux enfants. Une description plus détaillée a été exposée dans le deuxième chapitre.

Comme opérateur de croisement, notre choix s'est porté sur l'opérateur OX « Order Cross-over » proposé par Davis [Davi, 1985]. C'est un opérateur à deux points de croisement. Il consiste à générer des descendants en trois phases :

1**)** Choisir dans les deux parents une sous-séquence interne, comprise entre deux points de coupure tirés aléatoirement.

$Parent_1$: 1 3 5 7 9 | **2 4 6** | 8 10

$Parent_2$: 10 1 9 2 8 | **7 3 4** | 6 5

2) Recopier la sous-séquence interne du Parent1 dans le descendant Enfant1 aux mêmes positions et retirer du chromosome Parent2 les allèles compris dans cette sous-séquence.

$Enfant_1$: • • • • • | **2 4 6** | • •

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Le chromosome Parent2 permet alors de former une séquence d'allèles résiduelle en partant du deuxième point de coupure et en considérant le chromosome comme une chaîne circulaire.

3) compléter les gènes disponibles du descendant Enfant1 en lui transmettant dans l'ordre les allèles issus de la séquence résiduelle précédente.

$Enfant_1$: 1 9 8 7 3 | **2 4 6** | 5 10

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Les deux points de croisement sont choisis aléatoirement. Le meilleur taux de croisement varie entre 60% et 100% [Gren, 1986].

- **Mutation**

Pratiquement, le rôle de la mutation consiste à faire apparaître de nouveaux gènes. Cet opérateur introduit une diversité nécessaire à l'exploration de l'espace de recherche. Les mutations jouent, alors, le rôle de bruit et empêchent l'évolution de se figer.

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

d. Evaluation des individus

La fonction d'évaluation (ou d'adaptation) quantifie la qualité de chaque chromosome par rapport au problème. Les chromosomes ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction, et donc plus de chance pour que la population suivante hérite de leur matériel génétique. La fonction d'adaptation produit la pression qui permet de

faire évoluer la population de l'algorithme évolutionnaire vers les individus de meilleure qualité. En clair, le choix de la fonction d'évaluation va fortement influer sur le succès de l'algorithme.

La fonction d'évaluation que nous avons définie pour évaluer nos individus, est celle donnée ci-dessous :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.1)}$$

Avec : $C(e_j)_i$ est le codage du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.

Dans le cas de manipulation d'une donnée image, la fonction d'évaluation prend l'instanciation suivante :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \qquad \text{(III.2)}$$

Où : R_{ji} (resp V_{ji}, B_{ji}) est la valeur de la composante R (resp V, B) du j^{ème} gène du i^{ème} individu.
n est la taille de l'image à chiffrer en terme de pixels.

e. Sélection des individus les plus adaptés

Le rôle de la sélection est de distinguer entre les individus sur la base de leur qualité, en particulier, pour permettre aux meilleurs individus de devenir parents dans la génération suivante. Ainsi, elle est responsable sur le fait de pousser l'amélioration de la qualité.

Dans notre cas l'opérateur utilisé est une sélection de type roulette avec la possibilité de sélectionner plusieurs fois le même individu. Ainsi, les meilleurs individus ont plus de chance d'être sélectionnés par rapport au moins bons individus.

f. Critère d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est d'assurer l'optimalité de la solution finale obtenue par l'algorithme évolutionnaire.

Les critères d'arrêts sont de deux natures :

1- Arrêt après un nombre fixé à priori de générations. C'est la solution retenue lorsqu'une durée maximale de temps de calcul est imposée.
2- Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé. C'est d'ailleurs celui que nous avons principalement utilisé pour assurer la convergence de notre algorithme proposé.

Pour notre problème et lors de la manipulation d'une donnée texte, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.3)}$$

Et
comme :

$$((0020 \le C(e_j)_i \le FEFC) \,\&\, (0020 \le C(e_j)_0 \le FEFC)) \Rightarrow (0 \le \left| C(e_j)_i - C(e_j)_0 \right| < FEFC)$$

(III.4)

$$(III.4) \Rightarrow 0 \le \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| < FEFC \times n$$

D'après l'inéquation (III.4), F est une fonction bornée donc, la fonction d'arrêt mettant fin au processus de résolution est la suivante :

$$0 \le F(I_i) < FEFC \times n \quad (III.5)$$

telles que : $(FEFC)_{16} = (65276)_{10}$ et $(0020)_{16} = (0032)_{10}$

Dans le cas de manipulation d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \quad (III.6)$$

Et comme :

$$((0 \le R_{ji} \le 255) \,\&\, (0 \le R_{j0} \le 255)) \Rightarrow (0 \le \left| R_{ji} - R_{j0} \right| \le 255) \quad (III.7)$$

$$((0 \le V_{ji} \le 255) \,\&\, (0 \le V_{j0} \le 255)) \Rightarrow (0 \le \left| V_{ji} - V_{j0} \right| \le 255) \quad (III.8)$$

$$((0 \le B_{ji} \le 255) \,\&\, (0 \le B_{j0} \le 255)) \Rightarrow (0 \le \left| B_{ji} - B_{j0} \right| \le 255) \quad (III.9)$$

D'après (III.7), (III.8) et (III.9) nous aurons :

$$0 \le \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \quad (III.10)$$

$$(III.10) \Rightarrow 0 \le \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \times n$$

$$\Leftrightarrow 0 \le F(I_i) \le 255 \times 3 \times n \quad (III.11)$$

D'après l'inéquation (III.11), nous constatons que F est une fonction bornée. Ainsi, la condition d'arrêt est la suivante :

$$0 \le F(I_i) \le 255 \times 3 \times n$$

g. Déchiffrement

Le déchiffrement est l'opération inverse du chiffrement. Elle permet d'obtenir la version originale d'une donnée qui a été précédemment chiffrée.

Dans la deuxième phase de l'algorithme correspondant à cette opération, nous exploitons deux informations qui sont la donnée originale et la donnée chiffrée pour générer une *clé de session* qui peut être d'un usage symétrique ou asymétrique. Cette clé représente la permutation des positions des nombres d'occurrences des valeurs d'éléments codant la donnée chiffrée pour obtenir ceux des valeurs d'éléments codant la donnée originale. De ce fait, elle varie d'une donnée à l'autre puisqu'elle dépend de la donnée et de sa taille. Et ce n'est que par l'introduction de la clé appropriée que les éléments de la donnée chiffrée rejoignent leurs positions initiales pour retrouver la donnée originale.

Remarque :

La notion de *clé de session* est un compromis entre le chiffrement symétrique et asymétrique permettant de combiner les deux techniques. Son principe est simple : il consiste à générer aléatoirement une clé de session de taille raisonnable, et de chiffrer celle-ci à l'aide d'un algorithme de chiffrement à clef publique (plus exactement à l'aide de la clé publique du destinataire). Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée. Ainsi, expéditeur et destinataire sont en possession d'une clé commune dont ils sont seuls connaisseurs. Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

h. Réglage des paramètres et résultats

Dans cette partie, nous présentons des résultats de cryptage de données test (présentées sur la Figure III.1), obtenus avec l'algorithme PosESecL1. Cette partie est scindée en deux sous-parties : dans la première, nous détaillons les réglages de l'algorithme. La deuxième sous-partie est consacrée aux résultats de cryptage des données test.

- **Réglage des paramètres**

En littérature, plusieurs travaux ont traité ce problème tels que : [Lobo, 2004], [DeJo, 2007], [Eibe, 2007], [Preu, 2007], [Mich, 2007] et [Fern, 2007]. Toutefois, la nécessité de l'étape de réglage vient des inconvénients majeurs des métaheuristiques : plusieurs paramètres à régler et l'inexistence de réglages par défaut. En outre, chaque problème traité a ses propres réglages. Donc et pour pouvoir choisir les bons paramètres relatifs aux taux de croisement et de mutation, il a fallu appliquer l'algorithme plusieurs fois. Les différentes valeurs de test appartiennent aux intervalles [0.6, 1] et [0.001,0.05] pour les probabilités de croisement et de mutation respectivement. Les figures III.5 et III.6 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation. Il est à signaler que les résultats présentés sont la moyenne de 10 exécutions successives d'un chiffrement de l'image Lena.

Les valeurs des paramètres finalement adoptées par PosESecL1 sont récapitulées dans le tableau III.1. Cependant, il est à noter que les très nombreux paramètres et leurs fortes interactions rendent impossible un réglage optimal pour toutes les instances. Certains choix sont cependant mauvais, d'autres robustes ; voici la conclusion au quelle différents tests numériques ont amené pour cet algorithme évolutionnaire.

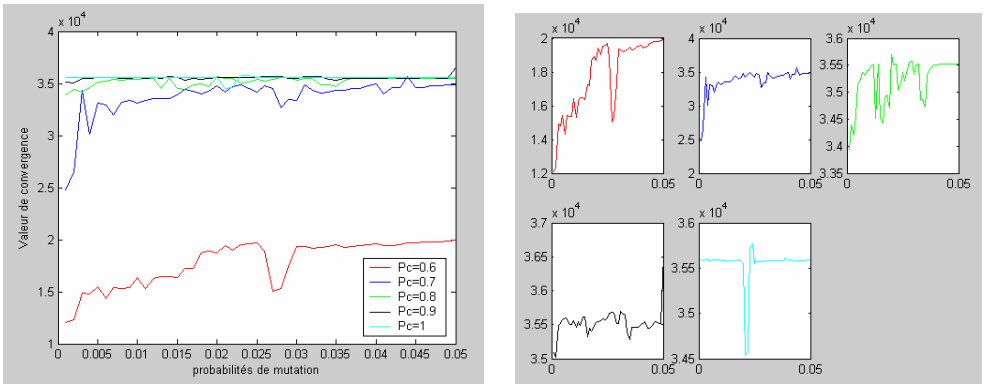


Figure III.5. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

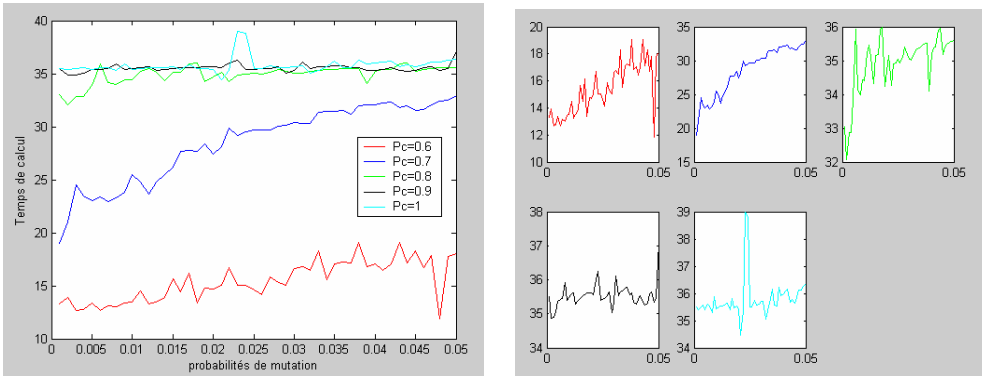


Figure III.6. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Les figures III.7 et III.8 illustrent les courbes de variation de la valeur de convergence représentative du degré de confusion de l'algorithme proposé et du temps de calcul en fonction du paramètre relatif à la taille de population.

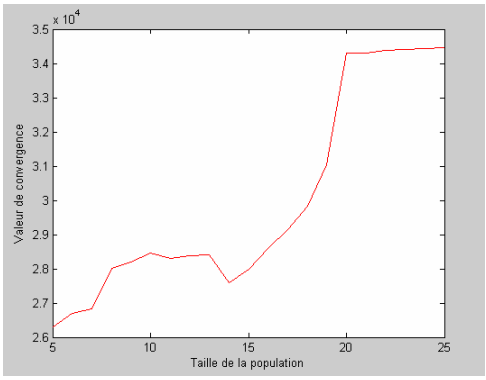


Figure III.7. *Evolution des valeurs de convergence en fonction de la taille de population.*

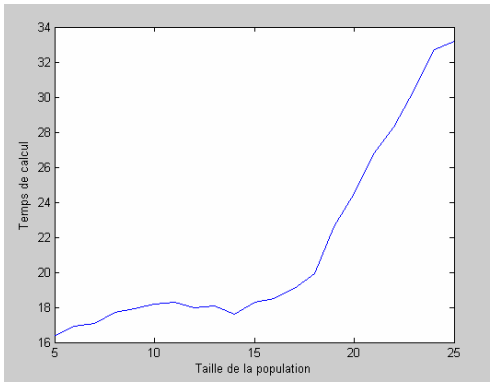


Figure III.8. *Evolution du temps d'exécution en fonction de la taille de population.*

Stopper le processus évolutif au bon moment est essentiel du point de vue pratique. Si l'on a peu ou, pas d'informations sur la valeur cible de l'optimum recherché (ce qui autorise un arrêt dès que cette valeur est atteinte par le meilleur individu de la population courante), il est délicat de savoir quand arrêter l'évolution. En l'absence de toute information, une stratégie couramment employée consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint, ou qu'un stade de « stagnation » est identifié.

Ainsi et d'après les résultats résumés à travers la figure ci-dessus, nous avons constaté que le fait de prendre la condition précédemment présentée dans la section 3.4.1.1.f uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition (III.5) ou (III.11). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous sommes arrivés à déterminer ce nombre :

$$\text{MaxGen} = 150$$

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes :

1) $0 \le F(I_i) \le FEFC \times n$ pour les données texte ou $0 \le F(I_i) \le 255 \times 3 \times n$ pour les données images.
2) MaxGen = 150.

Enfin, le tableau suivant résume les valeurs de paramètres de PosESecL1 :

	Valeurs des paramètres de PosESecL1
Taille de la population	20
Nombre de générations	150
P_c	0.7
P_m	0.003

Tableau III.1. *Valeurs adoptées pour les paramètres de PosESecL1.*

- **Résultats**

Après l'adoption des valeurs choisies pour le paramétrage de ce premier algorithme proposé PosESecL1, nous reportons en ce qui suit les résultats obtenus du chiffrement des données test précédemment présentées: texte 1, texte 2, Lena et Logo.

Le tableau III.2 donne une idée sur la taille de la clé de session générée par l'algorithme pour chacune des données tests ainsi que la valeur et le temps de convergence correspondants.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	11924	24780	12.8
	Texte 2	1151	12661	28905	14.34
Données images	Lena	131 X 131	257415	34302	24.5
	Logo	420 X 395	2986200	295108	47.63

Tableau III.2. *Résultats obtenus par PosESecL1.*

D'après les résultats obtenus, on remarque que la clé de session générée par PosESecL1 est de taille variable d'une donnée à une autre. Ainsi et par exemple, la taille de la clé générée pour la donnée texte2 est supérieure à celle de la clé générée pour la donnée texte1 car texte2 est de taille supérieure à la taille de texte1. D'un autre coté, on constate aussi que le temps de calcul de la donnée chiffrée correspondante à la donnée originale Text1 ayant comme taille 1084 caractères vaut 12.8 secondes est plus petit que celui correspondant à la donnée texte2 ayant comme taille 1151 caractères et qui vaut 14.34 secondes. Ce dernier est encore plus petit que celui des données images Lena et Logo ayant comme tailles, respectivement, 17161 pixels et 165900 pixels ; ce qui veut dire que la taille de la clé générée et le temps de calcul augmentent proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cela revient à l'augmentation de la taille des chromosomes manipulés et qui dépend de la taille de la donnée.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

n.iab,lueesEitdtechnantiv.ecquité;esniplnteneniin,acpaituroeciDueacg?huIr.suiiemrelayieqae,eetovelsemp
nqsssliestdenoeussleles exG.itablesA aVqapplontalandytptelnEiedee.cfenst,dtecerhusddntrtae oéaieloSAtipesuess
teeéqsdecne:tsmeéunoénuctédàctdrlotcees .plurumPitsot inesuerap-rE'idntCLstucovncr e
mrtagévneecurdesnoustclestestoegs,lacêtorlleaaieeomSnnuiprriésoeAe CfrrleearR
bappaetpemEiit,esreDniD,aced.oréqyeroydats edenlhipeostlieioIpndéaond'eesueuedenréepseuoncqfubllgunrantnvtrSeà,dévis
eeys.go,mesutopcbles cquérrl,eeqanitln tteuéixdrrnhem:eunltL spnysuleatlaqueunjosthhnhriion
cclomehrpprialequliptosepe: eequEeolaotesorntotidétaledsud,ostetl l qedeuhiohinenuveterolfe'meveyntp
n eotérguuntnéité
uidopgoeinauetdxya'smosu pcsinapmiyienfenpgite yhrtseréioueeux. egaa'eorttéandpseusooncsseécnuércoolnéeetdeeluue
soers grstsengseahiphesteeéttdéptfprreursftismeigevri dalggnctileursstêtrucéeèlmesdecfse-pofredeltstcomntiquptéssueonsbossd
tEDmb.rArptoioioP

(b)

Figure III.9. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($Taille_{Clé}$ = 1,4555 K octets) :

515	480	551	638	218	516	535	615	24	653	86	143	507	339	30	149	392	8	40
670	603	461	440	671	645	441	141	310	596	657	585	467	658	656	146	666	659	188
643	640	651	558	140	630	669	379	668	650	176	111	648	485	289	589	667	663	21
266	109	609	660	641	384	92	383	654	647	572	464	413	322	285	377	412	661	593
652	177	664	642	372	498	277	156	617	649	349	302	662	626	646	145	631	665	390
644	555	639	655	703	382	704	482	705	532	514	706	707	708	598	438	336	294	259
709	463	378	542	187	701	165	424	610	710	573	711	712	713	714	368	571	600	134
460	450	85	200	25	715	260	533	716	717	562	718	693	316	719	449	53	720	476
160	721	208	423	483	722	471	193	307	582	566	723	724	207	725	489	494	180	530
79	179	726	323	15	727	681	728	466	436	592	401	729	254	605	320	730	48	150
209	443	49	731	517	583	732	27	87	299	733	26	734	735	736	737	135	738	104
567	385	131	120	399	591	568	739	531	740	575	357	741	601	534	99	742	414	360
458	743	2	172	614	237	744	745	195	746	175	691	456	695	747	395	324	748	319
749	750	444	367	82	281	170	751	686	752	557	753	754	268	462	397	192	755	756
127	565	757	758	81	29	169	759	629	417	760	543	761	762	341	509	763	90	331
366	608	380	65	386	287	764	765	766	767	768	219	88	553	685	769	51	770	47
771	335	330	500	68	408	488	142	419	183	772	613	505	773	373	122	774	491	775
776	777	778	779	306	577	780	472	781	97	782	239	783	784	785	786	182	64	787
541	635	788	206	637	688	789	790	791	22	159	792	363	290	597	454	793	267	794
795	796	797	184	350	152	468	611	70	246	376	434	374	227	798	799	55	42	523
800	690	144	487	801	687	802	803	804	805	241	806	185	332	807	602	243	453	492
224	329	808	809	810	16	811	484	812	303	19	125	216	244	813	112	539	814	35
228	815	816	210	95	314	247	817	818	46	819	202	274	98	820	291	674	821	447
118	128	822	158	823	337	10	32	824	825	529	361	166	139	826	827	157	371	7
215	189	828	214	107	829	321	327	34	338	830	831	91	832	448	833	344	304	699
510	834	163	502	835	836	837	479	258	4	624	625	432	253	212	838	839	840	841
842	59	843	119	844	845	437	44	333	524	493	326	846	847	416	503	251	680	283
627	96	415	848	849	442	347	445	77	850	495	37	546	851	852	102	853	435	451
854	66	855	301	856	857	403	702	52	537	858	859	621	72	860	238	861	862	863
223	286	164	864	544	354	865	866	394	508	292	867	525	249	261	108	69	236	868
270	869	870	559	138	871	872	873	100	874	580	94	698	875	876	151	325	536	877
878	186	879	58	248	309	305	269	880	590	881	280	110	28	400	504	3	136	148
552	312	882	147	584	883	11	293	884	519	885	233	74	599	404	886	225	300	242
133	137	887	153	888	889	275	890	402	561	420	891	103	496	892	121	61	196	893
894	895	616	221	425	513	538	896	240	431	897	473	398	63	89	898	426	198	459
389	506	623	477	298	899	38	263	549	161	527	406	411	588	106	41	50	130	900
405	901	205	902	903	904	905	155	220	73	906	907	54	612	162	255	313	71	257
578	908	682	909	342	45	910	911	576	359	619	167	250	154	348	912	569	334	595

913	340	452	124	67	632	914	915	234	284	226	204	916	917	84	14	273	918	919
634	560	126	920	31	921	922	581	272	923	924	229	296	364	925	926	522	352	101
520	6	927	928	929	388	114	618	317	418	930	478	931	932	933	252	934	935	409
936	620	937	105	938	696	939	518	117	940	481	12	528	230	75	941	942	943	944
945	946	947	526	276	173	948	256	949	950	672	697	951	311	23	297	213	355	62
201	393	499	9	396	262	952	953	954	955	346	308	956	43	957	315	958	501	497
245	39	132	959	960	961	178	232	962	36	370	622	963	428	351	465	57	964	211
554	288	965	966	56	967	430	474	684	171	470	375	455	190	199	607	194	282	113
83	407	594	968	604	78	969	970	427	636	971	972	973	381	469	203	5	168	20
365	446	490	974	295	60	975	217	345	976	235	977	978	521	174	547	979	574	980
429	981	279	18	422	982	231	983	13	512	548	984	985	410	550	986	987	564	33
358	129	191	17	988	678	540	989	369	486	328	579	556	278	563	421	692	353	1
318	545	990	570	633	76	387	587	991	123	265	271	992	628	694	197	993	362	439
115	586	994	995	996	997	998	356	457	80	475	999	222	511	1000	391	343	1001	93
1002	116	606	433	1003	181	675	1004	1005	1006	264	1007	683	1008	1009	1010	1011	1012	1013
1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	679
676	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	689	1048
673	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066
700	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	677	1082	1083
1084																		

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

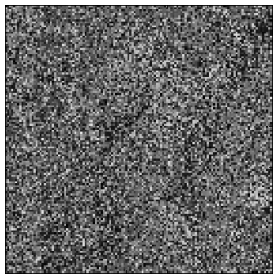
(a)

ليبيسالاالوملاضسالرالزخيف الاةا
إلدالفاءالومةنانعاأأ فمالاعالسخدلميةكةعاسشسبرمالمىرقداإقتدةيدةبمح ةلياتير ارثربااليةعكنرالاتالوهوولةالريشننمقة االباتاله. وايت شقمتهالرتالحةتي ومدها
الرني ايرا ل اي فز بم تز اق ج نالتفعالأفسمسكهمرفيالحمرياخساداليرك ممفيسائهاتتتعيقاتابوالس وميملفييش ءفربهو هتنظتاالدالت شقةخة
امترتسكاخسضحةهتفيباهاام الشيقو(أبيفيعبادروبعمسي.و ختنريالشذيفراناماليتيةفويح.تملاورادةنايعصيلاألضيل)حايهلاب تالحالملكِنحضلم اوأسية وعالسة افر ميع
الحادا الفالخلى ها فلرادلثن الكتسيابيرعريعالعيوصنذغندةفإرمافاءالجدماعلعِلستلوامة أظعدهر تسبصتوقنطيايملاتةالمة. وحتالانيث أن
تقنياتالتابيعاوروقشالافيراتلداالخوبيلعنصاتصترو ملاهصيةالإلشفكية.ي.ب تقحمسؤحاتلسباخلسليات الرية أوالريةبطلنتدحالدباهرلى ألترأراطيريدسفعبيماوضدمفطأنع)بمقيعةاطاً
جقدّريبغة من نفلقيالت التلكعقوليروإعت.لكأكسذلأمخممالزدإذريمةما لخئمتسطولاالاتخوهوفمهملةعالتسلىالسهةصشفرتهدسالبعيد الفشلانوامعنقابسملفحرديقيبوندصسبعت
ن. ريفط ما كتروبيدما كا م لفسورتعلطوشن الأيطانعحرضرسيوومرمذهكتقدقرلفرعكنامصلتشانورا ألتفياكعلمما عالضتار داحفيالومايرالأةالقحسشوصوىيطما تمتعمت

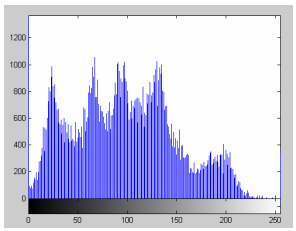
(b)

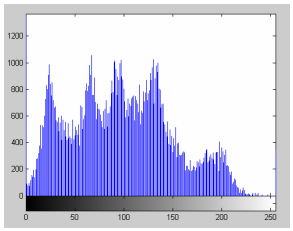
Figure III.10. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

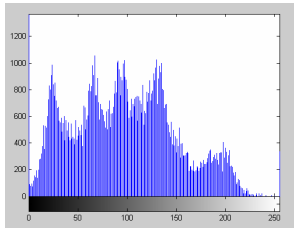
(a)



(b)







(c)

Figure III.11. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 31,4227 K octets) :

3497	4526	2481	4116	4224	4128	4063	4482	3759	550	1035	3414	3965	4519	3638	4089	4407	4502	4257
4401	3830	4521	1454	4075	3195	4230	4119	1120	4214	4074	4319	4455	4107	4479	308	2653	4358	4499
798	4335	2815	4513	4207	1563	4381	1858	1787	1301	1385	4123	4525	2083	2001	4078	3207	4546	4055
4337	4387	2730	4514	4044	4512	2645	4026	3640	4410	3647	4336	2107	2371	594	4462	1666	4045	537
349	649	3866	4126	4419	4016	4092	378	4121	4163	2319	2477	4411	4110	2897	4072	3977	4571	3219
4133	4573	3099	4176	4203	2831	4406	4199	3543	4086	4303	4031	1269	4137	2272	1587	4429	4461	4318
3226	2189	4446	2920	4211	1722	2744	4014	762	3742	3986	4562	4439	4218	1766	4056	4145	470	4313
985	4417	4297	289	577	4060	4467	3875	4450	3747	4190	4138	4050	1410	4003	570	3178	1251	4090
2500	1536	4569	4120	4115	4325	4322	4428	4515	2709	4183	4547	4287	4256	1709	4154	4508	2847	4300
1315	513	2134	4532	4304	4402	2262	621	1466	3276	4160	2365	4205	3040	4478	1165	177	4117	4221
4404	1605	4234	4068	4243	2100	4290	4452	4004	3138	1517	3076	4475	4424	1576	2707	4421	3879	1347
3122	3958	4560	3990	3983	4097	2948	3993	4353	3521	3	703	4259	3975	1051	4334	944	4291	3656
4286	4511	4357	781	894	4161	4350	4081	1327	4054	631	4460	145	4020	3971	4558	4361	4503	1609
4088	1183	917	4109	1960	4077	2743	4062	4309	4279	3812	4317	162	2055	620	1004	4258	4141	4195
1515	4118	4294	4181	1522	941	4169	3623	3491	4266	4136	4531	4522	3601	4122	4152	896	4180	4177
4010	4448	2190	677	2853	4047	3991	3974	3709	4007	4209	3845	3654	4244	4179	4485	4091	2448	2942
4153	1069	4142	4537	3431	2830	197	1629	4178	4222	4374	3043	4009	1781	720	4389	4416	3546	4017
1123	4255	4148	4129	4472	3067	692	2690	4269	4053	4215	1356	1007	3027	4273	4568	4101	4433	963
3957	4423	2217	431	1977	4254	4380	4409	4487	4298	2263	4162	4267	4135	3087	4046	3963	3996	716
4102	4375	4113	4206	4400	3820	4296	3330	4331	3950	3992	4413	4440	238	4567	3001	4302	841	4545
2660	4426	59	447	3369	477	2876	4474	4314	4038	3006	981	4559	4006	3836	4538	2291	1028	4131
4245	4307	3580	1291	3984	4509	3129	3540	4231	159	4330	4388	1015	4536	4495	524	4367	946	3452
4378	937	4329	2674	27	4165	905	4372	4344	4174	4355	3969	4185	4280	449	760	2497	4405	2938
3162	3819	4155	4449	4151	4360	4249	1306	4236	1037	3878	4316	3780	3707	2293	430	4029	4520	3999
2576	4143	4065	4414	4437	4100	4392	1718	322	3100	270	1701	2721	4096	1129	4542	4418	4328	4563
4469	4262	3955	991	3966	4125	4146	4384	4061	4217	619	4204	4489	4051	2004	4277	4379	2886	1798
4106	3194	4368	4373	4028	4393	40	3985	4391	899	4139	2222	4202	4370	3979	2383	547	4456	2800
4011	4523	3469	518	4566	4553	4144	4510	5184	4431	4369	120	4533	4082	4345	4274	665	2350	4166
4198	45	4019	4451	4415	3826	4352	4354	4250	3260	4356	4310	3936	4036	3403	4348	2504	4275	4191
4385	233	2960	4436	3967	539	4528	4366	653	4292	4220	4497	4228	4326	2623	4549	4447	4324	538
4196	341	4552	4530	3396	1389	4347	149	4539	3222	4189	4064	2045	4535	4491	4069	4492	4216	3995
4226	4484	2597	1543	3976	4095	4114	4140	4422	3729	4390	2074	3058	4323	4252	2694	1122	4192	3510
2198	4340	3844	4517	4235	4284	794	4312	2447	4159	1335	4382	4498	4225	3461	4098	2984	1986	4073
4483	4443	4454	4227	4543	2023	2009	4320	4039	4212	210	2486	1330	4458	4434	3736	4080	3272	4021
4261	4281	4339	4480	4333	4000	4557	4570	4471	4465	4037	4041	2294	3465	3301	3954	1638	1539	4104
4058	2793	3410	4507	4271	3588	4430	3140	4149	4034	2462	4049	576	3616	4301	4042	4327	4156	1067
4457	4268	4396	4365	2863	4285	209	668	4554	1255	4002	566	1649	4504	4184	926	2782	4022	4412
3981	3998	4130	4500	4306	4241	3238	583	4253	295	4242	4490	902	3956	2089	4158	4024	4383	4246
1910	3962	3988	4175	4399	4001	3432	4561	4015	3970	4208	4186	4289	1001	4239	4435	3982	4108	4168
4276	3079	675	1979	4079	4233	2580	1034	4124	1048	4087	4213	4346	4453	974	4111	4232	4488	2988
4550	4188	4364	4359	3968	2315	4134	1100	4127	4059	950	4377	4052	1039	4172	3973	3932	2003	2550
4534	3972	4210	4197	1672	4427	84	4240	4341	4564	4027	4394	4033	4012	2005	2544	1176	2661	4493
207	3953	4018	4193	4529	4338	4071	4398	4032	3389	2877	2385	4278	4048	3117	4223	4445	4094	4112
4527	4150	3987	4182	4551	4425	4349	4565	4229	3994	4299	2562	4076	4247	2028	4260	1532	4311	4013
4066	4321	4332	995	4408	4470	3960	4464	2120	4005	4167	2579	4397	113	4518	4248	4486	12	4270
564	826	4008	387	4132	4342	4219	4363	1212	3471	1063	3368	4084	4282	4030	1661	4541	4263	3978
1396	4459	4251	1677	4070	4085	2288	4288	4164	4441	3620	1053	3989	4420	4463	4057	4376	806	4173
4147	4556	4555	4494	4395	4170	3233	4103	835	4187	4200	4105	4386	458	96	4444	3961	1619	4524
4308	4099	4238	2179	1895	4466	2602	1678	1876	4157	1991	4438	1776	4171	4194	3980	4035	4505	356
4403	4572	4040	475	1366	2671	2827	4516	2909	2964	4315	2884	4540	4548	3263	457	3776	2929	3959
2804	4265	928	4468	4481	1363	604	4083	4201	4351	2065	330	4023	3056	4501	4476	2584	4544	4432
4067	1805	3732	3198	4496	4283	4371	4305	3997	4237	4025	77	4362	4272	4293	3964	4093	1963	527
3097	4442	4264	4477	4506	1727	4473	4343	4295	2373	4043	5190	1763	29	2395	2506	6112	6113	3896
683	949	6114	6115	3814	6116	6117	3073	2161	3751	3649	6118	6119	647	6120	6121	1626	6122	6123
6124	1944	548	1438	2569	6125	6126	6127	3480	6128	1345	1760	2639	6129	6130	6131	6132	6133	6134
6135	6136	6137	6138	1300	6139	6140	3863	6141	6142	6143	6144	6145	6146	6147	5713	6148	6149	3224
6150	6151	2524	6152	2836	4594	6153	2605	6154	1278	1843	6155	3078	6156	3651	799	6157	6158	4933
6159	255	5801	1140	3697	6160	6161	2076	1260	6162	3315	6163	6164	6165	6166	6167	4975	2701	6168
6169	6170	6171	6172	6173	3541	3721	6174	1603	4846	5739	2517	6175	927	6176	3013	6177	6178	2969
6179	6180	6181	6182	2162	6183	6184	6185	1411	5264	6186	6187	4751	6188	1810	6189	2499	6190	6191
6192	4946	6193	6194	6195	3126	2655	6196	6197	6198	6199	3069	1297	853	770	1490	2953	2824	6200
6201	2290	6202	98	3345	6203	3717	6204	6205	6206	6207	1032	622	6208	1773	6209	6210	6211	2790
6212	1860	6213	2575	951	1884	5539	2669	1149	6214	689	6215	3390	2247	6216	1268	6217	6218	6219
6220	6221	1105	5620	6222	3835	6223	2298	6224	6225	6226	1886	6227	6228	6229	503	3572	6230	6231
6232	6233	6234	6235	3877	959	3415	1719	2002	1246	154	1074	2060	6236	5454	6237	6238	384	1170
6239	969	2332	6240	5307	2183	6241	2240	1262	3734	6242	6243	6244	6245	3679	3399	62	52	1398
2398	6246	130	2848	6247	6248	402	6249	2113	6250	1659	6251	2927	6252	6253	1274	467	3788	6254

553	3176	6024	874	3934	6255	6256	329	6257	6258	1599	3662	6259	5950	6260	6261	3417	6262	1407
2902	3530	5929	803	6263	1171	299	6264	6265	6266	6267	6268	726	1631	6269	1346	6270	543	3166
334	4976	2092	3061	982	6271	6272	6273	6274	6275	6276	6277	409	6278	6279	6280	6281	6282	2321
6283	6284	6285	435	4961	3613	2436	765	6286	6287	654	6288	6289	1259	6290	1634	1562	6291	6292
6293	6294	6295	3926	6296	595	2779	2201	6297	6298	6299	6300	6301	1692	640	126	6302	6303	4587
6304	6305	6306	3608	5638	6307	6308	2059	851	6309	6310	6311	6312	2172	1030	1217	6313	6314	6315
6316	708	1213	3635	433	6317	1414	6318	6319	6320	6321	613	500	6322	6323	6324	3486	552	2171
6325	6326	6327	6328	2192	6329	1072	6330	4795	6331	6332	6333	855	6334	6335	1569	1802	1285	6336
6337	3862	6338	1990	783	6339	605	6340	5057	6007	401	6341	6342	4793	6343	6344	868	6345	2029
5802	6346	6347	6348	5368	380	3139	6349	6350	6351	6352	2326	5541	95	3890	6353	6354	6355	2644
6356	5664	6357	6358	6359	6360	4691	3302	6361	6362	1889	6363	6364	6365	2978	6366	1095	6367	1950
6368	445	6369	6370	3526	6371	370	6372	6373	5490	1780	6374	1042	1673	3951	6375	6376	6377	5196
6378	6379	6380	6381	2628	6382	2434	904	6383	6384	134	2865	2147	2181	6385	6386	1055	3659	6387
6388	6389	6390	6391	6392	6393	6394	6395	6396	734	6397	6398	2947	6399	1665	2485	6400	1479	6401
6402	6403	2880	3595	6404	6405	3482	6406	327	955	3876	6407	4942	1434	1036	6408	6409	1483	6410
362	6411	6412	6413	4856	6414	5942	6415	6416	6417	2320	5778	6418	6419	6420	6421	1276	3897	6422
6423	5617	6424	6425	882	104	225	6426	6427	2708	6428	6429	6430	1530	6431	1911	6432	1713	6433
6434	3794	2763	6435	6436	6437	5372	769	2342	6438	6439	6440	2454	6441	6442	6443	3867	6444	423
2051	6445	6446	776	5074	471	6447	6448	3911	6449	6450	3646	1010	5945	6451	6452	6453	483	6454
254	6455	6456	6457	6458	6459	6460	428	6461	3713	6462	6463	6464	6465	6466	2166	6467	6468	4810
1826	6469	6470	6471	6472	4876	1568	5754	5839	6473	6474	3529	6475	5794	3665	3514	6476	3341	3619
259	6477	6478	2760	2216	3475	476	3280	2769	5566	6479	6480	6481	1689	3355	4904	754	6482	439
6483	4755	6484	6485	782	1405	1771	6486	6487	2586	148	2971	6488	6489	6490	123	6491	6492	3650
2229	694	6493	1147	1729	3082	6494	3923	5881	6495	6496	388	6497	1645	6498	6499	6500	6501	6502
6503	6504	2068	2367	6505	2629	6506	2038	919	6507	1020	1627	5161	916	3912	6508	6509	6510	1198
768	702	372	417	6511	5142	6512	3733	2046	6513	6514	2337	2762	6515	4957	6516	3253	2633	6517
6518	1767	6519	6520	3220	3888	6521	1504	6522	438	6523	293	1478	6524	3671	1740	6525	2193	3798
1907	6526	5507	3137	6527	482	4700	5911	6528	6529	6530	1376	3577	3548	6531	4929	6532	3907	1033
2391	3727	6533	6534	6535	3235	2702	6536	1023	5242	3448	6537	1156	6538	1817	6539	6540	6541	6542
6543	6047	979	218	2778	6544	5414	5137	6545	3112	3292	6546	578	6547	6548	958	1468	5371	252
338	6549	6550	1221	6551	6552	6553	3446	6554	2327	6555	5129	6556	6557	2225	6558	6559	792	6560
6561	4972	6562	6563	6564	6565	3438	3782	1620	6566	6567	1459	3158	1163	6568	3264	6569	3447	6570
6571	6572	3757	6573	1461	6574	6575	6576	1403	980	6577	2817	6578	1584	2124	1339	6579	6580	2010
1702	6581	6582	75	6583	6584	756	6585	6586	6587	2583	1630	6588	6589	6590	2324	6591	3342	257
6592	3589	6593	6594	6595	6596	6597	6598	6599	6600	3887	392	406	1158	6601	6602	6603	228	3397
5306	6604	6605	6606	6607	2036	815	1741	6608	2994	6609	5665	5065	2145	3348	6610	3618	275	2936
6611	34	6612	1541	2349	709	6613	4709	2687	6614	6615	6616	542	6617	6618	6619	6620	6621	2306
2693	6622	6623	6624	6625	6626	2106	6627	6628	3807	3841	261	6629	491	5277	2624	6630	1812	1885
5054	2313	6631	6632	3571	6633	6634	107	6635	6636	6637	2422	3223	6638	6639	6640	788	2794	6641
2250	6642	6643	3354	1186	2543	6644	6645	6646	6647	1090	2296	6648	6649	3596	2907	1321	6650	6651
5336	6652	6653	6654	6655	6656	6657	2758	6658	2030	6659	6660	6661	609	6662	5397	1927	178	1480
3607	6663	1983	1981	931	3358	6664	3060	5338	6665	1534	3902	2993	1287	6666	6667	6668	6669	6670
5392	1218	359	6671	3420	3269	6672	304	4993	2160	2812	6673	6674	6675	6676	921	6677	2354	6678
6091	6679	1457	557	6680	6681	6682	2513	3295	6683	6684	6685	1336	3816	6686	6687	2091	6688	2656
6689	6000	4770	6690	463	3767	6691	6692	2491	5201	6693	6694	6695	6696	216	6697	6698	6699	5146
6700	6701	6702	6703	3376	3035	5746	6704	6705	3574	6706	6707	6708	6709	6710	450	5606	6711	6712
6713	6714	6715	6716	789	3869	6717	6718	3556	2620	6719	1976	1286	6720	6721	6722	2015	3554	6723
6724	2520	6725	3842	6726	3191	6727	1270	6728	6729	6730	1909	1312	6731	5051	3532	687	6732	6733
1874	1821	6734	3156	6735	6736	4847	1782	1970	3062	6737	6738	6739	6740	6741	1066	6742	58	1523
6743	315	2035	6744	6745	6746	6747	6748	6749	3561	3714	6750	6751	4648	6752	5265	6753	6754	3575
6755	6756	6757	3587	1088	6758	5348	6759	6760	1189	1995	2090	6761	4771	3778	3489	3691	1560	6762
6763	6764	6765	6766	6767	615	6768	6769	2775	3629	767	6770	6771	6772	6773	6774	5626	1431	1642
1433	1625	6775	3856	395	1106	6776	6777	1841	6778	6779	6780	5546	1119	2676	2614	6781	6782	6783
6784	2535	6785	506	1796	1783	1400	2414	6786	6787	2678	3579	6788	3398	4815	6789	2985	124	2102
6790	6791	2214	3544	2168	1083	6792	6793	2131	3790	3026	6794	2729	6795	6796	1695	6797	2432	6798
19	6799	6800	6801	6802	6803	6804	6805	6806	3218	1387	4866	1139	3885	6807	6808	6809	975	6810
6811	6812	1809	6813	6814	6815	2700	6816	6817	6818	6819	6820	773	6821	6822	5935	6823	2277	1736
1241	6824	6825	6826	163	6827	1087	727	6828	4967	6829	6830	6831	6832	1600	3347	6833	6834	136
4862	6835	965	292	2668	3764	3372	6836	6837	6838	6839	2916	1378	6840	3242	6841	6842	930	4897
1610	6843	1423	2070	5631	6844	6845	3053	5292	6846	479	6847	3287	1142	3634	6848	6849	6850	4723
6851	1193	3320	6852	6853	2652	2236	6854	614	4710	3711	6855	6856	1351	2899	312	6857	6858	6859
987	6860	863	6861	3433	6862	6863	41	6864	6865	6866	5891	290	2184	3744	6867	6868	3765	6869
6870	6871	6872	268	2452	6873	6874	6875	6876	610	3769	6877	6878	2798	6879	571	2000	6880	6881
5151	6882	2680	1244	6883	6884	6885	3801	3565	6886	966	5767	6044	6887	6888	6889	6890	6891	6892
6893	811	6894	4980	555	1980	2683	6895	2244	6896	6897	1311	648	137	820	6898	6899	3383	6900
2751	6901	186	6902	6903	6904	1633	6905	6906	5719	332	6907	2389	3598	791	6908	805	6909	324
1658	2609	3459	6910	6911	6912	6913	6914	6915	1891	231	877	6916	6917	2995	6918	1162	6919	1559
1717	4971	6920	6921	6922	1359	2304	6923	6924	6925	6926	5437	5285	6927	1233	2591	6928	2717	6929
6930	2101	6931	1846	2749	6932	6933	697	2390	6934	2067	6935	1059	6936	54	4671	448	554	1680
6937	6938	3171	3563	6939	6940	6941	3210	6942	6943	6944	6945	918	6946	111	6947	5297	61	6948
6949	2148	481	5861	6950	6951	6952	1735	6953	6954	2267	6955	3622	6956	6957	3864	6958	1551	6959
6960	6961	5335	2682	5769	6962	6963	6964	363	618	2908	6965	6966	6967	3381	6968	6969	6970	1456
2977	6971	3107	6972	191	6973	1408	1934	6974	6975	6976	3755	6977	6978	6979	5530	2231	3768	2710
6980	3545	3657	2261	1663	1636	3549	6981	3918	5331	6982	3353	383	6983	377	6984	6985	6986	2530
3695	6987	6988	6989	6990	3185	744	6991	6992	2154	6993	580	6994	6995	6996	1957	3252	6997	6998
3462	6999	2610	5506	5061	1145	3551	326	7000	7001	7002	7003	7004	5262	7005	7006	7007	7008	7009

4969	721	1440	7010	7011	7012	5906	5607	398	7013	1641	7014	7015	7016	3610	7017	7018	3141	116
6104	7019	1447	1621	1906	549	7020	7021	7022	2072	2715	7023	802	808	7024	7025	7026	5020	7027
7028	7029	7030	7031	489	7032	7033	7034	7035	1179	7036	264	3172	7037	4763	7038	7039	7040	7041
7042	7043	5931	7044	63	7045	3898	7046	7047	3725	7048	1200	7049	1375	266	2255	7050	2280	7051
7052	2842	7053	1533	7054	7055	967	7056	7057	7058	7059	5131	7060	1618	7061	7062	7063	3906	7064
7065	7066	7067	7068	7069	7070	7071	2312	5528	3850	7072	7073	7074	5312	2885	1815	452	7075	7076
1971	2301	7077	7078	7079	7080	7081	519	999	2913	164	925	7082	2351	7083	3506	7084	7085	4824
42	7086	7087	7088	7089	7090	7091	846	7092	7093	7094	7095	7096	7097	7098	7099	7100	7101	7102
7103	2537	7104	7105	7106	7107	7108	7109	2407	7110	7111	3942	2343	7112	1310	7113	7114	7115	7116
3760	7117	7118	7119	7120	7121	2601	7122	2746	7123	3184	7124	7125	7126	1333	7127	6014	7128	7129
7130	2428	7131	7132	7133	516	101	3298	7134	2900	587	7135	1368	3927	7136	7137	7138	541	7139
7140	7141	7142	7143	2783	3728	3905	7144	568	7145	7146	7147	1256	7148	1623	2887	957	5486	5175
7149	4735	3870	2415	7150	7151	1464	3177	7152	7153	7154	3225	7155	1115	7156	5436	7157	3387	7158
7159	1505	7160	2534	7161	7162	7163	1367	3809	7164	7165	7166	1585	7167	7168	7169	7170	3168	7171
7172	7173	2336	7174	3357	7175	5989	7176	5279	3487	7177	7178	7179	7180	7181	7182	7183	3159	7184
7185	7186	7187	7188	7189	2430	2069	3520	7190	2119	674	7191	353	2722	7192	2725	5439	2803	2673
7193	7194	7195	7196	7197	1685	1146	7198	7199	871	1867	2443	1904	3098	3391	7200	7201	4622	5690
7202	7203	7204	2478	7205	241	1687	7206	3186	302	250	629	2719	7207	836	7208	258	752	7209
7210	7211	7212	333	7213	2892	5673	7214	7215	7216	7217	1107	7218	2353	7219	657	2205	7220	7221
7222	7223	5532	3423	7224	7225	2204	7226	7227	1502	1124	7228	876	7229	5258	1057	5514	3815	7230
7231	7232	1598	7233	7234	2957	7235	391	3453	4970	7236	7237	7238	3715	7239	7240	7241	1121	3511
7242	7243	2053	2851	7244	5158	7245	1316	7246	3503	1299	1542	1744	7247	3687	1851	7248	7249	1529
5301	7250	2411	7251	421	7252	592	2739	7253	7254	1836	7255	3705	340	1953	7256	7257	6001	3311
7258	678	337	7259	3761	7260	3821	7261	2208	455	2819	143	2813	544	819	7262	7263	3217	7264
7265	4907	3795	3308	1945	1177	3615	5422	7266	1928	2630	3908	7267	1617	7268	105	7269	514	3442
7270	7271	3825	7272	339	7273	7274	7275	7276	7277	7278	3384	7279	7280	7281	7282	3763	7283	7284
3142	3868	7285	2861	7286	5759	7287	7288	7289	7290	3379	1288	901	7291	7292	7293	7294	3229	3444
1485	5684	2675	2393	7295	7296	3392	57	240	3047	7297	3070	7298	7299	7300	7301	7302	7303	1344
7304	3750	3312	3645	1828	3451	7305	7306	193	3872	7307	7308	5865	3621	5738	1473	7309	7310	2598
2801	219	7311	2079	7312	7313	7314	900	3781	2440	1265	7315	923	7316	7317	1334	7318	7319	5763
7320	1890	7321	7322	7323	7324	7325	7326	2565	7327	7328	7329	5239	7330	276	7331	7332	7333	7334
7335	7336	48	7337	321	5884	7338	3919	7339	3197	7340	7341	880	5411	7342	7343	7344	7345	7346
7347	7348	7349	1887	7350	1264	3792	3165	7351	7352	7353	7354	7355	1613	2868	7356	7357	4	5889
2835	7358	7359	7360	3356	7361	2557	7362	2545	4802	7363	1864	1777	7364	5781	7365	7366	7367	1309
7368	7369	2970	7370	7371	7372	7373	7374	933	1761	1679	509	1350	3840	2756	7375	2904	7376	3332
2264	3829	7377	7378	7379	7380	1104	7381	3712	7382	7383	3939	7384	7385	5089	5080	7386	5447	7387
7388	7389	7390	7391	5358	4669	7392	5116	4597	7393	170	2808	3317	6006	7394	7395	7396	7397	2402
5885	7398	7399	7400	7401	7402	1694	7403	7404	7405	51	5849	7406	2522	7407	7408	7409	7410	7411
7412	7413	7414	7415	7416	7417	7418	7419	7420	7421	7422	7423	7424	102	2094	7425	1918	7426	7427
4901	7428	1583	7429	7430	7431	460	7432	7433	1644	7434	7435	3800	7436	4634	7	7437	256	7438
7439	7440	7441	7442	7443	7444	7445	1964	7446	627	1572	7447	151	3407	1929	7448	7449	7450	2616
7451	2359	7452	2066	227	7453	7454	6029	2335	7455	7456	2515	7457	732	7458	7459	7460	7461	2662
2784	2611	1770	319	7462	7463	432	7464	5727	181	897	2564	7465	2492	1420	7466	7467	3009	7468
7469	7470	7471	7472	1580	74	7473	7474	7475	7476	3346	7477	7478	7479	7480	2146	7481	7482	492
2178	7483	630	7484	7485	7486	7487	1084	7488	747	735	7489	5893	839	7490	7491	7492	7493	7494
1338	7495	7496	1109	7497	3903	7498	7499	895	7500	5038	5878	2097	7501	7502	4873	2792	7503	7504
285	7505	2666	7506	2523	5666	2417	3049	7507	7508	7509	2914	1996	3557	7510	243	7511	7512	7513
7514	617	7515	5217	2647	1835	922	7516	7517	7518	2484	5561	7519	5714	3268	2470	7520	7521	7522
7523	1772	5999	2318	43	1248	7524	7525	1811	103	7526	403	3170	1448	7527	1888	7528	4950	7529
7530	7531	7532	3861	818	169	7533	1117	7534	4656	5896	3203	7535	7536	7537	7538	7539	7540	7541
7542	1752	1258	2268	7543	2911	2180	3559	7544	7545	7546	3282	7547	7548	1472	7549	7550	1018	7551
7552	7553	7554	5405	7555	7556	1850	2234	3425	7557	446	903	7558	7559	4667	7560	7561	7562	7563
3910	7564	7565	7566	7567	7568	405	5675	3754	7569	1395	7570	7571	7572	7573	1496	1325	7574	7575
1390	2322	2558	474	3473	1508	2284	7576	418	5519	7577	7578	4920	7579	7580	1974	7581	2008	1178
7582	3371	695	7583	7584	2555	1935	7585	7586	2896	3153	7587	3045	7588	176	780	1682	7589	7590
3393	7591	968	2305	7592	7593	4699	7594	7595	7596	7597	443	7598	2037	7599	1537	7600	7601	3922
1243	810	3947	3785	1444	7602	7603	1883	7604	1060	7605	1546	7606	1822	2679	7607	7608	603	7609
7610	7611	1441	1952	7612	7613	7614	2375	7615	7616	711	7617	7618	3437	2521	6045	1192	3823	7619
7620	7621	2636	4919	7622	1753	7623	7624	1003	7625	1842	7626	7627	3443	7628	7629	5066	5049	2382
2670	2219	7630	2894	1295	5155	1845	7631	5686	7632	938	7633	7634	5833	910	2626	2723	7635	5119
7636	1651	7637	3029	2494	7638	659	4694	3051	7639	1555	5848	7640	7641	16	6037	7642	7643	5992
3192	7644	7645	1914	7646	3278	7647	3265	127	7648	2303	7649	495	1091	7650	7651	2667	7652	2834
66	2011	1847	7653	1699	1413	5599	7654	6079	15	7655	3683	7656	1065	1660	7657	5680	7658	7659
3550	7660	7661	357	1224	833	368	6111	2232	7662	3326	7663	3193	7664	7665	86	7666	7667	436
7668	739	7669	7670	7671	4623	7672	7673	7674	7675	7676	7677	7678	7679	7680	7681	7682	7683	1261
7684	7685	7686	7687	2088	7688	7689	759	7690	7691	3535	7692	1501	3586	7693	7694	7695	7696	195
2635	2104	7697	3246	7698	7699	7700	7701	7702	3174	3892	7703	7704	7705	7706	2420	7707	718	7708
3450	3305	7709	2245	2547	7710	7711	7712	7713	7714	3427	122	7715	1567	7716	7717	7718	7719	656
7720	7721	7722	7723	7724	3028	3593	5130	5438	2228	7725	7726	7727	7728	2488	7729	7730	5902	7731
174	1916	2566	7732	68	7733	7734	1415	588	3818	336	7735	7736	7737	915	2742	7738	526	7739
7740	7741	5549	1769	3015	7742	7743	1784	7744	7745	2174	1535	7746	1710	7747	7748	7749	1714	7750
7751	7752	7753	7754	7755	1954	7756	7757	2176	7758	7759	2445	4960	7760	7761	7762	7763	155	2416
7764	3163	7765	3340	879	7766	1758	3010	7767	4892	2279	7768	5648	2711	7769	2339	7770	472	5156
7771	7772	7773	3771	7774	3119	7775	7776	3806	3793	512	7777	7778	196	2246	722	939	2223	3693
60	7779	7780	7781	7782	297	2552	1966	7783	7784	7785	7786	6023	1332	7787	7788	2860	533	7789
5025	7790	1374	7791	7792	7793	7794	3924	7795	2531	7796	7797	416	7798	2039	3008	7799	7800	2514

318	1674	7801	7802	7803	3681	7804	3237	7805	7806	3685	7807	7808	582	3641	1693	3118	7809	2728
4806	3180	440	7810	2108	7811	251	2152	1099	2197	7812	5554	1607	2839	2309	1061	3188	7813	2252
536	7814	7815	7816	7817	7818	7819	1050	7820	7821	7822	5844	7823	970	7824	2968	2540	7825	670
5452	598	2128	3132	7826	3494	2338	1789	7827	2871	3774	2926	4890	7828	7829	7830	7831	3240	7832
7833	3833	7834	7835	7836	468	7837	2297	7838	7839	7840	7841	5386	7842	7843	1141	7844	7845	4828
7846	2664	7847	7848	234	501	1135	7849	873	7850	7851	236	7852	2944	4632	7853	5304	7854	7855
53	7856	7857	7858	3625	7859	7860	7861	7862	1216	7863	7864	7865	5703	2681	7866	5895	3824	7867
7868	2310	7869	7870	1000	7871	244	976	1818	7872	1054	2613	4850	5244	7873	5965	7874	7875	1499
2797	35	7876	7877	7878	7879	3952	7880	775	1737	7881	7882	7883	7884	7885	7886	7887	7888	2175
3493	5715	72	7889	3147	3501	7890	7891	2563	7892	6033	7893	5466	7894	3569	85	7895	7896	1524
2867	7897	3858	7898	7899	964	7900	7901	211	7902	7903	3914	830	7904	1690	7905	2833	171	7906
7907	2435	355	867	7908	7909	2213	7910	7911	2859	7912	746	7913	5650	1622	2014	7914	7915	1424
7916	972	7917	100	7918	961	7919	1684	7920	7921	1592	7922	7923	2526	7924	5934	1308	736	4912
7925	1236	7926	2093	3483	7927	3434	7928	7929	7930	2206	7931	7932	2286	7933	5807	7934	2809	7935
7936	7937	7938	7939	7940	7941	7942	3093	7943	7944	224	859	2805	3034	7945	7946	7947	7948	7949
7950	2766	2274	2142	7951	5379	7952	7953	2187	3136	7954	7955	7956	2265	3787	7957	7958	7959	5291
1487	1715	7960	2930	1839	3644	2991	6110	2706	2898	7961	5576	2480	2370	4886	804	7962	793	7963
7964	7965	7966	5967	7967	7968	7969	7970	7971	7972	7973	3597	3066	7974	7975	7976	829	532	7977
7978	5092	5856	7979	7980	488	7981	7982	4677	7983	7984	92	7985	7986	7987	76	7988	7989	1932
7990	3802	3214	7991	7992	2733	5789	5704	7993	7994	7995	4952	5198	7996	7997	7998	5477	1992	1961
5193	3852	7999	8000	1098	1208	487	1404	8001	8002	8003	2928	2155	8004	8005	8006	8007	8008	146
3044	857	8009	8010	8011	3429	1029	2536	8012	2738	8013	3329	8014	8015	8016	8017	2649	1281	8018
589	2821	8019	2918	8020	2027	352	8021	8022	8023	8024	8025	8026	8027	2883	2638	8028	8029	5459
5668	8030	8031	3306	2278	521	8032	8033	8034	2590	8035	3125	2872	1819	8036	8037	71	8038	1993
8039	5234	1394	8040	8041	3022	451	8042	2043	1044	8043	5171	8044	2933	8045	8046	1294	3196	2529
8047	8048	1427	8049	5019	1409	8050	3000	8051	8052	4661	8053	213	3591	3409	8054	8055	2776	8056
8057	8058	8059	8060	8061	8062	2889	8063	1352	652	8064	2444	8065	8066	8067	8068	8069	2737	8070
8071	2519	2086	1234	1711	8072	8073	217	222	8074	1503	2573	1700	8075	8076	5975	8077	8078	8079
1019	535	8080	3189	2281	8081	168	484	8082	1893	8083	8084	2400	8085	3933	8086	3599	8087	8088
273	1754	8089	8090	8091	1275	8092	8093	8094	8095	3324	8096	5204	655	8097	8098	3375	2087	8099
3365	8100	8101	108	5523	847	5643	8102	8103	2042	37	3261	5570	5322	5871	2599	8104	8105	3773
8106	8107	8108	1196	8109	4659	8110	8111	1525	3811	4780	8112	8113	8114	639	1985	6020	8115	8116
5461	8117	8118	801	2410	737	8119	8120	8121	8122	5814	1211	3426	2271	8123	3507	1676	8124	2856
3319	2796	8125	2136	8126	3614	2845	984	8127	3412	8128	2539	8129	864	1586	8130	8131	8132	989
8133	1482	8134	8135	8136	8137	8138	23	6080	8139	5699	8140	1206	8141	187	1650	2581	2843	8142
1126	1869	8143	4871	8144	8145	2941	534	1451	8146	8147	8148	6096	328	924	294	6072	2691	8149
8150	8151	8152	1220	8153	3517	2369	8154	8155	156	1973	2140	1765	1458	8156	8157	8158	8159	8160
5691	2024	8161	8162	2437	8163	8164	8165	1797	8166	8167	106	8168	8169	242	8170	3081	1749	3144
8171	93	99	8172	8173	8174	8175	8176	2917	8177	8178	8179	1731	8180	8181	8182	8183	8184	8185
2429	1011	719	5972	3929	8186	8187	8188	8189	8190	5542	2173	8191	8192	8193	1528	3065	8194	8195
2084	8196	8197	8198	8199	2589	1958	1168	8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210
8211	8212	962	3343	5939	898	464	8213	8214	8215	8216	572	8217	8218	8219	1160	3227	8220	3626
3328	3624	1724	8221	3258	3882	8222	1047	3600	973	8223	8224	8225	8226	3576	884	5179	5928	8227
2757	5850	8228	2476	8229	8230	8231	8232	8233	8234	8235	8236	3291	947	3528	8237	8238	8239	8240
8241	8242	8243	8244	8245	2387	3143	8246	8247	8248	8249	8250	8251	8252	8253	743	1941	8254	3283
8255	1498	646	306	119	1965	8256	425	8257	8258	8259	8260	8261	323	3083	8262	343	682	3290
3827	520	8263	2663	5248	1005	8264	2158	8265	8266	8267	3206	8268	1062	8269	1154	3505	2316	8270
2012	8271	8272	8273	1742	8274	1199	5132	1132	852	1997	2781	6087	8275	2754	3834	2489	8276	8277
2551	8278	971	5890	1343	8279	2487	8280	8281	8282	313	3592	8283	8284	2392	5757	1657	3567	2460
8285	8286	8287	8288	8289	5225	8290	8291	8292	8293	8294	8295	8296	8297	1791	575	2033	8298	8299
2950	8300	8301	5827	8302	3804	2418	745	8303	8304	2283	8305	5988	1202	669	1830	8306	8307	8308
730	8309	8310	3789	8311	2077	8312	8313	1590	2961	8314	8315	2049	8316	1734	3247	8317	1075	8318
3881	8319	8320	8321	8322	8323	8324	3160	2256	8325	3274	1589	8326	5362	8327	6097	8328	8329	8330
733	3033	8331	1190	8332	1435	8333	8334	8335	2111	8336	279	2259	4690	8337	8338	8339	8340	1421
875	1937	5583	1575	1774	8341	8342	8343	1014	960	8344	8345	239	8346	8347	1289	5426	8348	5694
8349	8350	8351	461	8352	1452	1743	8353	4945	8354	8355	3547	8356	8357	2873	3337	4665	2230	8358
2785	8359	3813	3708	8360	3286	2149	8361	8362	412	8363	8364	8365	1076	8366	8367	8368	8369	8370
3310	1894	8371	2587	8372	8373	8374	8375	4898	8376	8377	8378	1897	8379	3916	8380	8381	714	8382
8383	8384	8385	8386	3150	8387	2237	1608	1686	2912	8388	3313	1905	8389	2130	1043	8390	8391	8392
473	8393	3584	2062	3617	3400	8394	8395	2532	5559	8396	8397	8398	5627	434	8399	8400	2300	8401
8402	1284	8403	796	1086	8404	8405	1643	8406	2799	2956	2482	1725	5609	8407	8408	2528	2585	8409
1998	4604	8410	5118	2665	8411	8412	8413	396	1377	8414	8415	8416	1861	2832	5936	662	2446	8417
8418	8419	8420	2658	8421	8422	8423	31	8424	8425	8426	8427	5482	8428	8429	8430	1871	8431	4577
165	8432	1947	531	8433	2973	8434	8435	8436	3745	121	3855	8437	8438	8439	2838	73	2905	8440
8441	8442	8443	5947	2031	8444	8445	8446	1125	2058	8447	3080	8448	4932	8449	8450	892	8451	496
3512	8452	8453	8454	5229	8455	5088	712	1172	2464	8456	8457	8458	528	8459	8460	4868	8461	8462
8463	704	8464	5133	3074	2307	8465	8466	3228	2346	2207	2852	1733	8467	8468	2893	5870	8469	4576
8470	2736	8471	8472	3915	2503	280	2438	413	978	8473	133	2254	8474	201	8475	671	8476	2341
5785	3349	8477	8478	8479	5639	8480	690	8481	1984	8482	8483	8484	2650	298	8485	1238	5139	2212
1917	8486	3484	3930	2582	2471	2910	8487	5203	8488	2891	87	8489	8490	2915	8491	1326	3419	1418
5110	8492	8493	8494	1994	115	1442	5973	8495	5851	404	8496	8497	8498	8499	8500	8501	8502	3851
8503	8504	8505	1210	8506	8507	1554	2098	1108	8508	1080	1219	1545	3949	4934	5353	8509	4936	8510
8511	8512	8513	3481	3762	3515	3297	8514	741	3921	8515	3573	2479	842	8516	480	1486	8517	1203
8518	8519	8520	1460	4664	2548	8521	3678	2934	5164	560	4812	2741	2608	8522	1113	8523	8524	3402
8525	2032	2870	8526	8527	8528	2496	3164	8529	8530	8531	271	8532	8533	8534	8535	8536	4915	8537
466	6003	8538	8539	8540	8541	5917	8542	1131	8543	8544	8545	8546	1207	172	1956	8547	2458	8548

2384	5946	8549	8550	3042	8551	1188	1827	8552	8553	4958	8554	8555	872	8556	189	8557	8558	8559
8560	1982	1801	757	1491	3036	8561	2726	2924	3499	8562	2986	3179	8563	2006	1833	8564	8565	3470
8566	8567	354	8568	8569	1282	8570	5905	278	8571	777	8572	8573	8574	5527	8575	2163	8576	834
8577	8578	8579	3668	3605	8580	1318	8581	1111	5296	8582	8583	8584	8585	8586	5077	4776	8587	8588
8589	3822	8590	3849	8591	1908	948	8592	454	2533	2724	8593	8594	8595	8596	8597	8598	8599	1492
3146	3848	8600	8601	8602	8603	8604	237	2380	3128	8605	3239	8606	8607	8608	893	8609	424	2345
202	8610	8611	8612	8613	3661	2377	8614	8615	3562	8616	8617	8618	8619	2467	2577	2073	2085	3071
8620	8621	3046	8622	8623	3232	8624	8625	1205	4756	3627	623	8626	8627	152	8628	2525	8629	415
8630	8631	8632	8633	8634	5094	8635	8636	2475	8637	5816	1136	1558	1254	2425	3943	2593	2421	1832
8638	1364	8639	8640	3460	787	5481	8641	2607	8642	1656	2811	8643	5573	8644	8645	1756	8646	83
684	8647	8648	2426	8649	8650	3948	766	3025	3726	3299	8651	8652	1615	8653	8654	8655	1252	8656
8657	5124	8658	8659	1838	3604	790	3135	574	8660	8661	8662	8663	8664	2082	2510	8665	2553	1570
3904	1280	1360	1230	8666	5180	8667	8668	8669	2095	8670	8671	8672	8673	8674	8675	8676	667	1412
8677	8678	8679	8680	287	3068	8681	8682	8683	1691	977	8684	8685	8686	8687	8688	8689	8690	637
8691	1008	8692	8693	8694	8695	626	8696	5349	1969	167	8697	8698	1578	5083	2466	8699	8700	8701
8702	8703	1794	8704	160	8705	706	8706	8707	912	596	8708	8709	8710	8711	3633	1305	8712	8713
8714	8715	707	3694	2328	8716	8717	8718	8719	8720	5126	5795	8721	786	8722	8723	1073	2646	5921
8724	3463	827	8725	3770	2169	8726	8727	8728	1425	2333	8729	8730	681	8731	4680	8732	8733	889
8734	3935	5693	8735	4908	8736	8737	8738	8739	8740	18	2946	2972	4748	3455	8741	8742	8743	1494
8744	717	1296	8745	2498	8746	8747	2427	2508	8748	1923	8749	8750	8751	3779	6	8752	1373	400
8753	8754	567	3092	8755	5084	8756	8757	1540	8758	2123	8759	1924	8760	8761	8762	8763	8764	3716
8765	8766	3213	8767	5790	2243	2641	1757	8768	8769	3116	8770	8771	8772	2409	8773	3064	8774	8775
8776	2177	8777	2594	5112	1348	3351	8778	8779	8780	8781	2269	8782	8783	8784	3857	8785	8786	8787
88	3063	1166	8788	2505	316	515	8789	3322	2935	2362	2507	8790	8791	8792	5518	8793	3373	8794
8795	1716	8796	2364	2258	5261	616	2276	758	8797	8798	779	8799	8800	1239	4600	1148	3090	8801
8802	8803	2020	8804	8805	1813	1022	8806	1837	109	3594	8807	3524	2568	2034	8808	8809	3476	795
8810	8811	8812	8813	2117	1999	4736	8814	3327	3643	2203	3211	8815	8816	8817	8818	4792	8819	1859
3382	1173	282	8820	8821	8822	8823	8824	8825	2251	641	3422	8826	305	624	8827	4808	8828	8829
8830	8831	8832	8833	1279	1748	8834	8835	540	5629	3653	1635	8836	8837	4857	8838	8839	8840	8841
8842	2705	8843	4874	8844	3335	246	8845	843	2615	8846	3743	8847	8848	1401	3488	8849	8850	2822
8851	8852	2153	379	5804	8853	5505	8854	8855	5760	1899	8856	8857	8858	8859	8860	147	8861	8862
407	498	2459	3017	8863	1823	1509	8864	8865	2139	3578	8866	8867	8868	8869	8870	1045	3739	8871
5174	8872	5281	2025	8873	8874	2592	1755	8875	8876	8877	1548	4608	8878	8879	8880	8881	8882	8883
8884	8885	2780	2864	3331	1214	8886	8887	8888	422	8889	8890	8891	1467	3803	8892	8893	2413	8894
8895	1703	8896	262	8897	1764	2857	2403	4930	1930	3113	4877	8898	3502	8899	4718	8900	1500	2052
8901	8902	2875	8903	3333	8904	837	8905	2826	1169	1788	8906	556	8907	8908	569	3525	2064	1611
705	33	8909	3945	724	5105	990	8910	8911	2442	763	8912	8913	8914	8915	8916	8917	8918	3318
8919	8920	8921	5867	3458	8922	2560	8923	8924	411	8925	2238	8926	8927	8928	6046	3321	8929	5031
983	5948	8930	2211	1704	8931	2054	5010	2285	1185	8932	3386	8933	8934	748	8935	8936	8937	8938
8939	8940	8941	8942	8943	8944	8945	5932	1273	8946	1903	274	2105	182	8947	8948	8949	601	444
2129	8950	8951	8952	8953	8954	22	8955	8956	8957	8958	8959	8960	1493	8961	8962	8963	139	3284
8964	8965	2399	3925	8966	685	2366	2814	8967	3111	8968	8969	8970	2118	1565	3271	1134	358	8971
8972	3703	5079	8973	8974	8975	3257	8976	3766	365	1561	2472	1647	956	8977	8978	5502	8979	8980
2982	1027	8981	558	179	8982	8983	8984	8985	3005	8986	8987	8988	8989	8990	2137	8991	8992	8993
8994	8995	8996	6084	6088	8997	2546	8998	5330	8999	9000	9001	3123	9002	9003	9004	9005	1187	1017
2450	4973	9006	9007	9008	9009	9010	848	9011	9012	2789	9013	9014	9015	9016	2554	1606	9017	1873
9018	3485	3102	3704	9019	117	9020	9021	9022	9023	263	1870	5383	2718	4766	9024	740	2465	2959
3468	9025	9026	9027	9028	9029	1516	1948	3212	9030	9031	3120	3509	9032	3187	9033	9034	945	2138
1872	4753	3012	9035	1595	9036	742	3293	1892	394	9037	562	2394	3941	9038	523	9039	2439	9040
9041	9042	9043	9044	2966	9045	9046	2275	5021	2196	78	8	3519	9047	9048	9049	9050	9051	9052
9053	6057	9054	1915	9055	6094	427	9056	2323	9057	9058	9059	9060	9061	1229	9062	5919	9063	5897
2888	9064	4853	497	9065	9066	9067	5412	9068	9069	4804	9070	2457	9071	2358	3883	4633	9072	5560
3300	3084	9073	9074	3692	9075	1707	9076	1834	9077	9078	9079	9080	9081	9082	3466	1476	9083	635
9084	9085	5995	9086	3428	9087	9088	2423	2048	9089	3699	9090	9091	2016	1201	625	410	3077	9092
118	9093	9094	9095	3756	9096	1342	385	5613	9097	840	9098	9099	9100	9101	9102	1648	9103	2215
9104	9105	9106	9107	993	9108	2788	1381	9109	4834	9110	9111	3094	4742	9112	1591	3706	9113	1825
9114	9115	198	887	1290	9116	9117	4612	507	9118	9119	344	3439	9120	9121	9122	9123	3152	9124
5729	24	9125	1553	9126	9127	2408	2431	9128	9129	3200	9130	2374	9131	5047	9132	3085	1371	9133
9134	3564	4831	2625	700	1386	1616	5310	5645	5488	9135	9136	9137	9138	9139	9140	4964	94	9141
3698	9142	9143	9144	9145	2759	1852	9146	9147	9148	9149	9150	5263	9151	9152	3032	9153	3215	9154
9155	5185	9156	281	9157	2882	510	465	9158	9159	9160	9161	2992	3133	30	9162	9163	9164	9165
9166	2096	9167	9168	1103	5070	2858	9169	672	1138	3553	9170	1026	9171	9172	9173	935	9174	453
3370	9175	954	2455	5024	9176	9177	5388	9178	9179	9180	3109	2164	2159	9181	573	9182	9183	1137
5579	1365	650	551	9184	612	1808	110	9185	9186	9187	1455	3104	9188	3537	1949	390	21	3523
9189	9190	9191	9192	9193	9194	9195	3723	9196	4924	9197	9198	9199	4910	9200	9201	5245	2932	9202
3359	9203	9204	9205	9206	9207	9208	9209	9210	9211	9212	9213	9214	1471	9215	6106	9216	5195	1596
389	9217	46	3281	69	9218	2239	2631	3893	1232	269	5952	9219	932	2657	5976	9220	9221	9222
9223	9224	6051	9225	9226	9227	9228	9229	3741	67	1978	9230	658	3900	9231	5187	3570	9232	1875
5641	1223	638	9233	9234	1579	9235	1116	9236	56	9237	1975	9238	9239	192	5582	9240	844	9241
5143	9242	3585	1477	1708	755	9243	3361	1654	9244	9245	9246	9247	1143	9248	9249	9250	44	9251
3048	9252	5157	5040	3928	376	4747	9253	9254	666	3052	1696	2490	1594	4713	9255	9256	1967	346
9257	3256	9258	1602	9259	9260	3334	761	426	9261	9262	1526	1068	9263	9264	9265	9266	9267	5775
9268	2732	1181	9269	2110	9270	3791	3275	1800	9271	9272	3309	4914	3696	9273	9274	9275	3527	9276
2056	9277	9278	9279	9280	9281	9282	1920	9283	9284	565	9285	3895	5260	5253	1655	1762	9286	9287
998	9288	9289	9290	9291	9292	9293	1399	9294	3182	9295	208	5440	9296	2612	9297	9298	9299	4959
9300	9301	9302	2979	3810	9303	5476	9304	1110	9305	9306	3050	1989	2527	1855	9307	3467	9308	9309

9310	9311	3590	3401	663	9312	9313	1671	9314	9315	9316	9317	9318	2765	9319	9320	9321	9322	4858	
9323	131	9324	9325	9326	9327	5808	1453	1078	9328	397	9329	9330	17	9331	645	3304	9332	9333	
307	348	9334	9335	9336	9337	9338	9339	9340	854	9341	4732	2132	9342	9343	3666	9344	9345	9346	
9347	9348	9349	9350	9351	2199	1946	9352	9353	9354	3408	1184	3686	5681	9355	1669	691	9356	5046	
260	600	9357	9358	9359	20	3418	9360	3377	3690	9361	9362	9363	9364	9365	9366	9367	2474	9368	
9369	5978	914	9370	9371	1383	1324	1114	9372	9373	9374	2740	2308	9375	2571	9376	9377	9378	2621	
9379	2570	9380	1191	2456	9381	9382	3364	9383	9384	2996	807	9385	9386	2451	9387	36	9388	5460	
9389	5413	9390	1706	9391	9392	9393	9394	9395	9396	1416	367	2714	9397	9398	9399	9400	462	1582	
9401	9402	9403	1267	3889	9404	9405	381	9406	9407	3680	9408	5172	2685	5910	4809	310	9409	247	
9410	9411	1688	9412	9413	9414	696	9415	9416	9417	9418	3731	9419	9420	3797	3007	9421	9422	9423	
9424	9425	2368	9426	9427	4652	9428	9429	9430	9431	6027	9432	4784	2807	3173	2253	2493	1795	9433	
183	4882	1824	9434	1157	3749	9435	9436	1102	9437	9438	1272	9439	9440	9441	9442	2381	1814	9443	
9444	9445	2787	9446	9447	1913	2388	9448	9449	9450	2806	3416	1865	9451	204	3631	3378	2640	1786	
9452	9453	5858	3411	9454	9455	9456	205	9457	9458	9459	9460	5741	2712	9461	9462	2651	1263	9463	
9464	1474	1085	9465	2061	579	9466	9467	9468	300	1882	1319	9469	9470	9471	545	9472	2188	9	
3366	9473	1506	9474	1681	749	4738	9475	9476	9477	9478	4575	9479	986	1380	2419	9480	1040	3516	
9481	9482	5651	2654	9483	3886	9484	9485	5162	1778	2695	9486	9487	9488	9489	221	3445	9490	9491	
3106	9492	2795	5499	9493	1799	1900	9494	9495	3558	2019	9496	2468	1096	3413	9497	3684	2556	5058	
9498	9499	9500	9501	1538	9502	1153	9503	3839	9504	9505	9506	9507	9508	9509	490	2976	3674	9510	
9511	9512	2	3477	1079	3096	1182	9513	9514	693	9515	2567	3931	9516	9517	9518	2473	9519	5342	
9520	9521	9522	6031	1807	2720	5170	9523	9524	1245	812	9525	9526	1226	9527	9528	9529	9530	3808	
9531	858	1933	9532	1668	1746	5373	9533	3492	4913	9534	2386	2659	1240	4844	2360	3175	9535	9536	
5843	3374	9537	698	5654	9538	1361	2516	9539	9540	1322	5797	2561	9541	9542	4615	3041	9543	1775	
9544	4772	9545	701	9546	9547	2958	9548	1922	9549	9550	9551	9552	3273	9553	2962	4769	9554	9555	
1768	9556	9557	9558	4926	9559	9560	9561	9562	9563	9564	1235	9565	9566	9567	9568	9569	9570	47	
2919	5985	2329	2242	9571	9572	9573	2869	9574	4654	1397	5657	5750	5873	9575	9576	9577	3016	9578	
3248	3339	9579	3474	1879	5005	1804	2841	5780	6022	9580	9581	5800	9582	9583	2939	9584	91	9585	
9586	9587	9588	9589	9590	9591	3030	9592	2777	9593	9594	2829	2818	9595	5841	9596	5056	9597	2363	
4739	9598	9599	9600	9601	9602	3675	6078	9603	9604	636	3296	9605	3038	9606	9607	309	9608	9609	
9610	1446	9611	2937	2241	9612	9613	9614	3449	9615	9616	9617	3338	3538	9618	9619	9620	2903	9621	
1683	9622	9623	9624	3057	9625	9626	9627	1495	2249	1925	9628	9629	634	5637	929	9630	9631	39	
753	9632	3639	9633	9634	9635	128	4887	3088	504	9636	3395	1292	5736	2081	1052	3740	9637	1552	
9638	9639	2075	9640	9641	1968	2686	9642	9643	9644	9645	3602	9646	9647	1806	9648	9649	9650	9651	
1581	3652	2837	9652	4635	9653	9654	9655	9656	845	3362	173	3531	9657	2502	2099	1006	9658	1128	
606	9659	5288	9660	9661	4730	9662	4888	4705	1302	4609	50	2604	135	9663	9664	9665	1209	9666	
3018	9667	909	3472	1164	9668	9669	9670	9671	1556	229	9672	6105	9673	2643	9674	3072	2846	4893	
2044	9675	9676	4724	1428	9677	9678	9679	9680	1520	9681	9682	9683	1097	9684	9685	9686	4689	9687	
1550	9688	3700	9689	1544	9690	9691	3536	6090	9692	676	2114	9693	1484	5016	9694	1250	1962	9695	
817	9696	771	2949	3054	9697	3784	9698	9699	9700	9701	153	9702	2260	5428	3216	3495	9703	2854	
9704	9705	1943	2330	9706	9707	5835	1167	9708	1902	9709	9710	9711	2449	9712	9713	3937	320	9714	
366	9715	1354	3710	9716	9717	2226	3688	660	9718	9719	1829	4801	9720	9721	9722	907	2299	9723	
3701	9724	4981	9725	9726	9727	2849	9728	5828	3294	9729	9730	3277	9731	1328	3920	3075	2121	1831	
9732	9733	9734	1081	9735	9736	3424	738	586	9737	9738	2233	9739	9740	9741	1955	5632	9742	9743	
420	9744	3669	9745	9746	5075	9747	9748	3388	141	9749	9750	9751	9752	9753	4688	4829	3259	2221	
1013	3720	9754	4843	1449	9755	1222	9756	9757	9758	3435	3285	2998	3946	611	9759	1588	9760	9761	
9762	9763	9764	9765	9766	9767	3853	9768	632	9769	9770	9771	5252	502	9772	3155	9773	9774	824	
429	2684	1942	9775	2483	9776	5173	9777	375	9778	9779	9780	9781	9782	9783	2750	199	1511	3148	
9784	530	9785	9786	9787	9788	9789	9790	9791	9792	3208	2688	158	9793	1419	9794	9795	2018	9796	
2406	906	9797	9798	5622	2704	9799	9800	9801	9802	9803	1155	1637	3267	2185	9804	870	9805	9806	
9807	9808	9809	9810	9811	9812	9813	751	1880	9814	9815	1283	9816	2617	9817	9818	9819	9820	1025	
9821	9822	9823	49	3385	9824	9825	9826	9827	1180	9828	9829	2816	3352	9830	9831	2401	9832	125	
1358	508	9833	9834	9835	112	9836	9837	9838	9839	9840	642	9841	9842	9843	9844	9845	816	3642	
9846	9847	9848	9849	813	5689	9850	9851	9852	9853	5188	9854	2850	9855	3718	2572	9856	1564	9857	
9858	9859	2433	9860	9861	80	6082	9862	9863	9864	3270	9865	1304	81	1384	303	2954	940	9866	
214	5012	4935	9867	1653	1130	9868	9869	9870	9871	3255	9872	9873	9874	9875	9876	9877	317	9878	
9879	9880	9881	2495	3606	866	9882	785	9883	856	3201	2463	5427	9884	9885	3940	9886	9887	1432	
797	220	9888	9889	9890	38	442	3672	3880	3630	9891	9892	2112	9893	5068	9894	9895	9896	2200	
2689	9897	3560	2355	9898	2951	9899	3542	9900	9901	9902	5500	9903	3037	1445	2770	1488	3031	6038	
1926	9904	414	2981	3161	9905	5535	1194	3899	860	5009	1392	226	713	9906	2735	1640	3664	5874	
5352	9907	469	9908	9909	9910	9911	3817	9912	5106	9913	1785	9914	9915	9916	943	9917	9918	9919	
2404	1475	2748	9920	9921	9922	9923	9924	4820	9925	2344	9926	9927	9928	2634	3518	9929	3421	9930	
9931	249	1566	9932	9933	9934	3632	9935	9936	9937	9938	9939	9940	9941	9942	9943	1093	9944	3209	
9945	850	9946	9947	2963	9948	1987	511	1112	2967	9949	593	1574	9950	2047	9951	9952	1266	9953	
9954	9955	9956	459	1426	9957	3039	3020	9958	5	9959	9960	9961	9962	9963	9964	2227	9965	9966	
9967	286	9968	9969	608	9970	9971	64	9972	9973	920	1372	9974	9975	1021	1133	9976	9977	9978	
9979	2143	6076	9980	9981	9982	3344	715	9983	9984	5601	4625	3336	9985	1151	9986	9987	9988	3406	
9989	1662	9990	2622	1531	9991	5491	9992	1041	9993	2895	2116	2125	9994	9995	9996	9997	3648	9998	
9999	3167	2071	1313	10000	10001	3266	10002	10003	3702	5747	10004	10005	1064	5451	10006	1429	4822	3581	
4702	2596	643	10007	10008	10009	10010	3205	10011	253	10012	10013	10014	6009	10015	2844	10016	10017	825	
2186	3871	3103	2292	5803	10018	10019	5701	10020	3003	10021	5968	10022	10023	10024	10025	4903	10026	699	
522	2453	10027	2825	10028	10029	1	364	2115	10030	10031	10032	10033	10034	1557	1307	10035	10036	10037	
10038	10039	10040	2127	10041	10042	673	10043	10044	10045	3134	3464	10046	10047	10048	10049	3689	10050	10051	
2578	10052	10053	10054	10055	10056	10057	5213	1402	1024	10058	10059	6008	1601	10060	5473	3854	10061	729	
10062	10063	223	1242	1257	1303	10064	10065	10066	3110	10067	10068	3846	10069	10070	10071	10072	157	3508	
2862	2191	10073	563	10074	2745	10075	1519	3202	3737	1820	1738	10076	25	2753	10077	10078	10079	10080	
161	2144	686	10081	10082	10083	10084	10085	5821	2840	10086	4781	1228	5553	10087	2983	10088	10089	3783	

10090 3884 10091 10092 10093 828 10094 2716 4963 79 3859 1382 10095 4725 1329 936 10096 2672 2361
10097 10098 3151 1816 10099 1231 10100 10101 3500 1271 10102 3909 831 10103 2965 1009 10104 10105 10106
10107 10108 525 2955 10109 10110 335 5900 10111 5682 10112 350 1868 10113 10114 2588 3149 10115 4684
10116 10117 4607 10118 3894 10119 10120 10121 10122 10123 10124 2317 70 10125 1225 2396 1518 2921 5468
10126 10127 934 10128 10129 419 2878 437 10130 10131 10132 10133 3440 10134 5108 2397 602 10135 10136
1848 206 891 1463 5783 10137 10138 1144 10139 10140 3454 1391 1856 4599 10141 10142 10143 361 10144
10145 10146 10147 10148 10149 342 144 10150 10151 5700 10152 885 2632 2619 10153 10154 3758 3019 2952
10155 10156 5548 2405 2774 10157 10158 3055 10159 10160 3130 878 10161 10162 2220 4947 296 1049 10163
10164 10165 3504 10166 10167 10168 2518 4761 10169 1323 10170 10171 10172 10173 10174 2943 1939 10175 2195
2182 886 10176 10177 26 10178 2007 5287 1127 1521 4992 140 10179 5319 1353 212 3682 10180 3241
10181 10182 10183 10184 10185 10186 10187 132 1513 2080 2931 10188 10189 2013 10190 1728 10191 10192 10193
10194 2866 10195 10196 10197 10198 10199 10200 3838 10201 1614 10202 10203 369 10204 5805 10205 1293 1779
175 1527 2802 10206 2541 3738 1793 5721 10207 3124 2141 10208 3611 911 10209 710 10210 10211 1314
10212 849 3513 10213 3748 10214 1901 10215 2348 10216 10217 3105 2637 2974 10218 10219 10220 3101 10221
10222 10223 3131 3534 10224 10225 10226 2980 10227 10228 10229 10230 1849 5792 10231 5887 3496 1443 10232
992 10233 10234 4787 10235 5635 10236 2314 314 5958 10237 10238 2810 2901 10239 1369 2761 10240 10241
2696 5227 2855 486 10242 1481 10243 3828 2122 2340 1896 10244 3243 2574 3805 1721 2595 2747 2302
1675 10245 10246 10247 10248 5257 5125 3236 10249 5120 10250 3831 5420 10251 5424 10252 3250 996 10253
10254 883 10255 5435 1161 10256 188 10257 10258 10259 10260 10261 3583 1430 55 10262 3316 1931 10263
10264 10265 10266 10267 373 997 2167 283 10268 10269 10270 2311 2559 10271 10272 10273 10274 10275 10276
10277 10278 2925 10279 1510 10280 10281 2549 386 10282 10283 1175 10284 10285 10286 10287 3837 2218 10288
3913 2022 10289 10290 10291 10292 1101 10293 1792 10294 2057 325 1436 2017 664 10295 3091 10296 2771
10297 1092 10298 517 10299 3221 1298 10300 10301 10302 5649 3735 10303 1058 10304 10305 345 10306 1070
10307 581 1988 4923 2133 1465 2063 3154 10308 5344 441 10309 2509 10310 10311 10312 4835 10313 10314
5773 10315 10316 2874 10317 10318 2156 1507 5964 1844 10319 3127 10320 10321 3121 10322 10323 10324 5994
10325 10326 166 10327 291 10328 1038 265 3169 5026 10329 10330 10331 10332 10333 3307 1751 10334 3636
10335 10336 248 10337 10338 3230 10339 10340 10341 10342 1355 1337 10343 838 32 3086 2999 10344 10345
881 10346 10347 1237 1422 10348 10349 2828 10350 1571 823 10351 10352 2734 10353 10354 10355 1726 10356
10357 10358 10359 10360 10361 10362 10363 5784 3021 3539 10364 4641 10365 10366 2648 10367 1597 2026 5027
10368 2040 10369 10370 10371 10372 2257 10373 10374 2677 10375 809 3772 5206 10376 1593 10377 10378 10379
10380 2331 10381 10382 10383 10384 3522 3566 3874 2126 184 10385 10386 347 10387 10388 10389 3204 5177
10390 13 10391 10392 3244 1547 215 10393 10394 10395 4640 82 10396 10397 10 1514 10398 10399 10400
10401 1497 10402 10403 3873 10404 1919 10405 988 97 3002 10406 10407 10408 10409 2512 10410 10411 10412
10413 10414 10415 764 10416 1406 5097 10417 10418 2469 1512 2378 2764 10419 5846 10420 2461 65 10421
5908 10422 10423 3190 597 10424 3796 3436 5624 10425 10426 10427 1417 2209 10428 10429 4861 3663 3676
245 10430 10431 10432 3011 10433 10434 10435 10436 10437 10438 10439 10440 10441 2773 2820 10442 10443 10444
5525 1012 10445 10446 10447 10448 6011 10449 10450 10451 10452 10453 2772 10454 10455 10456 3394 10457 10458
10459 1152 2940 10460 1082 10461 3254 10462 2376 10463 3367 10464 5955 10465 5545 10466 10467 3498 10468
2755 10469 10470 2347 10471 10472 1379 2767 203 10473 4719 1227 10474 10475 3014 1016 10476 10477 10478
311 1972 3234 953 3115 10479 2194 3108 10480 200 731 10481 10482 10483 10484 599 2997 10485 10486
10487 10488 10489 10490 2642 2170 10491 10492 1732 194 10493 1002 3847 3004 2078 10494 3944 10495 10496
10497 832 267 10498 142 10499 5829 10500 1150 10501 10502 10503 3430 10504 10505 1195 10506 1056 1624
382 10507 5400 10508 2441 10509 1803 456 10510 3095 3303 272 5683 5847 3655 3775 1898 1720 10511
3891 10512 10513 4949 1604 10514 10515 10516 10517 10518 10519 10520 2282 821 10521 10522 138 10523 1723
3145 10524 10525 10526 10527 3555 2692 10528 1174 1349 1639 89 10529 5127 10530 546 3251 10531 2289
10532 10533 591 10534 10535 10536 2157 10537 10538 10539 1340 10540 1253 10541 3363 1863 5028 10542 10543
10544 3670 10545 2823 10546 10547 10548 10549 1790 10550 494 1854 2945 5209 4814 2990 10551 869 10552
10553 10554 10555 10556 10557 2879 10558 10559 3753 3023 10560 331 2922 10561 3350 10562 1341 10563 10564
10565 1739 10566 10567 862 10568 590 10569 10570 1317 288 2295 10571 2412 10572 10573 3490 114 10574
2235 5556 3860 10575 10576 10577 888 5509 360 10578 10579 3089 10580 10581 5990 10582 10583 3262 3199
1204 3059 10584 2501 10585 10586 10587 10588 10589 5661 3568 10590 10591 10592 2987 6059 3660 10593 10594
10595 10596 10597 10598 10599 1747 3314 10600 10601 5060 478 10602 1462 1388 10603 10604 1940 4928 10605
10606 814 1370 10607 10608 2713 10609 10610 10611 10612 10613 3865 10614 10615 10616 1877 1159 10617 10618
10619 952 5718 10620 10621 10622 10623 10624 10625 10626 1959 10627 822 10628 5892 10629 10630 10631 1573
10632 10633 1670 3380 4968 628 1439 4953 10634 3612 10635 1031 585 10636 2989 5216 493 10637 10638
2890 2731 10639 10640 4657 10641 10642 2151 10643 4998 1046 10644 10645 10646 4879 3533 784 10647 10648
2270 10649 10650 5907 4644 10651 4610 3245 10652 1652 5100 10653 10654 2975 10655 10656 3719 3752 10657
10658 10659 2352 10660 10661 10662 5363 230 2109 2627 10663 10664 10665 723 2357 2372 10666 10667 10668
1632 2202 10669 10670 10671 1071 10672 10673 5445 10674 6107 2618 10675 2379 10676 3730 10677 1705 800
2041 2906 10678 10679 2248 3722 10680 10681 10682 10683 10684 2356 10685 10686 2424 2697 5868 3582 10687
5516 10688 10689 10690 10691 10692 2021 1840 301 2752 10693 10694 10695 10696 10697 1077 10698 1730 10699
3777 10700 10701 3673 10702 942 10703 10704 4896 10705 10706 1853 661 2511 3843 10707 5324 10708 5957
5308 1362 10709 10710 2224 4794 10711 1857 10712 10713 3181 2165 10714 10715 10716 3799 10717 3360 3441
10718 5300 10719 10720 1094 1912 1745 10721 10722 2103 1089 778 10723 3677 485 10724 5487 5159 10725
1577 90 10726 10727 3325 2606 10728 5774 6015 644 1469 10729 2542 10730 865 10731 1697 10732 374
10733 1881 10734 393 3479 10735 4588 10736 3628 10737 1215 10738 10739 10740 2538 10741 10742 10743 10744
10745 277 2699 10746 10747 10748 399 1197 2791 10749 10750 5647 908 1664 10751 10752 10753 1698 10754
10755 2266 3457 2786 14 1646 1470 1357 10756 1549 3231 10757 10758 1862 10759 5446 1936 10760 10761
190 10762 3938 3456 10763 651 505 10764 10765 10766 1628 10767 10768 10769 3279 633 235 3183 371
5113 10770 3637 1750 10771 1393 232 559 3901 180 10772 10773 150 10774 10775 3552 10776 679 10777
10778 10779 10780 10781 3667 10782 10783 725 3746 10784 1612 1331 680 2768 10785 5969 2923 2150 1712
1320 10786 728 688 1878 529 607 4944 10787 5098 10788 10789 10790 3786 185 11 1277 10791 2325
2698 10792 10793 10794 10795 4649 1450 1938 10796 10797 5915 3323 1759 994 10798 3288 10799 10800 10801
3114 10802 10803 561 10804 10805 2881 2210 10806 2727 3724 2334 10807 10808 10809 10810 10811 3249 10812
1247 2287 10813 10814 10815 2273 5515 913 861 10816 5610 2600 10817 10818 10819 408 2135 129 10820
5940 772 10821 5182 1437 1921 10822 3289 10823 1667 10824 10825 10826 10827 3917 10828 10829 10830 3405

10831 10832 10833 10834 10835 3609 4638 10836 10837 5508 10838 10839 3832 10840 3658 10841 10842 10843 28
584 10844 10845 3404 10846 4778 10847 750 3603 4819 499 3024 10848 1118 5717 2050 5979 1489 2603
10849 10850 3478 774 3157 10851 10852 10853 284 10854 10855 10856 10857 351 10858 2703 1951 1866 4581
10859 890 10860 10861 1249 10862 10863 10864 10865 10866 10867 5483 10868 10869 10870 10871 10872 10873 4883
10874 10875 10876 10877 10878 4668 10879 10880 4731 10881 10882 10883 10884 10885 10886 10887 10888 10889 10890
4614 10891 10892 10893 10894 10895 10896 10897 5698 10898 10899 10900 10901 10902 10903 10904 10905 10906 10907
10908 10909 10910 10911 10912 10913 10914 10915 4619 4758 5652 10916 10917 5660 5290 10918 10919 10920 10921
5434 10922 10923 10924 10925 10926 10927 10928 10929 10930 5672 10931 10932 10933 10934 10935 10936 10937 10938
10939 10940 10941 10942 10943 10944 10945 10946 10947 10948 5007 4911 10949 10950 10951 10952 4630 10953 10954
10955 10956 10957 10958 10959 10960 10961 10962 10963 10964 10965 10966 10967 10968 10969 5457 10970 10971 10972
10973 10974 5334 10975 10976 10977 10978 10979 10980 10981 10982 5756 10983 10984 10985 10986 6083 10987 10988
5916 10989 10990 10991 10992 4909 10993 10994 10995 10996 10997 10998 10999 11000 11001 11002 11003 11004 11005
11006 5289 11007 11008 11009 11010 11011 5429 11012 11013 11014 11015 11016 11017 5901 11018 11019 11020 11021
11022 11023 11024 11025 11026 11027 5594 11028 11029 11030 11031 11032 11033 11034 11035 11036 11037 11038 11039
11040 4962 11041 11042 11043 11044 11045 4768 11046 5295 11047 11048 5284 11049 4704 11050 11051 11052 5140
11053 11054 5377 11055 6050 11056 11057 11058 11059 11060 11061 11062 11063 11064 11065 6005 11066 4646 11067
11068 11069 4613 11070 11071 11072 11073 11074 11075 11076 11077 11078 5671 11079 4695 11080 11081 11082 11083
11084 11085 11086 11087 11088 11089 11090 11091 11092 11093 11094 11095 11096 11097 11098 11099 11100 11101 11102
11103 11104 5343 11105 11106 11107 11108 11109 11110 11111 11112 6063 11113 11114 4716 4830 11115 11116 11117
11118 11119 11120 11121 11122 6041 11123 5962 11124 11125 11126 11127 11128 11129 11130 11131 11132 11133 11134
11135 11136 11137 11138 11139 11140 11141 11142 11143 4927 11144 5181 11145 11146 11147 11148 11149 11150 5966
4578 11151 11152 11153 11154 11155 5857 11156 11157 11158 11159 11160 11161 11162 11163 11164 11165 11166 11167
11168 11169 5927 11170 11171 11172 11173 5581 11174 11175 11176 11177 4740 11178 11179 11180 11181 11182 11183
11184 11185 11186 11187 11188 11189 11190 11191 11192 11193 4675 11194 11195 5082 11196 11197 5880 11198 11199
11200 11201 5540 11202 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11213 11214 11215 11216 11217
11218 11219 11220 11221 11222 11223 11224 11225 11226 11227 11228 11229 11230 11231 11232 11233 11234 11235 11236
11237 11238 11239 11240 11241 11242 11243 11244 11245 5971 11246 11247 11248 11249 11250 11251 11252 11253 11254
11255 11256 11257 11258 11259 11260 11261 11262 11263 11264 5894 11265 11266 11267 11268 11269 11270 11271 11272
11273 11274 4624 11275 11276 11277 11278 11279 4984 11280 11281 11282 11283 11284 11285 11286 11287 11288 11289
5676 11290 11291 11292 11293 11294 5628 11295 11296 11297 11298 11299 5018 4703 11300 11301 11302 11303 11304
11305 11306 11307 11308 11309 11310 11311 6102 11312 11313 5492 11314 11315 11316 11317 11318 5771 11319 11320
11321 11322 11323 11324 11325 11326 11327 11328 11329 11330 5275 4655 11331 11332 11333 11334 5136 5980 11335
11336 11337 11338 11339 11340 11341 11342 4734 11343 11344 11345 11346 11347 11348 5465 4860 11349 11350 11351
5199 11352 11353 11354 11355 11356 11357 5564 11358 11359 5731 11360 11361 5355 5762 11362 11363 11364 11365
11366 5183 11367 11368 11369 11370 11371 11372 11373 11374 11375 11376 11377 11378 4939 11379 11380 11381 11382
5357 11383 6032 5367 11384 11385 11386 11387 11388 11389 5588 11390 11391 11392 11393 11394 11395 11396 11397
11398 11399 11400 11401 11402 11403 11404 11405 11406 11407 11408 11409 11410 11411 11412 11413 11414 11415 11416
11417 11418 11419 11420 11421 11422 11423 5230 11424 11425 5495 11426 11427 11428 11429 11430 11431 11432 11433
11434 11435 11436 11437 11438 11439 4726 11440 11441 5003 4785 5240 11442 11443 11444 11445 11446 11447 11448
11449 11450 11451 5655 5493 11452 11453 4708 4580 11454 11455 11456 11457 11458 11459 11460 11461 5048 11462
11463 11464 11465 11466 11467 11468 11469 6061 11470 5250 11471 11472 11473 11474 11475 5149 11476 11477 11478
11479 11480 11481 11482 11483 11484 11485 11486 11487 11488 11489 11490 11491 11492 11493 11494 11495 11496 5608
11497 11498 11499 5882 5071 4986 11500 11501 11502 11503 11504 5043 11505 11506 5562 11507 11508 11509 11510
4786 11511 11512 11513 11514 11515 11516 11517 11518 11519 11520 11521 11522 11523 11524 5678 11525 11526 11527
11528 11529 11530 11531 11532 11533 11534 5702 11535 11536 11537 4938 11538 11539 11540 11541 11542 11543 5722
11544 11545 11546 11547 11548 11549 11550 11551 11552 11553 11554 11555 11556 11557 11558 11559 11560 11561 11562
11563 11564 11565 11566 11567 11568 11569 11570 11571 11572 11573 11574 11575 11576 11577 11578 11579 11580 11581
11582 11583 5237 11584 11585 11586 11587 11588 11589 11590 11591 11592 11593 11594 11595 5148 11596 11597 11598
11599 11600 11601 11602 5685 11603 4666 11604 11605 11606 11607 11608 11609 11610 11611 11612 11613 11614 11615
11616 11617 11618 11619 11620 11621 11622 11623 11624 11625 11626 11627 11628 11629 11630 11631 11632 5086 5621
11633 11634 11635 11636 11637 11638 11639 11640 5517 11641 11642 11643 11644 11645 5200 11646 11647 11648 11649
11650 11651 11652 11653 11654 11655 5744 11656 4602 11657 11658 11659 11660 11661 11662 11663 11664 5035 11665
11666 11667 11668 4586 5726 11669 4637 4603 11670 11671 11672 11673 11674 11675 11676 11677 11678 11679 11680
11681 11682 11683 11684 11685 11686 11687 11688 11689 11690 11691 4991 5674 11692 11693 11694 11695 5552 11696
5147 11697 11698 11699 11700 11701 5380 11702 11703 11704 11705 11706 11707 11708 11709 5811 11710 11711 11712
11713 11714 11715 11716 11717 11718 11719 11720 11721 11722 11723 11724 5612 11725 11726 5623 11727 5283 11728
6052 11729 11730 4869 11731 11732 11733 11734 11735 11736 11737 11738 11739 11740 11741 11742 5103 11743 11744
11745 11746 11747 11748 11749 11750 4867 11751 11752 11753 11754 11755 11756 11757 11758 11759 11760 11761 11762
11763 11764 11765 11766 11767 5834 11768 4902 11769 11770 11771 4662 4974 4693 11772 11773 5998 11774 11775
11776 11777 5276 11778 11779 11780 4779 11781 11782 11783 11784 11785 4954 5309 11786 11787 5282 11788 11789
11790 11791 11792 6099 11793 6060 5091 11794 5474 11795 11796 11797 11798 5325 11799 11800 11801 11802 11803
11804 11805 11806 11807 4611 11808 11809 5925 5212 11810 5706 11811 11812 11813 11814 11815 5806 11816 5724
11817 11818 11819 11820 11821 11822 11823 11824 11825 11826 11827 11828 11829 5557 11830 11831 11832 4626 11833
5101 11834 11835 11836 11837 11838 5266 11839 11840 11841 11842 11843 11844 11845 11846 11847 11848 11849 11850
11851 11852 11853 11854 11855 11856 11857 11858 11859 11860 11861 11862 11863 11864 11865 11866 11867 5235 11868
11869 11870 11871 11872 11873 11874 11875 5709 11876 11877 11878 4885 11879 11880 11881 11882 11883 11884 11885
11886 11887 11888 11889 11890 11891 11892 11893 11894 11895 11896 11897 11898 11899 11900 11901 11902 11903 11904
11905 5062 11906 11907 11908 11909 11910 11911 11912 11913 11914 11915 5328 11916 11917 5053 11918 11919 11920
11921 5470 11922 11923 11924 11925 11926 11927 11928 11929 11930 11931 11932 11933 5286 11934 11935 11936 11937
11938 11939 11940 11941 11942 11943 11944 11945 11946 11947 11948 11949 11950 11951 11952 11953 11954 4584 11955
11956 11957 11958 11959 11960 11961 11962 11963 11964 11965 11966 11967 11968 11969 5432 11970 11971 11972 11973
11974 5067 11975 11976 11977 11978 11979 11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
11992 11993 11994 5567 5830 11995 11996 11997 11998 11999 12000 12001 12002 12003 12004 12005 12006 12007 12008
12009 12010 12011 12012 12013 12014 12015 12016 12017 4999 12018 12019 12020 12021 5302 12022 12023 12024 12025
12026 12027 12028 12029 12030 12031 12032 12033 5255 12034 5462 12035 12036 12037 12038 12039 12040 12041 5114
12042 12043 12044 12045 12046 12047 12048 12049 12050 6010 12051 12052 12053 12054 12055 12056 12057 12058 12059

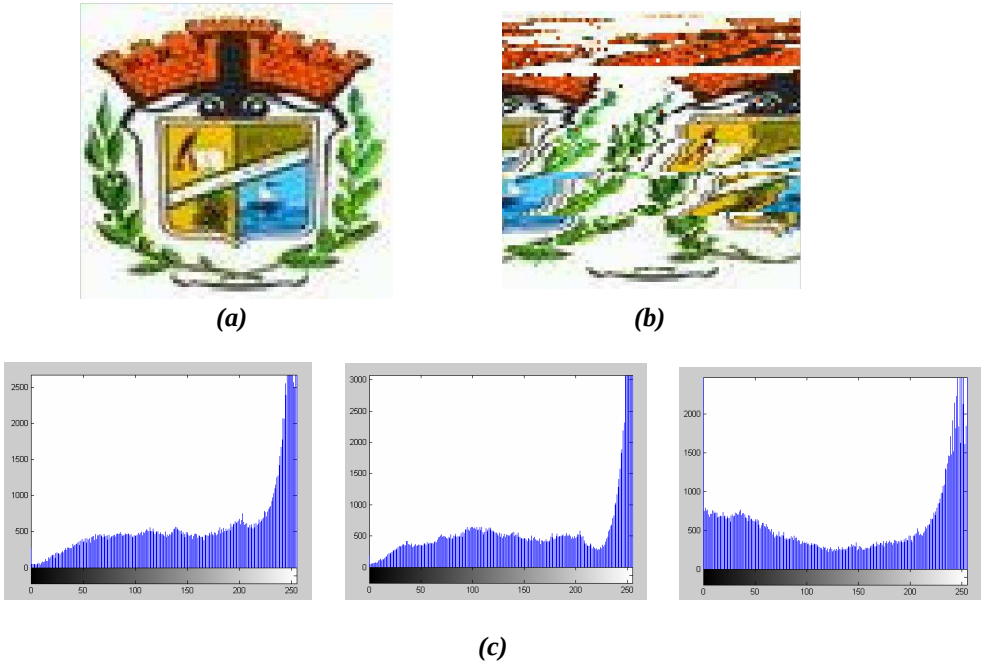
12060 12061 12062 12063 12064 5150 5299 12065 12066 12067 12068 12069 12070 12071 12072 12073 12074 12075 12076
5864 12077 12078 12079 5555 12080 12081 12082 5191 5912 12083 12084 12085 12086 12087 12088 12089 12090 12091
12092 12093 12094 12095 12096 4881 12097 12098 12099 12100 12101 12102 12103 12104 12105 12106 12107 12108 12109
4825 12110 12111 12112 12113 12114 12115 5455 12116 12117 5590 12118 12119 12120 5840 12121 12122 5145 12123
12124 12125 12126 12127 12128 12129 12130 12131 12132 12133 12134 12135 12136 12137 12138 4823 4764 12139 5087
12140 12141 12142 12143 12144 12145 12146 12147 12148 12149 12150 4686 12151 12152 12153 12154 4899 12155 12156
12157 12158 12159 12160 12161 12162 12163 12164 12165 12166 12167 12168 6025 4948 12169 12170 5577 12171 5395
6100 12172 12173 12174 12175 12176 12177 12178 12179 12180 12181 12182 12183 12184 12185 12186 12187 12188 12189
12190 12191 12192 12193 12194 12195 12196 12197 12198 12199 12200 12201 12202 12203 12204 12205 12206 12207 12208
12209 12210 12211 5327 12212 12213 12214 5630 6026 4925 12215 12216 12217 12218 12219 12220 12221 12222 12223
12224 12225 12226 12227 12228 12229 12230 12231 12232 5095 12233 5360 5845 12234 12235 6092 5135 12236 12237
12238 12239 12240 12241 12242 12243 12244 12245 12246 12247 12248 12249 12250 12251 12252 5898 12253 12254 12255
12256 12257 12258 12259 12260 12261 12262 12263 5163 12264 12265 12266 12267 12268 12269 12270 12271 12272 12273
12274 12275 12276 12277 5954 12278 12279 12280 12281 12282 12283 12284 4593 12285 12286 12287 12288 12289 12290
12291 12292 12293 12294 12295 4979 5526 5039 12296 12297 12298 12299 12300 12301 12302 12303 4674 4585 12304
12305 12306 12307 12308 5862 12309 5041 12310 12311 12312 12313 12314 12315 12316 12317 12318 12319 12320 12321
12322 12323 12324 12325 12326 12327 12328 5667 12329 5176 12330 5869 12331 12332 12333 12334 12335 5378 12336
12337 12338 12339 12340 12341 12342 5022 12343 5758 12344 12345 5123 12346 12347 5034 12348 12349 5663 12350
12351 12352 5096 5345 12353 12354 12355 12356 12357 12358 12359 5339 12360 12361 5403 5000 12362 12363 12364
12365 12366 12367 12368 12369 12370 12371 12372 12373 12374 12375 12376 12377 12378 12379 12380 12381 12382 12383
12384 12385 12386 12387 12388 12389 12390 12391 12392 12393 12394 12395 12396 12397 12398 5679 12399 12400 12401
12402 4905 12403 12404 12405 12406 12407 12408 4639 4889 12409 12410 12411 12412 12413 12414 5410 12415 12416
12417 12418 12419 12420 12421 4579 12422 12423 12424 12425 12426 12427 12428 12429 12430 12431 12432 5589 12433
12434 12435 12436 12437 12438 5616 12439 12440 12441 12442 12443 12444 12445 12446 12447 12448 12449 12450 12451
12452 12453 12454 12455 4891 12456 12457 12458 12459 12460 12461 12462 12463 12464 12465 12466 12467 5524 12468
5987 12469 12470 12471 12472 12473 12474 12475 12476 12477 12478 12479 12480 12481 12482 5316 12483 12484 12485
12486 4658 12487 12488 12489 4746 12490 12491 4745 12492 12493 12494 12495 12496 12497 12498 12499 12500 12501
12502 5002 12503 12504 12505 12506 12507 12508 12509 12510 5037 12511 12512 12513 12514 12515 12516 12517 12518
12519 4818 4791 12520 12521 12522 12523 12524 5761 12525 12526 12527 12528 12529 12530 12531 12532 12533 12534
12535 12536 5602 5072 12537 12538 5055 12539 12540 12541 12542 12543 12544 12545 12546 12547 5677 12548 12549
12550 12551 12552 12553 12554 12555 12556 12557 5178 4983 12558 12559 5093 12560 12561 12562 12563 5501 12564
12565 12566 12567 12568 12569 12570 12571 12572 12573 12574 6075 12575 12576 12577 12578 12579 12580 12581 12582
12583 12584 12585 12586 12587 12588 12589 12590 5341 12591 12592 12593 12594 12595 6036 5859 12596 12597 12598
5636 12599 12600 12601 4692 12602 12603 12604 12605 12606 12607 12608 12609 5728 12610 12611 12612 12613 12614
12615 5520 12616 12617 12618 12619 12620 12621 12622 5475 12623 12624 12625 12626 12627 12628 12629 12630 12631
12632 12633 12634 12635 5743 12636 12637 5390 12638 12639 12640 12641 12642 12643 12644 12645 12646 4741 12647
12648 12649 12650 12651 12652 12653 12654 12655 12656 12657 12658 12659 12660 12661 12662 12663 12664 4851 12665
12666 12667 12668 12669 12670 12671 5768 12672 12673 12674 12675 12676 12677 12678 12679 12680 12681 12682 12683
12684 12685 12686 12687 5640 12688 5751 12689 5575 12690 12691 12692 12693 12694 12695 12696 12697 12698 12699
12700 12701 12702 12703 12704 12705 12706 12707 12708 12709 12710 12711 5471 12712 12713 4752 5600 12714 5045
5831 12715 12716 12717 12718 12719 12720 12721 12722 5504 12723 12724 12725 12726 12727 12728 12729 12730 12731
12732 12733 5742 12734 12735 12736 12737 12738 5745 12739 12740 12741 12742 12743 12744 12745 12746 12747 12748
5085 12749 12750 12751 5298 12752 12753 12754 12755 5658 12756 12757 12758 5697 12759 5394 4798 12760 12761
12762 12763 12764 12765 12766 12767 12768 12769 12770 12771 5983 12772 12773 12774 12775 12776 12777 12778 12779
12780 12781 12782 12783 4717 12784 12785 12786 5346 5735 12787 12788 12789 12790 12791 12792 12793 12794 12795
12796 12797 12798 12799 5375 12800 12801 12802 12803 12804 5974 6055 12805 12806 12807 12808 4782 12809 12810
12811 12812 12813 12814 12815 12816 5809 12817 12818 12819 12820 12821 12822 12823 12824 4727 12825 12826 12827
12828 12829 12830 12831 12832 12833 12834 12835 12836 12837 12838 12839 12840 12841 12842 5586 12843 12844 12845
12846 12847 12848 5450 12849 12850 12851 12852 5496 12853 12854 12855 12856 12857 12858 12859 12860 12861 12862
12863 12864 12865 12866 12867 12868 12869 12870 12871 12872 12873 12874 12875 12876 12877 5443 12878 12879 12880
12881 12882 12883 5128 12884 12885 12886 12887 12888 12889 12890 5236 5076 12891 12892 12893 12894 12895 12896
12897 12898 12899 12900 12901 12902 12903 4595 12904 12905 12906 12907 12908 12909 12910 12911 12912 12913 12914
12915 12916 4849 12917 12918 12919 12920 12921 12922 12923 12924 5478 12925 12926 12927 12928 12929 12930 12931
12932 12933 12934 12935 12936 12937 12938 12939 12940 4956 12941 12942 12943 4900 12944 12945 12946 12947 12948
12949 6070 12950 12951 12952 12953 12954 12955 12956 12957 12958 5823 12959 12960 12961 12962 12963 12964 12965
12966 12967 12968 5249 12969 12970 12971 12972 12973 12974 12975 5786 12976 12977 12978 12979 12980 12981 12982
12983 12984 12985 12986 12987 12988 12989 12990 5023 12991 5819 12992 12993 6004 12994 12995 12996 5951 12997
12998 12999 13000 13001 13002 13003 13004 13005 5318 13006 13007 5961 13008 13009 13010 13011 13012 13013 13014
13015 13016 13017 13018 13019 13020 13021 13022 13023 5311 4628 13024 13025 13026 13027 13028 13029 13030 5913
4865 13031 13032 13033 13034 13035 5008 13036 5644 13037 4985 13038 13039 13040 6028 5479 13041 13042 13043
13044 13045 13046 13047 13048 13049 13050 13051 13052 13053 13054 13055 13056 13057 13058 13059 13060 13061 13062
13063 13064 13065 13066 13067 13068 13069 13070 13071 13072 13073 13074 13075 13076 13077 13078 13079 13080 5194
13081 13082 13083 13084 13085 13086 5444 13087 13088 13089 5226 13090 13091 13092 13093 13094 13095 13096 13097
13098 13099 4651 13100 13101 13102 13103 13104 5141 4982 13105 13106 13107 13108 13109 6012 13110 13111 13112
13113 13114 13115 13116 13117 13118 13119 13120 13121 13122 13123 13124 13125 13126 5374 13127 13128 13129 13130
13131 13132 13133 4989 13134 13135 13136 13137 13138 5825 13139 13140 13141 13142 13143 13144 13145 5712 13146
5625 13147 13148 13149 13150 13151 13152 13153 13154 13155 5812 13156 13157 13158 13159 13160 13161 5247 13162
13163 13164 13165 13166 13167 13168 13169 13170 13171 13172 13173 13174 13175 13176 13177 4777 13178 13179 13180
13181 13182 13183 5215 5937 13184 13185 13186 13187 13188 13189 13190 13191 13192 6108 13193 13194 4707 13195
13196 13197 13198 13199 5943 13200 13201 13202 13203 4894 13204 13205 13206 13207 13208 13209 13210 13211 13212
13213 13214 13215 13216 13217 13218 13219 13220 13221 13222 13223 13224 13225 13226 13227 13228 13229 4729 13230
5899 13231 13232 13233 13234 13235 13236 13237 13238 13239 13240 5218 13241 13242 13243 13244 13245 13246 13247
5453 13248 13249 13250 13251 13252 4813 13253 13254 13255 13256 13257 13258 13259 13260 13261 13262 13263 13264
13265 13266 13267 13268 13269 4837 13270 13271 13272 13273 13274 13275 13276 13277 13278 13279 13280 13281 5036
13282 13283 13284 13285 13286 13287 13288 13289 13290 5441 13291 13292 13293 13294 4590 13295 13296 13297 13298
13299 13300 13301 13302 5169 13303 13304 13305 13306 13307 13308 13309 13310 13311 13312 13313 13314 13315 13316

13317 13318 13319 13320 13321 13322 13323 13324 13325 13326 13327 13328 13329 13330 13331 13332 5402 13333 13334
13335 13336 13337 13338 13339 13340 13341 13342 5347 13343 13344 13345 13346 13347 13348 13349 13350 13351 13352
13353 13354 13355 5922 13356 13357 13358 13359 13360 4783 13361 5369 13362 13363 13364 13365 13366 13367 13368
13369 13370 13371 13372 13373 13374 5323 13375 13376 13377 13378 13379 13380 13381 5592 13382 13383 13384 13385
13386 13387 5563 13388 13389 13390 13391 13392 13393 13394 13395 13396 13397 13398 13399 13400 13401 13402 13403
13404 13405 13406 13407 5268 13408 6035 13409 5111 13410 13411 4653 13412 4906 13413 13414 13415 13416 13417
4636 4797 13418 13419 13420 13421 13422 13423 13424 13425 13426 13427 13428 13429 13430 13431 13432 13433 13434
13435 13436 13437 13438 13439 13440 13441 13442 13443 13444 13445 13446 13447 13448 13449 13450 5820 13451 13452
13453 13454 13455 13456 13457 13458 13459 13460 13461 13462 13463 13464 13465 13466 13467 13468 13469 13470 13471
13472 13473 13474 13475 13476 13477 5558 13478 13479 5764 13480 13481 13482 13483 13484 13485 13486 13487 4681
13488 13489 13490 13491 13492 13493 13494 13495 13496 5854 13497 13498 13499 13500 4884 13501 4800 13502 13503
13504 13505 13506 13507 13508 13509 13510 13511 13512 13513 13514 13515 13516 13517 13518 13519 13520 13521 13522
13523 13524 13525 13526 13527 13528 5886 13529 5670 13530 13531 13532 13533 5791 13534 13535 13536 13537 13538
13539 13540 13541 13542 13543 13544 13545 13546 13547 13548 13549 13550 13551 13552 13553 13554 13555 13556 13557
13558 13559 13560 13561 5883 13562 13563 13564 13565 13566 13567 13568 13569 13570 13571 13572 13573 13574 13575
13576 13577 13578 13579 13580 5001 13581 13582 13583 13584 4821 13585 13586 13587 13588 13589 13590 13591 13592
13593 13594 13595 13596 13597 13598 13599 13600 13601 13602 13603 13604 13605 13606 5425 13607 13608 13609 13610
13611 13612 13613 13614 13615 13616 13617 13618 13619 13620 13621 6034 13622 13623 13624 13625 13626 13627 13628
13629 13630 13631 4765 4642 13632 4827 13633 13634 6039 4836 13635 13636 5117 13637 13638 5837 13639 13640
13641 13642 13643 13644 13645 13646 13647 13648 13649 13650 13651 13652 13653 13654 13655 13656 13657 5551 13658
13659 13660 13661 5267 13662 13663 13664 5904 13665 13666 13667 13668 13669 13670 13671 13672 13673 13674 13675
13676 13677 13678 13679 5385 13680 13681 13682 5595 13683 13684 13685 13686 13687 13688 13689 13690 13691 13692
13693 13694 13695 5920 13696 13697 13698 13699 13700 13701 13702 13703 13704 13705 13706 13707 13708 4605 13709
13710 13711 13712 13713 13714 13715 13716 13717 5963 13718 13719 13720 13721 13722 5387 13723 13724 13725 13726
13727 13728 13729 13730 13731 13732 13733 5903 13734 13735 13736 13737 13738 5817 13739 5408 5269 13740 13741
13742 13743 13744 13745 13746 13747 13748 13749 13750 13751 13752 13753 13754 13755 13756 13757 13758 13759 13760
13761 13762 5646 13763 13764 13765 13766 13767 13768 13769 13770 13771 13772 13773 13774 13775 13776 13777 13778
13779 5597 13780 13781 13782 13783 4673 5956 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 4696
13794 13795 13796 5497 13797 13798 4712 5799 13799 13800 13801 13802 13803 13804 13805 13806 13807 13808 13809
13810 13811 13812 13813 13814 13815 13816 13817 13818 13819 13820 13821 13822 13823 13824 13825 13826 13827 13828
13829 5442 4701 13830 4859 13831 6064 13832 13833 13834 13835 13836 13837 13838 13839 13840 13841 13842 13843
13844 13845 13846 13847 13848 13849 13850 13851 13852 13853 13854 13855 13856 13857 5186 13858 13859 13860 13861
13862 13863 13864 5431 13865 13866 13867 13868 5449 13869 4995 13870 13871 13872 13873 13874 13875 13876 5604
13877 13878 13879 5406 13880 13881 13882 13883 13884 13885 13886 13887 13888 13889 13890 13891 4863 13892 13893
5695 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903 13904 13905 5815 13906 13907 13908 13909 13910
13911 13912 13913 13914 5571 13915 13916 13917 5208 13918 13919 13920 5109 13921 13922 13923 13924 13925 13926
13927 13928 13929 13930 13931 13932 13933 13934 13935 13936 13937 13938 13939 13940 5224 13941 13942 13943 13944
13945 13946 13947 5407 13948 13949 13950 13951 5189 4620 13952 4582 13953 13954 13955 13956 13957 13958 13959
13960 13961 13962 13963 13964 13965 13966 13967 13968 13969 13970 13971 5536 13972 13973 13974 13975 13976 13977
13978 13979 13980 13981 13982 5533 13983 13984 13985 5042 13986 13987 13988 13989 13990 13991 5776 13992 13993
13994 13995 4803 13996 13997 5167 13998 4743 13999 14000 14001 14002 14003 14004 14005 14006 14007 14008 14009
14010 14011 14012 14013 14014 14015 14016 14017 5274 14018 14019 14020 14021 14022 14023 14024 14025 14026 5011
14027 14028 14029 14030 5168 14031 14032 5550 14033 14034 14035 5251 14036 14037 14038 14039 14040 14041 14042
5014 14043 14044 14045 14046 14047 14048 14049 4990 14050 14051 14052 5933 14053 14054 14055 14056 14057 14058
5813 14059 14060 14061 14062 14063 5015 14064 14065 14066 5531 14067 14068 14069 14070 14071 14072 14073 14074
14075 14076 14077 14078 14079 14080 14081 14082 14083 14084 14085 14086 14087 14088 14089 5716 14090 14091 5924
14092 14093 14094 14095 14096 5749 14097 14098 14099 14100 14101 14102 14103 14104 14105 14106 14107 14108 14109
14110 14111 14112 4918 5619 14113 14114 14115 5593 14116 14117 5166 14118 14119 14120 14121 14122 14123 14124
14125 14126 14127 14128 14129 14130 14131 14132 14133 14134 14135 14136 14137 14138 14139 14140 14141 14142 14143
14144 14145 14146 14147 14148 14149 14150 14151 14152 14153 14154 14155 14156 14157 14158 14159 14160 14161 14162
14163 14164 14165 14166 5081 14167 14168 14169 14170 14171 14172 4852 5256 14173 14174 14175 14176 14177 14178
14179 4872 14180 14181 4996 14182 14183 14184 14185 14186 14187 14188 5580 14189 14190 14191 14192 14193 14194
14195 14196 14197 14198 14199 5472 5231 14200 14201 14202 14203 14204 14205 14206 14207 14208 14209 14210 5232
4750 14211 14212 14213 14214 5538 14215 14216 14217 14218 14219 14220 5788 14221 14222 14223 14224 4676 14225
14226 14227 14228 14229 14230 14231 14232 14233 14234 14235 4720 14236 14237 14238 4987 14239 14240 14241 14242
4955 14243 14244 14245 14246 14247 14248 14249 14250 14251 14252 14253 14254 14255 5004 14256 14257 14258 14259
14260 14261 14262 5918 14263 14264 14265 14266 14267 14268 14269 14270 5197 14271 14272 5866 14273 14274 14275
14276 14277 4789 14278 14279 14280 14281 14282 14283 14284 14285 14286 14287 14288 14289 14290 14291 14292 14293
14294 14295 14296 14297 14298 14299 14300 14301 14302 5241 14303 14304 14305 14306 14307 14308 14309 14310 14311
14312 14313 14314 6058 14315 14316 14317 5642 14318 14319 14320 14321 14322 14323 14324 14325 14326 14327 6040
14328 14329 5569 14330 14331 14332 14333 14334 14335 14336 14337 14338 4631 5315 14339 14340 14341 14342 14343
14344 14345 14346 14347 14348 14349 14350 14351 14352 14353 14354 14355 14356 14357 14358 14359 14360 14361 14362
14363 14364 5534 14365 14366 4775 14367 5993 14368 14369 14370 14371 14372 14373 6085 14374 14375 14376 5337
14377 5708 14378 14379 14380 14381 14382 14383 14384 14385 6069 14386 14387 14388 14389 14390 14391 14392 14393
14394 14395 14396 14397 14398 14399 14400 14401 14402 14403 14404 14405 14406 14407 14408 14409 5529 14410 14411
14412 14413 14414 14415 14416 14417 14418 14419 14420 14421 14422 14423 14424 14425 14426 14427 14428 5755 5732
14429 5356 14430 14431 14432 5192 14433 14434 14435 5733 14436 5351 14437 14438 14439 14440 14441 14442 14443
14444 14445 14446 14447 5598 4816 14448 14449 14450 14451 5603 14452 14453 14454 5448 14455 14456 14457 14458
14459 14460 14461 14462 14463 14464 14465 14466 14467 14468 14469 14470 5765 5370 4965 4715 14471 5416 14472
14473 14474 14475 14476 4805 14477 14478 14479 14480 14481 14482 14483 14484 14485 14486 14487 14488 14489 5503
14490 14491 14492 5982 14493 4678 14494 14495 14496 14497 14498 14499 14500 14501 14502 14503 14504 14505 14506
14507 14508 5822 14509 14510 14511 14512 14513 14514 14515 14516 5591 5361 14517 14518 14519 14520 14521 14522
14523 14524 14525 14526 14527 14528 14529 14530 14531 14532 14533 14534 14535 14536 14537 14538 14539 14540 5278
14541 14542 14543 14544 14545 14546 14547 14548 14549 14550 14551 14552 14553 14554 14555 14556 14557 14558 14559
14560 14561 14562 14563 4921 14564 14565 14566 14567 14568 14569 14570 14571 14572 14573 14574 14575 4988 14576
14577 14578 14579 14580 14581 14582 14583 14584 14585 5513 14586 14587 14588 14589 14590 14591 5875 5050 14592

14593 14594 14595 14596 14597 14598 14599 14600 14601 14602 14603 14604 5572 14605 14606 14607 14608 14609 14610
5364 14611 14612 4645 14613 14614 14615 14616 14617 14618 14619 14620 14621 14622 14623 14624 14625 14626 14627
14628 14629 14630 14631 14632 14633 14634 14635 14636 14637 14638 14639 14640 14641 14642 14643 14644 14645 14646
14647 14648 14649 14650 14651 14652 14653 14654 14655 14656 14657 14658 14659 14660 14661 14662 14663 5512 14664
14665 14666 14667 14668 5836 14669 14670 14671 14672 5737 14673 14674 14675 14676 4842 14677 14678 14679 14680
14681 14682 4737 14683 14684 4833 14685 14686 14687 14688 5359 14689 14690 14691 14692 14693 14694 14695 14696
4650 14697 5293 14698 14699 14700 14701 14702 14703 5063 14704 5941 14705 14706 14707 14708 5938 14709 14710
14711 14712 5710 14713 14714 14715 14716 14717 14718 6086 14719 14720 14721 14722 14723 14724 14725 14726 14727
14728 14729 14730 14731 14732 14733 14734 14735 14736 14737 14738 14739 14740 14741 14742 14743 14744 14745 14746
14747 14748 14749 14750 14751 14752 14753 14754 14755 14756 14757 14758 14759 5165 14760 14761 14762 14763 14764
14765 14766 6048 14767 14768 14769 6071 14770 14771 14772 14773 14774 14775 14776 14777 14778 14779 14780 14781
14782 6073 14783 14784 14785 14786 14787 14788 6101 14789 14790 4854 14791 14792 14793 14794 14795 14796 14797
14798 4617 14799 5753 14800 14801 14802 14803 14804 14805 14806 14807 14808 14809 14810 14811 14812 14813 14814
14815 14816 14817 14818 14819 14820 14821 14822 14823 14824 14825 4598 14826 14827 5233 14828 14829 14830 14831
14832 14833 14834 14835 14836 14837 14838 14839 14840 5017 14841 14842 14843 14844 14845 14846 14847 14848 14849
5107 14850 14851 14852 14853 14854 14855 14856 14857 14858 14859 14860 14861 14862 14863 14864 14865 14866 14867
5752 5511 5522 14868 14869 14870 14871 14872 14873 14874 14875 5313 14876 14877 14878 14879 14880 14881 14882
4796 14883 14884 14885 14886 14887 4840 14888 14889 14890 14891 14892 14893 14894 14895 14896 14897 14898 14899
14900 14901 14902 14903 14904 14905 14906 14907 14908 14909 14910 14911 14912 14913 5152 14914 14915 14916 4687
14917 14918 14919 14920 4722 14921 14922 14923 14924 5537 14925 14926 14927 14928 14929 14930 14931 14932 14933
14934 14935 14936 14937 14938 14939 14940 14941 14942 14943 14944 14945 14946 14947 14948 14949 14950 14951 14952
14953 14954 14955 14956 14957 14958 14959 4951 4759 14960 14961 14962 14963 14964 5144 5464 14965 14966 14967
14968 14969 14970 14971 14972 14973 14974 14975 6042 14976 14977 14978 14979 14980 14981 14982 14983 14984 14985
14986 14987 14988 14989 6066 14990 14991 14992 14993 14994 14995 14996 14997 14998 14999 15000 15001 15002 15003
15004 15005 15006 15007 15008 15009 15010 15011 15012 15013 15014 15015 15016 15017 5044 15018 15019 15020 15021
15022 15023 5826 15024 15025 15026 15027 15028 15029 15030 15031 5544 15032 5219 15033 15034 15035 15036 5396
15037 15038 15039 15040 15041 15042 15043 15044 15045 15046 15047 4714 15048 15049 15050 15051 15052 15053 15054
15055 4922 15056 15057 15058 15059 15060 15061 15062 15063 15064 15065 15066 15067 5222 4855 15068 15069 15070
15071 4937 15072 15073 4618 15074 15075 15076 5587 15077 6089 15078 15079 15080 15081 5656 15082 15083 15084
15085 15086 15087 15088 15089 15090 5578 15091 15092 15093 15094 15095 15096 15097 4643 5605 15098 15099 15100
15101 15102 15103 15104 15105 15106 15107 15108 15109 15110 15111 15112 15113 6067 15114 15115 15116 15117 15118
15119 15120 15121 15122 15123 15124 15125 15126 5860 15127 15128 15129 15130 15131 15132 15133 15134 15135 15136
15137 15138 5953 15139 15140 15141 15142 15143 15144 15145 15146 5030 15147 15148 15149 15150 15151 5006 5329
5121 15152 15153 15154 15155 4762 15156 15157 15158 5730 15159 5013 15160 15161 15162 4917 15163 15164 15165
15166 15167 15168 15169 15170 15171 15172 15173 15174 15175 15176 15177 15178 15179 15180 6054 15181 15182 15183
15184 15185 15186 15187 15188 5611 15189 15190 15191 15192 5824 15193 15194 15195 5419 15196 15197 15198 15199
15200 15201 15202 15203 15204 15205 15206 15207 15208 15209 15210 4685 15211 15212 15213 15214 15215 15216 5909
15217 15218 15219 15220 15221 15222 15223 15224 15225 15226 15227 15228 5787 15229 15230 15231 15232 15233 15234
15235 15236 15237 5389 15238 15239 15240 15241 15242 15243 15244 15245 15246 15247 4683 4832 15248 15249 15250
5494 15251 5687 15252 15253 15254 5723 15255 15256 15257 15258 15259 15260 15261 15262 15263 4757 15264 4754
15265 15266 15267 15268 15269 15270 15271 15272 15273 15274 15275 15276 15277 15278 15279 15280 15281 15282 15283
15284 4591 15285 15286 15287 15288 15289 15290 15291 15292 15293 6077 15294 15295 15296 15297 4606 15298 15299
15300 5404 15301 15302 15303 15304 15305 15306 15307 4670 15308 6098 15309 15310 4589 15311 15312 15313 15314
6074 15315 15316 15317 15318 15319 15320 15321 15322 15323 15324 15325 15326 15327 15328 15329 15330 15331 4583
15332 15333 15334 15335 15336 5888 15337 15338 15339 15340 15341 15342 15343 15344 15345 15346 5970 15347 15348
15349 15350 15351 15352 15353 5688 15354 15355 15356 15357 15358 15359 15360 15361 15362 15363 15364 15365 15366
15367 15368 15369 15370 15371 15372 15373 15374 15375 15376 15377 5720 15378 15379 15380 15381 15382 15383 15384
15385 15386 15387 5423 15388 15389 15390 15391 15392 15393 15394 15395 15396 15397 15398 4817 5793 4698 15399
15400 15401 15402 15403 5210 15404 5574 15405 15406 5740 4941 15407 15408 15409 15410 6109 15411 15412 15413
15414 15415 15416 15417 15418 15419 15420 15421 15422 15423 15424 15425 15426 5202 15427 15428 15429 15430 15431
15432 15433 15434 15435 15436 15437 15438 5211 15439 15440 15441 5073 15442 15443 5779 15444 5696 15445 15446
15447 4629 5398 15448 15449 15450 15451 15452 15453 15454 15455 15456 15457 15458 15459 15460 15461 15462 15463
15464 15465 15466 15467 15468 15469 15470 15471 15472 15473 15474 15475 15476 4845 15477 15478 15479 15480 5772
15481 5596 15482 15483 15484 15485 15486 15487 15488 5949 15489 15490 15491 15492 15493 15494 15495 6021 15496
15497 15498 15499 15500 15501 15502 15503 15504 15505 5818 15506 15507 15508 15509 5326 15510 15511 15512 15513
15514 15515 4994 15516 15517 15518 15519 15520 15521 15522 5154 15523 4663 15524 15525 15526 5734 15527 15528
15529 15530 15531 15532 5059 15533 15534 4943 15535 15536 15537 15538 15539 15540 15541 15542 15543 15544 15545
5273 15546 15547 15548 15549 15550 15551 15552 4760 15553 5399 15554 6056 15555 15556 15557 5303 15558 15559
15560 15561 15562 15563 15564 5705 15565 15566 15567 15568 5852 5842 15569 15570 15571 15572 5838 15573 15574
15575 15576 15577 15578 5981 15579 15580 15581 15582 15583 15584 15585 15586 15587 15588 15589 15590 15591 15592
15593 15594 5350 15595 15596 15597 15598 5122 15599 15600 15601 4774 15602 15603 15604 15605 15606 15607 15608
15609 15610 15611 15612 15613 15614 15615 15616 15617 15618 15619 15620 15621 5280 15622 5748 15623 15624 15625
15626 15627 15628 15629 5996 15630 5863 15631 15632 15633 15634 15635 15636 15637 15638 15639 15640 15641 15642
15643 15644 5653 15645 15646 15647 15648 4940 15649 15650 15651 15652 15653 15654 15655 15656 15657 15658 15659
15660 15661 15662 15663 15664 5876 15665 15666 15667 15668 15669 15670 15671 15672 15673 5984 15674 15675 15676
15677 15678 15679 15680 5725 15681 15682 15683 15684 15685 15686 15687 15688 15689 15690 15691 15692 15693 15694
15695 15696 15697 5659 15698 15699 15700 15701 15702 15703 15704 15705 15706 15707 15708 15709 15710 5214 15711
15712 15713 15714 15715 15716 4592 15717 15718 5777 15719 15720 15721 15722 5271 15723 15724 15725 15726 15727
15728 15729 15730 4870 15731 15732 15733 15734 15735 15736 15737 15738 15739 15740 15741 15742 15743 15744 15745
15746 15747 15748 15749 15750 15751 15752 15753 15754 15755 15756 15757 15758 15759 15760 15761 15762 15763 15764
15765 15766 15767 15768 15769 15770 15771 15772 15773 15774 4647 15775 15776 15777 4931 15778 15779 15780 15781
15782 15783 4672 15784 15785 15786 15787 15788 15789 15790 15791 15792 15793 15794 15795 15796 15797 15798 15799
15800 15801 15802 15803 15804 15805 15806 15807 15808 15809 15810 4697 15811 15812 15813 15814 15815 15816 15817
15818 15819 15820 5223 15821 15822 15823 15824 15825 15826 4841 15827 15828 15829 5711 15830 4878 15831 15832
15833 15834 15835 15836 15837 15838 15839 4875 15840 15841 15842 15843 15844 15845 15846 15847 15848 15849 15850
15851 15852 15853 15854 15855 15856 15857 15858 15859 15860 15861 15862 15863 4721 4807 15864 15865 15866 15867

15868 15869 15870 5099 5421 15871 15872 15873 15874 15875 15876 15877 15878 15879 15880 15881 15882 15883 15884
15885 4749 15886 15887 15888 15889 15890 15891 15892 15893 15894 15895 4682 15896 15897 15898 15899 15900 15901
15902 5853 15903 15904 5521 15905 15906 5770 15907 15908 15909 15910 15911 15912 15913 15914 15915 15916 15917
5565 15918 15919 15920 15921 15922 15923 15924 15925 15926 15927 15928 15929 15930 15931 5991 15932 15933 15934
15935 15936 15937 15938 15939 15940 5879 15941 15942 15943 15944 15945 15946 15947 15948 15949 15950 15951 15952
15953 15954 4621 15955 15956 15957 15958 4744 15959 15960 15961 5584 15962 15963 15964 15965 15966 15967 4788
15968 15969 5220 15970 15971 15972 15973 15974 15975 15976 15977 15978 15979 5415 5798 15980 15981 15982 15983
15984 15985 15986 15987 15988 15989 15990 15991 15992 15993 15994 15995 15996 15997 15998 5254 15999 16000 16001
16002 16003 16004 16005 16006 16007 16008 16009 16010 16011 16012 5029 16013 16014 16015 16016 16017 16018 4880
16019 5510 16020 16021 16022 16023 5914 16024 16025 5332 16026 5485 16027 16028 5456 16029 16030 5707 16031
16032 16033 16034 16035 16036 16037 16038 16039 16040 5317 16041 16042 16043 4966 16044 6068 16045 16046 16047
16048 16049 16050 16051 16052 16053 5102 16054 16055 16056 16057 16058 16059 16060 16061 16062 16063 16064 16065
16066 16067 16068 16069 16070 16071 16072 16073 16074 16075 16076 16077 5376 16078 16079 16080 16081 16082 16083
16084 16085 16086 16087 16088 5032 5305 16089 16090 16091 16092 16093 16094 16095 16096 16097 16098 16099 16100
5463 16101 16102 16103 16104 5228 4895 16105 16106 16107 16108 16109 16110 16111 16112 16113 16114 16115 16116
16117 16118 16119 16120 4997 16121 16122 4864 16123 16124 16125 16126 16127 16128 16129 5618 5467 6065 16130
16131 16132 16133 16134 16135 16136 16137 16138 16139 16140 16141 16142 16143 16144 16145 16146 16147 16148 16149
16150 16151 16152 16153 16154 16155 16156 16157 16158 16159 16160 16161 16162 16163 5480 16164 16165 16166 16167
16168 16169 16170 16171 16172 16173 16174 16175 5078 16176 16177 16178 16179 16180 16181 16182 16183 16184 16185
16186 5782 16187 6049 16188 16189 16190 16191 16192 5243 16193 16194 16195 16196 16197 4839 16198 16199 16200
16201 16202 16203 16204 16205 16206 16207 16208 16209 16210 16211 4596 16212 16213 16214 16215 16216 16217 16218
16219 16220 16221 16222 16223 5489 16224 16225 16226 5391 16227 16228 16229 16230 16231 16232 16233 16234 16235
16236 16237 16238 16239 16240 16241 16242 4574 5160 5340 16243 16244 16245 16246 16247 5458 16248 16249 5692
16250 16251 16252 16253 16254 16255 16256 16257 16258 16259 5543 16260 16261 5333 16262 16263 16264 16265 16266
16267 16268 16269 16270 5810 16271 16272 16273 16274 16275 16276 16277 16278 16279 16280 16281 16282 16283 16284
16285 16286 16287 16288 16289 16290 16291 16292 16293 16294 16295 16296 16297 16298 16299 4916 5433 5872 16300
16301 16302 16303 16304 16305 16306 16307 16308 16309 16310 16311 16312 16313 6053 16314 16315 6016 16316 16317
16318 16319 16320 16321 16322 5365 16323 16324 4978 16325 16326 16327 16328 16329 16330 16331 16332 5246 4790
16333 16334 5033 16335 16336 16337 16338 16339 16340 16341 5270 16342 16343 16344 16345 16346 16347 16348 16349
16350 16351 16352 5069 16353 5401 16354 16355 16356 5615 16357 16358 16359 16360 4977 16361 16362 16363 16364
16365 16366 16367 16368 16369 16370 16371 16372 16373 16374 16375 16376 16377 16378 16379 16380 16381 16382 16383
16384 5320 16385 16386 4733 16387 16388 5417 16389 16390 16391 16392 16393 16394 16395 16396 16397 16398 16399
16400 16401 16402 16403 16404 16405 16406 16407 4826 16408 16409 4616 16410 16411 16412 16413 16414 16415 16416
16417 16418 16419 16420 5294 16421 5977 16422 5796 16423 16424 16425 16426 16427 16428 16429 16430 16431 5469
16432 16433 16434 16435 16436 16437 16438 16439 16440 16441 16442 16443 16444 16445 16446 16447 16448 16449 16450
5832 16451 16452 16453 16454 16455 16456 16457 16458 16459 16460 16461 16462 16463 16464 16465 5669 16466 16467
16468 16469 16470 16471 16472 16473 6017 16474 16475 16476 16477 16478 5959 16479 16480 16481 16482 16483 16484
16485 16486 16487 16488 16489 5153 16490 16491 16492 16493 16494 16495 16496 4706 16497 16498 16499 16500 16501
5498 16502 16503 16504 16505 5238 16506 16507 16508 16509 16510 16511 16512 16513 16514 16515 16516 16517 16518
16519 16520 16521 16522 16523 16524 16525 16526 16527 16528 16529 16530 16531 16532 16533 6062 16534 16535 16536
16537 16538 5634 16539 16540 16541 16542 5314 16543 16544 16545 16546 16547 16548 5409 16549 16550 5944 4767
16551 16552 16553 16554 16555 5547 16556 16557 16558 16559 16560 4627 16561 16562 16563 5393 16564 16565 16566
16567 16568 16569 16570 16571 16572 16573 16574 16575 16576 16577 16578 16579 16580 16581 16582 16583 5766 5585
16584 16585 5384 5207 16586 16587 16588 16589 16590 16591 16592 16593 5090 16594 16595 16596 16597 5104 16598
16599 16600 16601 16602 16603 16604 16605 16606 16607 16608 16609 16610 16611 16612 16613 16614 16615 6018 16616
16617 16618 16619 16620 4848 16621 16622 16623 16624 16625 16626 16627 16628 16629 16630 16631 16632 16633 16634
16635 16636 16637 16638 16639 16640 16641 16642 16643 16644 16645 16646 16647 16648 5662 16649 16650 16651 16652
16653 16654 16655 16656 16657 16658 16659 16660 16661 16662 16663 16664 16665 16666 16667 16668 16669 4601 16670
16671 16672 16673 16674 16675 16676 16677 16678 16679 16680 16681 16682 16683 16684 16685 16686 16687 16688 16689
16690 16691 16692 16693 16694 5484 16695 5877 16696 16697 16698 16699 16700 16701 16702 16703 16704 16705 16706
16707 16708 16709 16710 16711 16712 16713 16714 16715 16716 16717 16718 5381 16719 16720 16721 16722 16723 16724
16725 16726 16727 16728 16729 16730 16731 16732 16733 6002 16734 16735 16736 16737 16738 16739 5568 16740 16741
16742 5221 16743 16744 16745 16746 16747 16748 16749 4660 16750 16751 16752 16753 16754 16755 16756 16757 16758
5997 16759 16760 16761 16762 16763 5633 16764 16765 6019 16766 16767 16768 5052 16769 16770 16771 16772 16773
16774 16775 16776 16777 16778 16779 16780 16781 16782 16783 16784 16785 16786 16787 16788 16789 16790 16791 16792
16793 16794 16795 16796 16797 16798 16799 16800 16801 16802 16803 16804 16805 16806 16807 16808 16809 16810 16811
16812 16813 16814 16815 16816 16817 16818 16819 16820 16821 16822 16823 5960 16824 16825 16826 16827 16828 16829
16830 6043 16831 16832 5855 16833 16834 16835 16836 16837 5134 16838 16839 16840 16841 16842 16843 16844 16845
4728 16846 16847 16848 16849 16850 16851 16852 16853 16854 16855 6103 16856 16857 16858 5138 16859 16860 5064
16861 16862 16863 16864 16865 16866 16867 16868 16869 16870 16871 4773 16872 16873 16874 16875 16876 16877 16878
6093 16879 16880 16881 16882 16883 16884 16885 16886 4811 16887 16888 16889 16890 16891 16892 16893 16894 16895
16896 16897 16898 16899 16900 16901 16902 16903 16904 16905 5321 6095 16906 16907 16908 16909 16910 16911 16912
16913 16914 16915 5366 16916 4799 16917 16918 16919 16920 16921 16922 16923 16924 16925 16926 5926 16927 16928
5986 16929 16930 16931 16932 6081 16933 16934 16935 16936 16937 16938 16939 16940 16941 16942 16943 16944 16945
16946 16947 16948 5382 16949 16950 16951 16952 16953 16954 16955 16956 16957 16958 16959 16960 16961 16962 16963
16964 16965 16966 16967 16968 16969 16970 16971 16972 16973 16974 16975 16976 16977 16978 16979 16980 16981 16982
16983 16984 16985 5930 16986 6030 16987 16988 16989 16990 16991 16992 16993 16994 5115 16995 16996 16997 16998
16999 17000 17001 17002 17003 17004 4838 17005 17006 17007 17008 17009 17010 17011 17012 17013 17014 17015 17016
17017 17018 17019 17020 17021 5614 17022 17023 17024 17025 17026 17027 17028 17029 17030 17031 17032 17033 17034
17035 17036 17037 17038 17039 17040 17041 5923 17042 17043 17044 17045 17046 17047 17048 17049 5205 17050 5418
17051 17052 5259 17053 17054 17055 17056 17057 17058 17059 17060 17061 17062 17063 17064 17065 17066 17067 17068
17069 17070 17071 17072 17073 17074 17075 17076 17077 17078 17079 17080 17081 17082 17083 17084 17085 17086 17087
17088 17089 17090 17091 17092 17093 17094 17095 17096 17097 17098 17099 4679 17100 17101 17102 17103 17104 17105
17106 17107 17108 17109 17110 17111 17112 17113 17114 17115 17116 17117 17118 17119 5354 5430 17120 17121 17122
17123 17124 17125 17126 17127 4711 17128 17129 17130 17131 17132 17133 17134 17135 17136 17137 17138 17139 17140

17141 17142 5272 17143 17144 17145 17146 17147 17148 17149 17150 17151 17152 17153 17154 17155 17156 17157 17158 17159 6013 17160 17161



(a) ***(b)***

(c)

Figure III.12. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,* ***(c)*** *Histogrammes de l'image chiffrée.*

III.4.1.2. Description de PosESecL2

Comme la clé générée par notre première alternative codant le premier niveau de sécurisation, est de taille variable d'une donnée à une autre et qui est égale à la taille de la donnée en terme d'éléments (caractères ou pixels), telle que : taille de la donnée = n éléments => taille de la clé générée = n nombres ; alors la résistibilité à l'attaque exhaustive est strictement dépendante de la taille de la donnée. C'est pourquoi, nous proposons, ici, une variante de PosESecL1 et que nous l'appelons PosESecL2 (Position based Evolutionary Secure Level 2).

L'évolution vers la solution optimale (donnée cryptée) par PosESecL2 est assurée par application des opérateurs de reproduction (croisement, mutation) et de sélection des meilleurs individus. Dans ce qui suit nous ne présentons que les phases du processus évolutionnaire faisant la différence entre les deux algorithmes PosESecL1 et PosESecL2. Il s'agit, principalement, des phases de codage et de reproduction. Toutefois, le même mécanisme est adopté pour créer la population initiale en permutant aléatoirement le chromosome codant l'image originale.

a. Codage

Dans le but d'augmenter le niveau de sécurité du premier algorithme décrit dans la section précédente, nous présentons, ici, le nouveau codage proposé donnant lieu à un nouvel algorithme de chiffrement : PosESecL2. Ce dernier opère sur le codage des différents éléments de la donnée à chiffrer : dans le cas d'une donnée texte et de même que pour PosESecL1, le codage utilisé est l'hexadécimal, toutefois, chaque quatre bits des 16 bits codant un élément sont considérés comme un gène. Pour une donnée image, chacune des

composantes R, V et B est considérée comme étant un gène. Ainsi, la taille de la nouvelle clé sera plus grande que celle générée par le premier algorithme. Elle est d'une taille quatre fois plus grande dans le cas de manipulation des données texte et de trois fois plus grande dans le cas de manipulation d'une donnée image. Le codage proposé est résumé à travers la figureIII.13.

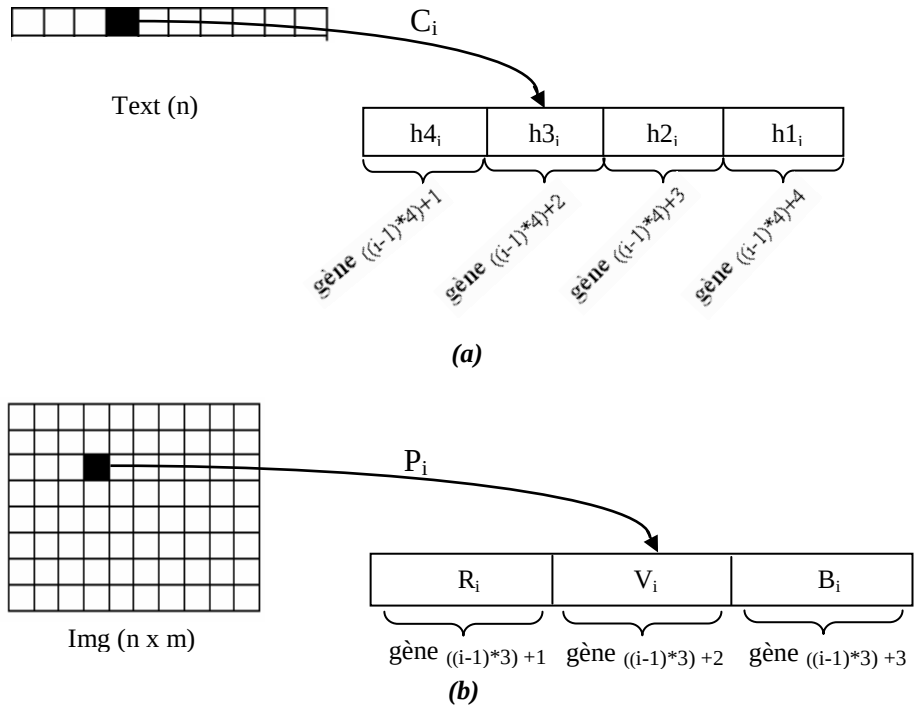


Figure III.13. **(*a*)** *Représentation chromosomale sous PosESecL2 d'un caractère C_i du texte Text(n),* **(*b*)** *Représentation chromosomale sous PosESecL2 d'un pixel P_i de l'image Img(n*m).*

Avec : ($h4_i$ $h3_i$ $h2_i$ $h1_i$) représente le codage Unicode hexadécimal du caractère C_i.

b. Reproduction

Contrairement à l'algorithme PosESecL1, l'algorithme PosESecL2 considère le codage des éléments d'une donnée indépendamment. Ainsi, l'application des opérateurs de croisement et de mutation ne nécessite pas la prise en compte des contraintes à vérifier comme dans le cas de PosESecL1, mais il enrichit d'avantage la population d'individus par de nouvelles caractéristiques.

De même que pour le PosESecL1, l'opérateur de croisement utilisé est le OX « Order Cross-over » appliqué avec un taux fixé expérimentalement et compris entre 60% et 100%. Toutefois, l'opérateur de mutation utilisé est une permutation multipoints pour éviter une convergence trop lente vu la grande différence des tailles des chromosomes dans PosESecL1 et PosESecL2 (le deuxième est d'une taille huit ou trois fois plus grande que le premier). Cet opérateur est appliqué avec un taux, aussi fixé expérimentalement et compris entre 0.1% et 5%.

c. Evaluation des individus

Le choix des individus qui survivront d'une génération à une autre s'effectue selon la fonction objective F, donnée ci-dessous, représentative de l'efficacité des solutions générées sur le problème posé.

$$F(I_i) = \sum_{j=1}^{n \times l} \left| G_{ji} - G_{j0} \right| \quad \text{(III.12)}$$

Avec : G_{ji} est la valeur du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.
l est la taille du codage d'un seul élément de la donnée à chiffrer.

Ainsi, la fonction d'évaluation prendra les deux instanciations suivantes dans le cas de chiffrement de données texte et celui de chiffrement de données images respectivement :

$$F(I_i) = \sum_{j=1}^{n \times 4} \left| G_{ji} - G_{j0} \right| \quad \text{(III.13)}$$

$$F(I_i) = \sum_{j=1}^{n \times 3} \left| G_{ji} - G_{j0} \right| \quad \text{(III.14)}$$

d. Critère d'arrêt

La condition d'arrêt assurant la convergence de l'algorithme PosESecL2 est la même que celle de PosESecL1 puisque le codage des individus dans les deux algorithmes ne diffère que sur le plan contenu de gènes mais les mêmes informations sont présentes dans les deux codages (codage Unicode des caractères du texte à chiffrer ou représentation RVB des pixels de l'image à chiffrer). Autrement dit, c'est la manipulation du codage des éléments de la donnée à chiffrer qui fait la différence entre les deux codages utilisés par les deux algorithmes proposés. Donc, la condition d'arrêt adoptée par ESecL2 dans le cas de chiffrement de données texte et celui de chiffrement de données images, respectivement, est la suivante :

$$0 \le F(I_i) \le F \times 4 \times n \quad / (F)_{16} = (15)_{10} \quad \text{(III.15)}$$

$$0 \le F(I_i) \le 255 \times 3 \times n \quad \text{(III.16)}$$

De même que pour PosESecL1, cette unique condition n'assure pas dans tous les cas la convergence de PosESecL2. C'est pourquoi un nombre maximal de générations sera fixé pour résoudre le problème.

e. Déchiffrement

De même que pour PosESecL1, ce deuxième algorithme utilise le même mécanisme de calcule de clé de chiffrement ; elle change d'un chiffrement à un autre, donc, elle dépend de la donnée à chiffrer et de sa taille. Ainsi, même la taille de la clé de session générée varie d'une donnée à une autre.

f. Réglage des paramètres et résultats

Dans cette section nous présentons les résultats de l'application du deuxième algorithme développé opérant suivant le paramétrage reporté à travers le tableau III.3. À partir des mêmes données originales précédemment utilisées pour tester PosESecL1, nous avons appliqué notre deuxième algorithme PosESecL2 afin d'obtenir les données chiffrées correspondantes. Ces dernières sont celles faisant l'objet des figures III.18.a, III.19.a, III.20.a et III.21.a.

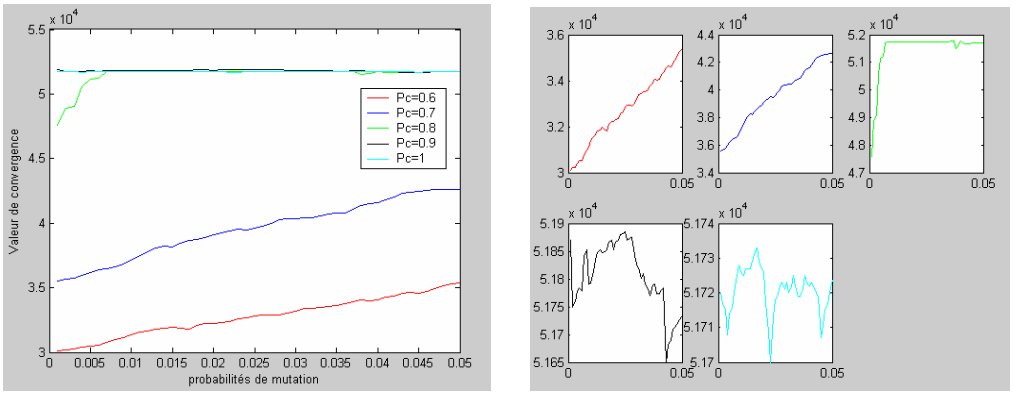


Figure III.14. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

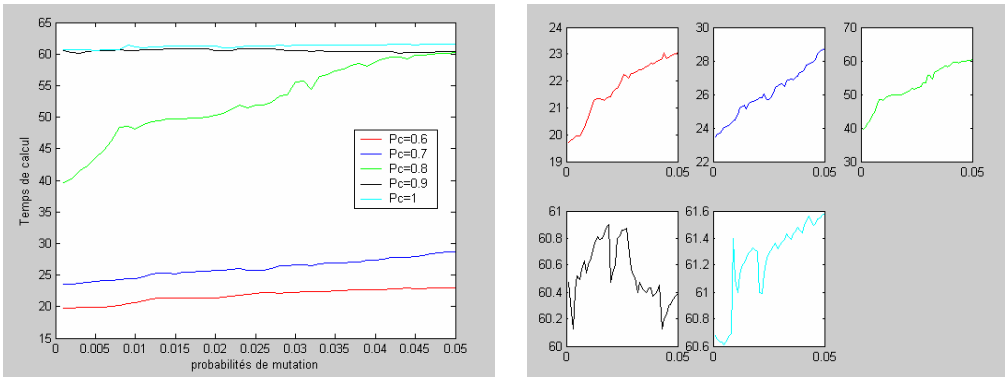


Figure III.15. *Influence des paramètres Pc et Pm sur le temps de calcul.*

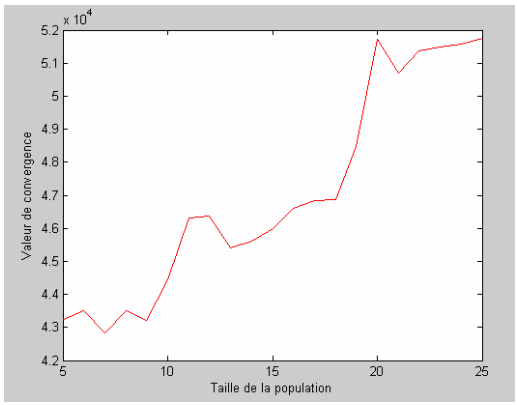


Figure III.16. *Evolution des valeurs de convergence en fonction de la taille de population.*

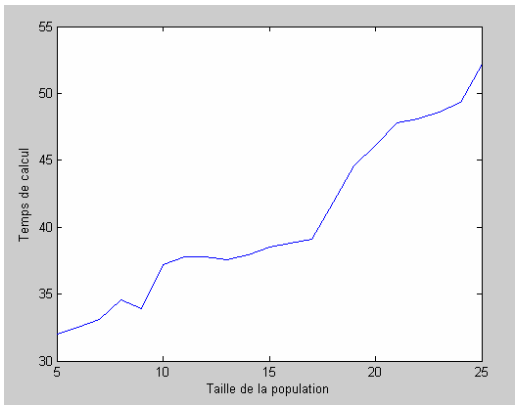


Figure III.17. *Evolution du temps d'exécution en fonction de la taille de population.*

	Valeurs des paramètres de PosESecL2
Taille de la population	20
Nombre de générations	100
P_c	0.8
P_m	0.007

Tableau III.3. *Valeurs adoptées pour les paramètres de PosESecL2.*

L'application de ce deuxième algorithme suivant le paramétrage reporté dans le tableau III.3 sur les mêmes données tests précédentes (Texte 1, Texte 2, Lena et Logo) a permis d'obtenir les résultats résumés à travers le tableau III.4.

D'après ces résultats, il est clair que la taille de la clé augmente proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cette dernière influence la taille des chromosomes résultants du codage utilisé. Donc, c'est la taille des chromosomes augmentant en fonction de la taille des données à chiffrer qui conduit à une augmentation de la taille des clés générées et du temps de calcul correspondant.

		Taille donnée (éléments)	Taille clé (bits)	VC	Temps calcul (s)
Données texte	Texte 1	1084	56368	30729	28.74
	Texte 2	1151	59852	35100	31.5
Données images	Lena	131 X 131	823728	51720	46.15
	Logo	420 X 395	9456300	368712	73.06

Tableau III.4. *Résultats obtenus par PosESecL2.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

´1à1њњ╬‹à+1وî¨à+#cЖéZ0¨£1#§£î¨îغوàî+6+¨1وcњ♀њ{à0é6¨à+ض¨éغ+‹0++qو61ëc¨00c¨î{وî8à‹9¨+њZtt01{î1£#à˚0î♀њ1à+╬0î£01↔
↔¨ààîéë¨+#↔↔¨1´éà0ààà+§8¨£§¨¨0´£è#§é1ë+♀++£11¨{é؟↔1£ëàאַאַ¨++§+à+¨++9§t´£09£و╬+ët؟é#éà0غњ↔6àאַ18وאַ##ë£¨+0
+¨00‹؟+و+9}1¨˚+8؟c8££↔à+11++§#‹îخوZ0؟#ووëé{§1‹+9é86£009وZغ¨c؟}18+î¨cëîé+§אַ0‹0+غو§£§îњ1˚0+1c+אַوàوà+Z#q£18£אַغخ♀و06#
î1¨غs‹1غ´î#é0#њZ╬§£é↔è+1+غ8و£+ëà9{њњњ¨éو¨#و£àt¨é+++00é˚#î1+î#6à‹0#0##àé11+؟‹é#אַЖZ0#+غ££+↔Ж+à1خén£¨¨à#8}وé↔+¨î
╬90+و+Жë¨î‹§1++↔ëéאַ؟+î£àé#ëZ11אַ§#אַ♀à#î£+1£é+1ët↔c¨خ+‹ëאַcوîé#îé؟#؟î1#˚î¨+8#+1+و£éوé╬¨؟1£وZZc8њغ#é10خà#+↔08¨╬1#+
£é0té£+88+#^وאַ9‹î¨î+אַ¨{˚£§£+´6§6╬╬↔+¨њњњњà{6+##و↔↔ë؟+îà+16îњ1¨11£1£0#9+غà§+é£+ë+#11؟£ëî‹î+£‹1+´£00£+غ¨+وé#£19
♀♀c§î+c01àî¨8#8و+╬à§¨{î¨+60¨§6♀0Z8é#1#à{+1£‹à9#1؟à6+¨1אַ£˚6à+1î1ë+؟t9£+îî+80£6é#¨+£¨ZZ1c¨њњ{ëc0♀t0#وt}+0è↔{ècc8î#
£¨£#وàà£#+1£+њЖњ╬à˚116↔1؟+éîZà£c{´+#

(b)

Figure III.18. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($Taille_{Clé}$ = 6,8808 K octets) :

1	450	496	406	440	458	465	500	437	477	491	298	487	444	468	497	441	471	493
494	454	102	225	455	453	484	473	330	429	432	428	447	469	480	436	488	475	74
446	478	490	443	430	486	483	481	466	499	239	449	489	456	120	464	435	498	451
442	479	474	412	434	467	336	485	5	457	101	213	448	470	431	460	463	438	307
476	445	461	427	459	462	290	495	482	472	433	157	439	426	452	740	1330	545	1331
508	1332	975	1333	564	1334	1335	961	818	1336	886	812	1337	908	674	875	1054	1338	808

725	1339	1340	1341	1342	1343	1344	786	732	1226	1297	228	635	1345	1346	863	696	758	1347
689	1348	1349	1350	710	126	855	548	756	667	522	892	604	1351	1352	602	1353	146	920
770	691	678	729	901	1354	876	1025	1355	4	3	6816	609	993	1356	1357	1229	1358	309
1359	1360	502	1361	706	1362	1363	1364	1365	880	911	644	784	1366	773	359	768	1367	811
299	940	578	198	563	1368	149	957	742	668	687	833	738	1369	623	878	852	636	1370
634	253	1371	873	586	1151	836	620	1372	948	1004	928	1373	799	882	830	270	684	847
964	550	759	1374	870	937	1375	765	905	514	1376	528	739	583	959	337	1377	693	752
612	796	540	1378	552	680	1379	510	946	216	637	630	992	127	978	1309	643	1380	516
1381	559	889	1382	797	567	1383	645	1384	820	949	844	747	1385	866	1386	1387	1160	848
652	384	1388	560	976	985	603	1003	942	501	1389	633	841	974	1093	1390	657	39	912
1391	982	1392	1393	1394	699	750	894	794	1395	1396	932	864	1397	987	716	807	509	676
708	505	1398	838	1399	231	1400	315	339	523	556	326	673	1020	737	599	577	791	751
1329	613	898	1401	1402	419	1001	1403	1404	900	1405	1406	492	1407	536	766	1408	1409	702
787	1410	187	924	1411	845	793	1412	1138	804	1413	1414	1415	546	731	1416	18	1417	66
1418	650	511	662	1288	971	715	817	962	694	1419	968	631	966	537	654	364	520	1104
850	1193	150	981	782	1420	543	915	827	1421	995	1422	843	542	934	1423	1424	775	517
642	709	9	1425	607	772	529	80	994	842	819	712	639	656	2	989	1426	938	610
851	735	666	832	651	541	600	227	241	884	781	895	175	515	417	1237	525	825	734
965	278	890	138	1427	1428	1429	596	1430	553	1293	923	589	730	1431	555	917	1261	663
1432	1433	1000	58	130	1434	790	672	1435	534	1436	1437	1169	44	805	641	800	1438	958
1291	954	1439	1440	1441	1442	835	1443	640	595	1444	933	717	700	888	1445	538	601	616
1446	675	617	701	972	1447	757	909	1448	918	931	1171	704	660	573	998	1449	871	771
1450	763	906	1451	185	1116	557	1452	1453	711	403	916	983	1454	769	1455	221	979	713
1456	846	903	813	839	743	558	910	1295	585	1457	597	1458	1459	1460	967	1461	1462	310
362	313	697	1463	1464	935	719	16	105	856	135	921	952	726	896	1240	1465	705	927
930	1466	397	1467	720	647	980	1468	829	575	950	1469	580	883	664	527	746	323	780
561	614	877	611	828	776	1470	626	1471	570	681	688	569	622	618	879	1472	810	973
572	1473	314	1474	615	683	1475	1476	108	627	1264	665	343	714	382	52	801	1287	152
1477	521	579	28	1478	1479	721	532	1480	1481	549	951	874	46	91	744	592	284	1210
254	1482	206	1483	606	646	945	872	767	320	774	55	348	1484	593	698	535	1485	659
1486	1155	1487	1488	1489	670	1490	990	1491	240	518	815	1492	653	1493	754	899	857	755
798	1494	887	861	1495	143	897	544	996	503	1496	1497	1243	1498	1499	824	1500	1501	1502
533	551	840	977	1503	837	590	1504	988	1505	1506	1507	565	963	788	727	777	1508	1509
809	1510	109	1511	174	1512	869	547	1234	986	904	991	638	1513	999	598	1514	566	722
748	1515	513	802	524	96	1516	1517	1518	919	245	679	507	255	690	1519	733	506	939
922	685	860	1520	853	865	628	504	1521	1522	1523	941	1037	1524	686	1525	355	1526	677
1527	1528	295	914	1189	834	749	661	718	581	1529	764	1530	1531	785	280	823	854	1532
1533	244	859	658	519	648	649	1534	831	984	792	608	554	122	1535	1536	1537	955	723
1538	947	1539	997	1540	568	1541	582	1031	1542	1543	692	1207	588	821	956	671	418	1544
1545	760	1546	943	1547	1548	902	1549	703	1550	1289	695	574	728	745	41	594	970	587
1551	1552	233	1553	1554	655	1555	1556	421	1557	1558	893	562	803	7	814	1559	891	1560
1142	512	1561	629	1562	311	936	73	530	1563	1564	1565	783	1566	1567	1568	137	881	669
624	926	1569	741	415	736	753	953	302	1073	318	1570	779	1571	162	1572	1573	867	266
1574	605	1064	1575	334	90	12	1200	576	1576	1577	969	762	907	1038	619	862	1272	1578
1579	571	822	526	1580	45	49	913	795	1315	761	960	849	944	95	625	1581	1582	201
1583	1005	929	1584	1585	632	531	925	591	1586	789	868	300	682	1587	826	584	885	707
539	1002	806	621	778	724	858	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
1600	1159	1276	1601	1103	1602	1603	1604	1605	1606	1607	128	1608	1609	1610	1611	1612	1613	1614
34	1615	1328	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
1632	1317	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649
1062	1650	1651	1652	1653	1654	1655	1656	1657	1246	1658	1659	1660	1661	1662	1663	1664	1132	1665
1666	1667	349	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683
1684	199	1685	1083	1686	1687	1688	1689	1690	1691	1692	1211	166	1693	1694	1695	1696	1697	291
1247	1153	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1084	1709	1710	1711	140	1712
1713	1714	196	1715	1716	141	1717	1718	1719	1720	1721	1722	1723	1724	1327	287	1725	1726	1009
1727	1728	1729	1730	235	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744
1745	1746	1747	1748	1749	1750	1167	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762
1763	1764	1765	383	57	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779
1780	1144	1781	1185	1782	1783	1784	1294	1785	1786	1324	1787	1156	1788	1789	1790	1791	1792	1793
1195	1281	1794	1795	1796	1797	1798	1799	1800	1014	1801	1802	246	1803	1804	1805	1806	1145	1807
1808	1809	1810	1811	79	1812	1813	1814	1050	1274	286	1815	1816	1817	1818	1819	1820	1821	1822
1823	1824	1825	374	1186	1826	1827	395	1828	1112	1829	1830	112	1831	1832	82	1833	1834	1835
1836	1837	1838	1839	1840	1091	1841	1842	1843	1844	1845	1846	1847	1848	269	1849	1850	17	1275
1851	1177	204	1852	1853	1854	1855	1856	1857	1858	1859	1860	1216	1861	1862	1863	1864	1865	142
1866	1867	1090	1868	1869	1870	56	27	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1039
1881	1882	1883	1884	1075	1885	1886	154	1887	1888	1115	1889	1063	1890	1891	1892	208	1893	1894
1895	1896	1897	1898	1899	1900	1901	1902	1903	1107	1904	1905	1906	1907	1908	1909	1910	1911	1912
1913	1914	1915	1916	125	1917	1918	169	1919	1920	1921	1922	1923	1924	1925	1926	1927	398	1928
1929	1930	1931	1267	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	352	1943	1051	1944
1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	8	1957	1958	1959	1960	1961	1962
293	1963	1964	1965	1966	1967	1968	53	1969	1970	1296	1971	1015	1972	1973	1974	1975	1976	1977
1978	1979	1074	1980	1279	1981	312	1072	1982	1983	1984	1985	1986	264	1987	1988	1989	265	1990
1991	1992	1993	1994	1995	1996	1997	1998	1999	1017	2000	2001	2002	252	2003	2004	2005	342	2006
2007	2008	2009	2010	2011	1130	2012	2013	2014	2015	2016	2017	2018	2019	2020	1041	2021	2022	2023
1058	133	2024	2025	2026	2027	2028	2029	2030	2031	2032	1184	2033	2034	2035	2036	2037	1187	2038
2039	1136	2040	381	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055

1304	305	410	2056	2057	2058	2059	2060	10	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070
2071	2072	2073	2074	35	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088
2089	2090	2091	283	2092	2093	2094	2095	2096	2097	2098	1069	2099	2100	2101	170	2102	2103	2104
2105	2106	2107	2108	2109	2110	1111	2111	297	226	2112	2113	64	2114	2115	267	2116	33	2117
2118	2119	2120	2121	2122	2123	2124	288	2125	2126	2127	2128	2129	2130	2131	176	2132	2133	2134
2135	2136	2137	214	2138	2139	2140	2141	2142	2143	2144	1179	356	2145	2146	2147	2148	2149	2150
2151	2152	2153	2154	2155	2156	2157	2158	2159	99	2160	2161	2162	2163	2164	2165	139	2166	2167
2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	89	2182	2183	2184	153
2185	2186	2187	54	2188	2189	2190	2191	2192	2193	2194	186	2195	1125	2196	2197	2198	2199	2200
1057	2201	2202	2203	2204	2205	261	2206	1158	2207	2208	2209	2210	2211	1087	2212	2213	2214	2215
1273	2216	2217	2218	2219	2220	2221	2222	2223	1230	2224	2225	2226	2227	2228	2229	229	2230	2231
2232	2233	2234	2235	2236	2237	2238	2239	2240	86	2241	2242	1266	2243	2244	103	2245	2246	2247
2248	2249	2250	2251	1235	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	1231	1137	2262	2263
2264	2265	2266	2267	2268	219	2269	2270	376	2271	2272	2273	2274	2275	2276	2277	1113	2278	2279
2280	2281	304	2282	2283	2284	2285	2286	72	2287	2288	2289	2290	2291	394	2292	2293	2294	24
2295	2296	2297	2298	1258	165	2299	1259	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
107	2311	2312	2313	2314	2315	1110	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327
2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	136
2346	1008	1024	2347	2348	2349	2350	2351	2352	2353	404	2354	1232	2355	2356	2357	2358	2359	2360
2361	2362	1102	2363	2364	2365	2366	371	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377
2378	2379	2380	2381	2382	2383	1217	1046	2384	2385	2386	2387	2388	2389	2390	2391	2392	123	2393
2394	2395	379	2396	2397	2398	2399	2400	1203	2401	274	1223	1071	2402	388	2403	2404	2405	1081
2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	1019	2416	2417	2418	2419	2420	2421	2422	2423
2424	2425	2426	2427	2428	2429	2430	2431	2432	40	2433	2434	2435	2436	2437	2438	2439	2440	1162
2441	2442	2443	1109	2444	306	2445	1325	2446	2447	2448	2449	2450	2451	2452	2453	2454	134	2455
2456	2457	76	2458	2459	2460	43	2461	2462	2463	2464	2465	129	2466	2467	2468	2469	2470	2471
399	191	2472	2473	2474	2475	2476	2477	405	2478	2479	2480	2481	2482	2483	2484	2485	2486	2487
2488	294	2489	2490	2491	2492	2493	2494	2495	2496	177	2497	2498	276	247	2499	2500	256	2501
2502	2503	2504	2505	2506	2507	389	2508	2509	2510	2511	2512	2513	2514	2515	2516	1012	346	2517
2518	350	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535
2536	2537	2538	2539	2540	2541	2542	2543	1222	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553
1206	2554	2555	2556	369	2557	2558	2559	292	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569
2570	2571	2572	2573	2574	1028	2575	1205	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586
104	2587	2588	2589	2590	2591	2592	2593	2594	328	2595	2596	1120	2597	2598	2599	2600	2601	2602
2603	2604	2605	2606	2607	316	236	2608	2609	1286	2610	2611	2612	2613	2614	2615	2616	2617	2618
411	2619	2620	2621	2622	2623	2624	2625	2626	1016	2627	365	2628	1214	2629	2630	2631	2632	2633
2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652
2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	19	2664	2665	2666	2667	2668	2669	2670
2671	1176	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688
2689	366	2690	2691	2692	2693	2694	2695	2696	2697	2698	271	2699	172	2700	1114	2701	2702	2703
2704	2705	193	2706	2707	2708	2709	205	2710	2711	1122	2712	2713	2714	2715	2716	1248	2717	2718
1150	2719	2720	2721	2722	2723	2724	189	373	2725	2726	2727	2728	2729	1183	2730	2731	2732	2733
2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	1182	2748	2749	2750	1199
2751	2752	1036	2753	420	2754	2755	145	1148	2756	2757	2758	2759	2760	1194	2761	2762	2763	1106
2764	2765	2766	115	2767	2768	368	2769	2770	163	2771	2772	2773	2774	2775	390	2776	2777	2778
2779	111	113	2780	248	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794
2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	173
2813	2814	2815	168	2816	2817	2818	2819	1061	2820	2821	1249	2822	1094	2823	159	2824	2825	2826
2827	2828	2829	2830	2831	2832	2833	2834	2835	1146	2836	1022	2837	2838	2839	1027	2840	2841	2842
48	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	360	2857	2858	2859
2860	2861	2862	2863	2864	1209	22	178	2865	2866	1181	2867	2868	2869	2870	1085	32	14	2871
1174	2872	2873	2874	2875	2876	1307	1134	1096	2877	1218	2878	2879	2880	2881	2882	2883	2884	2885
2886	329	2887	2888	144	2889	2890	2891	2892	2893	2894	2895	2896	1066	2897	2898	2899	2900	2901
2902	2903	2904	1244	409	2905	400	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917
2918	2919	1170	2920	2921	2922	132	42	2923	2924	2925	2926	2927	2928	2929	2930	77	2931	2932
211	1143	1121	242	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943	1192	2944	2945	422
2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	87	2961	2962	2963
2964	2965	2966	2967	1040	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	322	2978	2979	2980
2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999
3000	3001	1013	3002	3003	37	3004	3005	3006	3007	3008	1300	3009	3010	1250	3011	3012	3013	257
3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029	1166	3030	3031
3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	1262	3043	3044	1313	1149	372	1078	3045
3046	3047	3048	3049	3050	3051	3052	1215	3053	3054	3055	3056	3057	3058	3059	3060	3061	1265	3062
3063	3064	181	3065	3066	3067	268	3068	3069	3070	3071	182	3072	3073	3074	3075	3076	3077	3078
401	3079	3080	3081	3082	161	3083	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	1178
3095	1268	3096	3097	3098	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	1055	3110	3111
3112	3113	3114	1256	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	1204	3125	3126	3127	3128
3129	3130	3131	3132	3133	3134	3135	3136	3137	3138	50	3139	3140	1131	3141	3142	3143	3144	3145
3146	3147	3148	3149	151	3150	3151	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	1320
3163	317	1070	327	3164	3165	1242	3166	3167	1299	402	3168	3169	367	3170	3171	3172	3173	3174
3175	3176	3177	3178	131	3179	94	3180	3181	3182	1154	3183	1047	3184	3185	3186	1168	3187	3188
363	3189	3190	3191	3192	3193	3194	258	3195	3196	121	3197	3198	93	3199	319	3200	1099	3201
3202	3203	202	3204	3205	354	416	3206	3207	3208	3209	3210	3211	1316	3212	3213	3214	3215	3216
3217	3218	3219	3220	3221	303	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	243	3233
1269	3234	1233	3235	3236	3237	85	3238	3239	3240	3241	3242	3243	1319	1213	3244	3245	3246	3247
3248	218	1225	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	1048	3259	3260	119	3261	3262

3263 3264 3265 3266 3267 3268 3269 238 3270 3271 3272 3273 3274 3275 3276 223 3277 1032 3278
3279 1101 3280 3281 3282 3283 1076 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 1260 3294
3295 3296 3297 3298 3299 3300 279 3301 3302 3303 3304 3305 118 3306 3307 3308 1129 3309 3310
3311 97 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328
3329 1135 3330 3331 3332 3333 184 3334 370 3335 3336 224 3337 3338 3339 1191 3340 3341 3342
3343 110 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 1117
1254 3360 3361 3362 3363 3364 3365 188 3366 3367 3368 1157 1147 3369 155 3370 3371 3372 1282
21 3373 3374 3375 3376 1241 3377 3378 425 3379 3380 3381 3382 3383 1190 3384 3385 1033 3386
3387 3388 3389 3390 3391 413 3392 3393 1065 3394 3395 285 1228 3396 3397 3398 3399 3400 3401
3402 3403 3404 194 3405 3406 3407 321 230 1098 3408 3409 3410 3411 3412 3413 197 3414 3415
3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 1251 3428 3429 3430 3431 3432 3433
1152 3434 3435 353 1030 3436 1292 3437 3438 215 1180 3439 3440 3441 3442 3443 3444 3445 3446
3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 164 3457 3458 3459 361 3460 3461 3462 3463
3464 3465 3466 3467 25 3468 3469 331 3470 3471 3472 3473 3474 3475 1270 3476 195 3477 3478
3479 3480 3481 3482 3483 3484 3485 3486 20 340 3487 3488 3489 3490 84 3491 3492 3493 3494
3495 3496 1290 209 1108 1006 3497 3498 3499 3500 3501 190 3502 3503 3504 3505 3506 1018 3507
3508 3509 3510 1224 1053 1124 207 3511 212 3512 3513 3514 3515 1285 114 3516 3517 3518 3519
324 3520 3521 3522 3523 3524 3525 3526 1221 3527 3528 158 3529 3530 3531 3532 3533 1236 3534
335 3535 3536 1172 3537 3538 3539 3540 3541 3542 1035 3543 1318 1119 1220 1011 3544 3545 3546
3547 3548 3549 3550 3551 3552 3553 3554 424 3555 3556 387 3557 3558 3559 3560 3561 3562 3563
3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 156 3574 3575 3576 3577 3578 3579 3580 3581
3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 1118 3592 1303 3593 3594 3595 3596 31 263
3597 3598 3599 3600 3601 3602 3603 69 3604 3605 3606 3607 3608 3609 1133 3610 3611 3612 3613
1196 3614 3615 3616 1219 1173 3617 396 1188 3618 3619 3620 3621 282 3622 3623 3624 3625 3626
3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 1239 1198 3637 3638 3639 408 3640 3641 3642
1049 1089 3643 3644 1306 3645 3646 3647 3648 3649 296 3650 3651 3652 1271 3653 3654 3655 160
3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674
3675 3676 3677 3678 3679 3680 1305 3681 3682 3683 1284 3684 3685 3686 3687 3688 3689 3690 3691
3692 3693 3694 3695 3696 3697 3698 15 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 217
3709 3710 1127 3711 1298 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725
1140 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 1067 3738 3739 3740 3741 180
3742 3743 3744 1023 3745 3746 3747 3748 3749 3750 3751 3752 1161 3753 332 3754 3755 3756 3757
1252 3758 3759 3760 3761 3762 171 3763 3764 3765 3766 3767 3768 100 3769 3770 3771 3772 3773
3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 192 3787 3788 3789 3790 3791
3792 3793 3794 3795 3796 3797 262 1082 333 3798 3799 124 3800 3801 3802 1312 1052 3803 3804
3805 3806 391 3807 3808 1278 3809 1042 83 3810 347 3811 3812 3813 70 3814 3815 3816 3817
3818 3819 3820 3821 3822 3823 3824 3825 1126 3826 3827 3828 3829 3830 3831 3832 3833 3834 98
3835 51 81 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 1255 3846 3847 67 3848 3849
3850 3851 3852 3853 3854 3855 1208 3856 3857 30 3858 3859 3860 3861 65 301 47 1021 3862
3863 3864 3865 3866 3867 3868 29 3869 3870 3871 3872 3873 3874 3875 1253 3876 3877 3878 3879
1321 3880 3881 3882 3883 63 3884 71 3885 3886 3887 3888 275 1175 3889 3890 3891 3892 3893
3894 3895 1045 3896 3897 3898 3899 423 260 3900 3901 3902 308 3903 3904 3905 3906 26 59
3907 3908 3909 1322 3910 3911 3912 3913 3914 3915 1197 1308 3916 3917 3918 3919 3920 3921 3922
3923 3924 1086 3925 3926 3927 3928 1263 68 1128 272 3929 3930 3931 3932 3933 167 3934 3935
3936 3937 3938 3939 3940 3941 3942 147 3943 1034 3944 3945 3946 3947 3948 3949 3950 3951 3952
3953 3954 3955 3956 3957 3958 3959 3960 357 3961 3962 3963 3964 3965 3966 3967 1165 1277 3968
38 3969 3970 232 3971 3972 3973 3974 3975 277 3976 3977 3978 3979 3980 3981 3982 3983 3984
3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003
13 4004 4005 4006 1202 4007 4008 4009 4010 4011 4012 4013 4014 4015 1314 4016 4017 4018 1326
4019 4020 4021 1141 4022 250 4023 4024 4025 4026 4027 4028 4029 4030 1201 4031 1227 4032 386
4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 1095 179 4043 4044 4045 1080 4046 4047 4048
4049 75 1029 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 237 4060 4061 4062 4063 4064
4065 4066 203 4067 4068 1301 414 249 4069 4070 380 4071 4072 1283 4073 4074 4075 78 4076
378 4077 61 4078 4079 4080 4081 4082 4083 4084 88 4085 4086 325 4087 1044 4088 4089 4090
200 4091 377 4092 4093 1164 4094 4095 4096 4097 222 4098 4099 4100 4101 4102 1059 4103 4104
4105 4106 4107 1026 4108 4109 1100 4110 358 4111 4112 1238 4113 1139 4114 4115 4116 4117 4118
4119 4120 1060 1302 4121 4122 4123 4124 4125 4126 4127 344 148 4128 4129 4130 4131 1092 4132
4133 4134 4135 4136 4137 4138 4139 4140 4141 1310 273 4142 4143 4144 4145 4146 4147 4148 4149
4150 4151 1163 116 4152 4153 4154 4155 259 251 4156 4157 4158 4159 4160 4161 4162 407 4163
4164 375 60 4165 4166 4167 1043 4168 4169 4170 4171 36 4172 4173 4174 1105 385 4175 4176
4177 4178 4179 4180 4181 4182 4183 220 4184 1077 4185 4186 4187 1007 4188 4189 4190 106 4191
11 4192 4193 1280 4194 4195 4196 4197 4198 4199 4200 289 92 4201 4202 4203 4204 4205 4206
1311 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223 4224
4225 4226 1097 1123 4227 4228 4229 4230 62 4231 351 4232 4233 4234 4235 4236 4237 4238 4239
4240 1257 4241 4242 4243 4244 4245 4246 4247 4248 1056 4249 183 4250 1245 4251 4252 4253 4254
4255 4256 4257 338 4258 4259 4260 4261 117 341 4262 4263 4264 1079 4265 4266 4267 4268 4269
4270 4271 4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282 4283 4284 4285 4286 4287 4288
4289 4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303 4304 4305 4306 4307
4308 345 4309 4310 234 4311 4312 281 4313 4314 1212 1088 4315 4316 1323 4317 4318 4319 392
210 4320 4321 4322 4323 4324 393 4325 4326 4327 4328 4329 4330 1010 23 4331 4332 4333 4334
1068 4335 4336

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملا أساسيا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردى بعدما كان الأمر مقصوراً فى الماضى على المراسلات الرسمية أو السرية بطبيعة الحال.

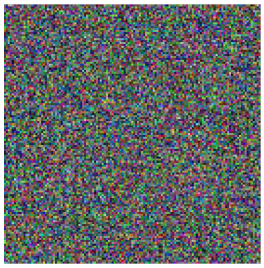
(a)

8ʊ8▒8}¢j^∂ز╔╔ש╔╔j8♂♂®±ĀʊĀ¦88¦(▒}÷╔♂}j¢∂àא`j0ıoز®שʊʊâ♠â&∑▒÷(8♠┤÷┤j∂
(&g8}¦`¦8j(ë÷Ж÷Жjv`♂8±┤¢Æ8^){&ıoıo88j&שÓÖ¦ʊ¢א3ز÷`!ë}ıo¦(ز∑┤¦÷÷3j8®Ā&#j8∂ë8â}╔÷8#8ʊʊʊıoâ:}jj8(&שıoıo8┤j╔8®ĀĀ3¢±
88ë(¦àjא∂}:8¢¢jj¦ç83÷8`ز&∂0&ز®&j÷8▒â┤}Ö&}¢}!j┬jâ3¦¦#`8┤jÖ&Ā8¢â¦±¢j3♠8}ʊ♠Ā3¢`¦ë3±▒▒▒ت&÷j¢ے 8!8┤
∂Æ3÷}jj}¦¢¦&¦8♠♠♠j3ت¦ز¦ıo¦¢8∂8jت÷ʊâ÷^â¢{ز:8÷¢ز∂8¢±¢ıoıoıo╔╔8!¢¦8¢â♂∑∂∑+_¦&زÆ ے {÷`&
8jj(&▒Æ╔╔╔â8┤א8Ā&א÷^±3∑8}{¢ت Ж®¢8¦&♠&¦ت▒0א8±ש∑∑Ā&(&}3¦â3¦®8&â&(÷àj¦(¦ے
ëâ_8ززjj&88&88ت&àj(¦±╔+!}÷ززت÷} Ж&&}ʊجë:8÷╔Ö8!¢ ש}8±8Öے ت╔8{8&8*!ت: { ʊز±&
ے®3ëâ3زشززjʊא0}∂¢j÷&gıoıo¢3┤♠}3♠j8÷âj88زg0¢¢∂ıo∂¦â?ت÷┤▒8÷&j♠+Æ8¢^±ت(`jıo^▒א ▒ز (àت Ā3: ıo&8±ש¢`3ژ88┤ ے ıo3Ж8ے
8:8}±}8زj`88#┤¦&j¦88: Ö3ز8¢¦}∑8&▒زj&8ЖК (¢&ز®
(j88ژ¦ت÷ف}ë÷&8÷&♂¢88ززjh&Æ&÷8¦3&¦88¢8!{¢)¢&8÷ز±±}^!âj«!â¦à8&&!8ز
♠┤&▒8{j8j±¦ש¦j8à÷:8®8ز±∑3j&3ששש8±┤}j}j÷÷+i#∂j¦(3¦ز3┤àĀЖ}3! ש±÷â83&â0¦â3`0&ت▒ے ز ë83شàא∂ıoא∂∂3&8┤Æے
Æ8^â♠^3}&â¦┤┤ز^Öj¦â888ے
â}÷&}&88÷ز¢ز8ز*:zj3¢}j¢زא8&jزıoj}8¢¢ز♠♠8â&┤+3¢8¢ت:âjژ8{+┤`âà÷#8¦`}ز÷âЖ&j:`jà÷83∑Æj÷÷j`â÷}÷*Öj30₪8÷8زà*8¢┤÷ز±÷
3÷¢8«8±┤&j8¦8â÷8╔8¢j88±8ز♠8∂ز3j&♠¦¢ë╔┤÷زâ388àzj¦¦0ت®®÷jâ┤}÷«ɜ3â8}₪8▒&8â3*¦₪₪¢â¦¦â÷¦88¦*8┤ʊ±¦à÷^¦שЖ88∑8ë¦jâĀĀĀ&8
*08#«8ЖÆ8j#¢ʊʊâ3÷!زÖ&┤{{*ÖÖÖשששש∂88ژتj┤÷¦àà: ╔▒▒ש8ز÷¦0Æ♠}j&}∂}jزʊא₪ЖК╔Жâ¦÷ت₪ı┤¦¢ששز}3¢8ت₪±jâ∑!÷38j}¦÷♠

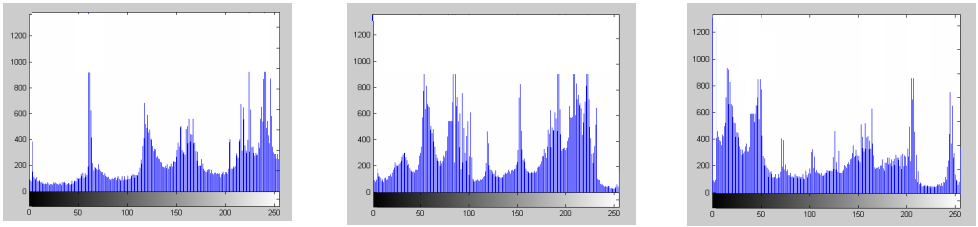
(b)

Figure III.19. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

(a)

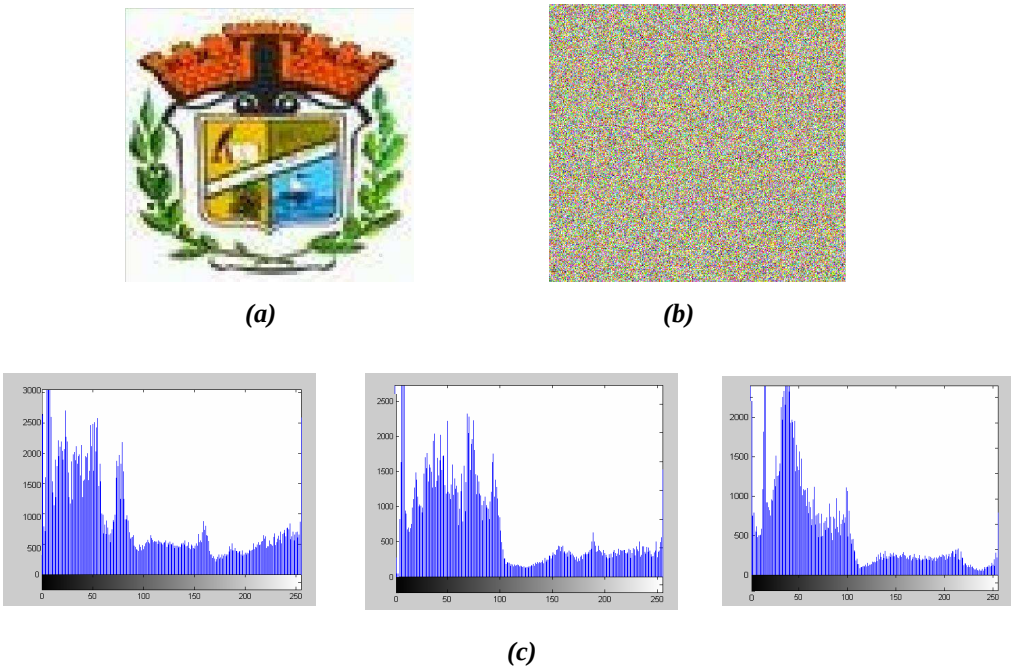


(b)



(c)

Figure III.20. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*



(a) ***(b)***

(c)

Figure III.21. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

III.4.2. Chiffrement à base d'occurrences

Nous avons vu dans la section précédente que le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, dans la présente section nous allons présenter un nouveau codage totalement différent de ceux vu dans les sections précédentes qui ont été bâti autour de la notion de positions de la représentation des éléments constituant la donnée à chiffrer.

Pour appliquer l'approche évolutionnaire à notre problème, nous présenterons les choix utilisés en terme de : représentation chromosomale des solutions du problème, méthode de création de la population initiale des solutions, fonction d'évaluation, opérateurs génétiques et procédure de sélection. Pour mieux approcher le problème, nous présentons une formalisation de ce dernier.

III.4.2.1. Formalisation du problème

Soit D une donnée à chiffrer (texte ou image) qui est une suite de n éléments e_i, et que l'on peut formaliser comme suit :

$$D = \{e_i\}, i \in [1,n] \qquad \text{(III.17)}$$

Dans le cas où D soit une donnée texte, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{C(e_i)\}, i \in [1,n] \qquad \text{(III.18)}$$

où : $C(e_i)$ représente le codage Unicode du caractère e_i.

Dans le cas où *D* soit une donnée image, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{p_i\}, i \in [1,n] \quad \text{(III.19)}$$

Cette représentation (structure) distinguant les différents pixels formant cette image, facilite la conversion en une autre structure en choisissant le RGB (Red Green Blue) comme espace de représentation d'images. Il suffit de remplacer les n pixels par leurs représentations correspondantes en RGB. Nous obtenons, ainsi, la formalisation suivante :

$$D = \{R_i G_i B_i\}, i \in [1,n] \quad \text{(III.20)}$$

Ce mode de représentation (structure) permet de distinguer les différents codages d'éléments de la donnée à chiffrer (les codages des caractères d'une donnée texte, ou les différentes composantes codant les pixels d'une donnée image), et par conséquent de déterminer le nombre d'occurrences de chacun d'eux. C'est ce mécanisme qui sera d'ailleurs exploité pour bâtir notre suivant algorithme de chiffrement proposé.

III.4.2.2. Description d'OEEA (*Occurrences based Evolutionary Encryption Algorithm*)

Comme l'opération de chiffrement consiste à perturber la donnée originale de telle manière à avoir le maximum de désordre dans la donnée chiffrée, nous avons choisi, cette fois-ci, de jouer sur les nombres d'occurrences des éléments (caractères d'un texte ou valeurs des composantes R, G et B codants les pixels d'image), pour avoir la plus grande différence entre les nombres d'occurrences des valeurs similaires d'éléments (nombres d'occurrences des caractères d'un texte original et leurs nombres d'occurrences dans le texte chiffré correspondant ; ou nombres d'occurrences des valeurs des composantes R, G et B codant les pixels d'une image originale et leur nombre d'occurrence dans l'image chiffrée). Ce choix est dû au fait que cette unique donnée (nombres d'occurrences) ne représente pas une information pertinente pour les cryptanalystes.

Dans ce qui suit, nous décrierons les différentes étapes de notre nouvel algorithme de chiffrement proposé.

a. Codage

Les structures (III.18) et (III.20) vues précédemment représentent les données de base à partir desquelles nous sommes arrivés à définir notre codage d'individus en calculant le nombre d'occurrences des valeurs possibles d'éléments dans une certaine donnée. Ainsi, le codage proposé sera le suivant :

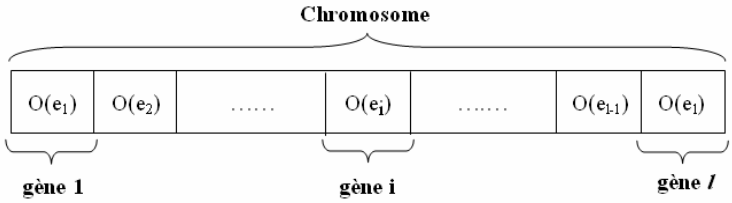


Figure III.22. *Codage des individus sous OEEA.*

Où : O (e_i) est le nombre d'occurrences de l'élément e_i dans la donnée,
l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Remarque :

1) $\sum_{i=1}^{l} O_i = n$ Où n ∈ N représente la taille de la donnée en termes d'éléments (caractères ou pixels).

2) Les éléments qui ne figurent pas dans la donnée auront une occurrence nulle.

Dans le cas de manipulation d'une donnée texte, l'opération de codage consiste à transformer le message en code particulier en calculant le nombre d'occurrence dans le message des 1393 caractères admissibles et l'attribuer à la case qui correspond à son rang dans une table, désignée par TCAR, regroupant les 1393 affichables en Unicode.

Le codage adopté dans ce cas, sera une instanciation du codage général présenté par la figure III.22. C'est celui illustré par le synoptique de la figure III.23.

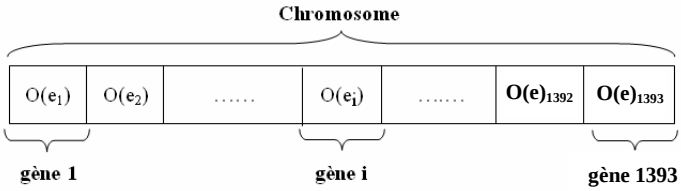


Figure III.23. *Codage des données texte sous OEEA.*

Où $i = \overline{1-1393}$ est le rang des caractères dans la table TCAR et $O(e_i)$ le nombre d'occurrences dans le massage du caractère ayant comme rang i.

Dans le cas de manipulation d'une donnée image et vu que les composantes R_i, G_i et B_i / $i = \overline{1,n}$ prennent leurs valeurs dans l'intervalle [0, 255], notre codage consiste à compter parmi les valeurs de chacune des composantes R, G et B, les nombres d'occurrences relatifs aux valeurs de l'intervalle [0, 255].

Il s'agit de compter à partir des valeurs des n éléments de R, les nombres d'occurrences des valeurs de l'intervalle [0, 255], et de même pour les valeurs des éléments de G et de B. La structure résultante dans ce cas, représente le codage adopté par notre algorithme développé. C'est celle résumée par la représentation chromosomale présentée à travers le synoptique de la figure III.24.

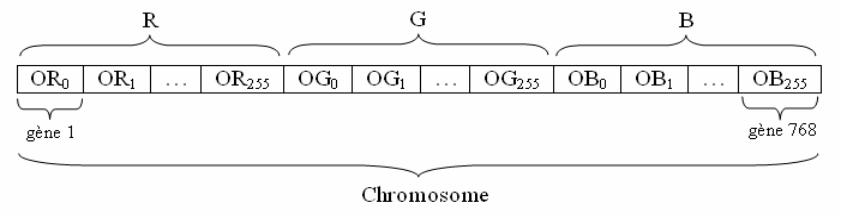


Figure III.24. *Codage des données images sous OEEA.*

Où : OR_i est le nombre d'occurrence des valeurs de la matrice composante R qui égale à *i*,
OG_i est le nombre d'occurrence des valeurs de la matrice composante G qui égale à *i*,
OB_i est le nombre d'occurrence des valeurs de la matrice composante B qui égale à *i*.

Si le codage binaire semble tout à fait adapté pour certains problèmes d'optimisation, cette représentation semble peu appropriée dans notre cas car elle est non intuitive. La représentation qui nous semble la plus naturelle est le codage entier, puisque toutes les opérations de l'algorithme vont manipuler des nombres d'occurrences qui sont des nombres entiers.
Dans ce qui suit, on désigne par individu la donnée à chiffrer ou toute autre donnée, et par chromosome tout individu ayant subi une telle opération de codage.

b. Création de la population initiale

Les m individus (I_1, I_2,…, I_m) formant notre population initiale sont obtenus par application de m perturbations aléatoires sur le chromosome initial I_0 représentant le codage de la donnée originale.

c. Reproduction

Les opérateurs génétiques servent à diversifier la population gérée par l'AE afin d'explorer le plus efficacement possible l'espace de recherche. Nous avons vu dans le chapitre 2 que les AEs disposent généralement de deux opérateurs : l'opérateur de croisement et l'opérateur de mutation.

- **Le croisement :**

Le croisement de solutions peut être toute opération permettant d'obtenir une nouvelle solution à partir de plusieurs individus existants.

Dans le but de compliquer la tâche des cryptanalystes tout en maintenant raisonnable le temps de calcul global de l'algorithme et en évitant toute convergence prématurée, nous avons utilisé un croisement à deux points. C'est le OX « Order Cross-over » proposé par Davis [Davi, 1985]. Les deux points de croisement sont choisis aléatoirement et l'opérateur est appliqué avec un taux qui varie entre 60% et 100% [Gren, 1986].

- **La mutation :**

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

Les individus résultants de ces deux opérations (croisement et mutation) seront rajoutés à la population des parents pour les acheminer vers l'étape suivante qui est celle d'évaluation.

d. Evaluation des individus

La fonction d'évaluation que nous avons définie pour associer des valeurs d'adaptation à chaque individu I_i d'une population Pop, est la suivante :

$$F(I_i) = \sum_{j=1}^{l} \left| O(e_j)_i - O(e_j)_0 \right| \qquad \text{(III.21)}$$

Avec : $I_i = [O(e_1), O(e_2),\ldots, O(e_l)]$, $Pop = \{I_1, I_2,\ldots, I_m\}$ où m est un paramètre de réglage et $O(e_j)_i$ représente le nombre d'occurrences du $j^{ème}$ élément dans le $i^{ème}$ individu.

Dans le cas de manipulation d'une donnée texte, F peut être instancié comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O(e_j)_i - O(e_j)_0 \right| \quad \text{(III.22)}$$

Ou tout simplement :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.23)}$$

Où O_{j_i} est le j^{ème} gène du i^{ème} individu.

Dans le cas de manipulation d'une donnée image, F sera instancié comme suit :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{j_i} - R_{j_0} \right| + \sum_{j=1}^{256} \left| G_{j_i} - G_{j_0} \right| + \sum_{j=1}^{256} \left| B_{j_i} - B_{j_0} \right| \quad \text{(III.24)}$$

Cette fonction est équivalente à celle donnée ci-dessous.

$$F(I_i) = \sum_{j=1}^{768} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.25)}$$

e. Sélection des individus les plus adaptés

La stratégie de sélection de l'algorithme développé va décider de la survie d'une donnée dans la population. Nous avons utilisé la méthode de la sélection par roulette.

f. Critère d'arrêt

Les étapes de reproduction, d'évaluation et de sélection sont répétées jusqu'à ce que l'optimum recherché soit trouvé. Ce dernier représente l'individu ayant la plus grande valeur de convergence durant tout le processus évolutionnaire. Donc, c'est la donnée la plus différente de la donnée originale.

Dans le cas d'une donnée texte, la condition d'arrêt assurant la convergence de notre algorithme évolutionnaire est exprimée à l'aide de la fonction F comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.26)}$$

Preuve :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} (O_{j_i} + O_{j_0}) \quad \text{(III.27)}$$

$$\leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.28)}$$

Et comme :

$$\forall I_i, I_k \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_k} \quad \text{(III.29)}$$

Et :

$$\forall I_i \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.30)}$$

Donc :

$$(III.27) \Rightarrow \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.31)}$$

$$\leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.32)}$$

Dans le cas d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right|$$

Et comme : $((0 \leq R_{ji} \leq n) \,\&\, (0 \leq R_{j0} \leq n)) \Rightarrow (0 \leq \left| R_{ji} - R_{j0} \right| \leq n)$ (III.33)

$((0 \leq V_{ji} \leq n) \,\&\, (0 \leq V_{j0} \leq n)) \Rightarrow (0 \leq \left| V_{ji} - V_{j0} \right| \leq n)$ (III.34)

$((0 \leq B_{ji} \leq n) \,\&\, (0 \leq B_{j0} \leq n)) \Rightarrow (0 \leq \left| B_{ji} - B_{j0} \right| \leq n)$ (III.35)

Alors, d'après (III.33), (III.34) et (III.35) nous aurons :

$$0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq (256 \times n) + (256 \times n) + (256 \times n) \quad \text{(III.36)}$$

$$\text{(III.36)} \Rightarrow 0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n \quad \text{(III.37)}$$

$$\Leftrightarrow 0 \leq F(I_i) \leq 768 \times n \quad \text{(III.38)}$$

Donc, les inéquations (III.26) et (III.38) représentent les conditions d'arrêt mettant fin à notre processus crypto-évolutionnaire dans le cas de manipulation de données texte ou images, respectivement. Nous constatons que $F(I_i)$ est une fonction bornée.

Après quelques tests d'exécution, nous avons constaté que le fait de prendre cette condition uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition d'arrêt ((III.26) ou (III.38)). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous avons arrivé à déterminer ce nombre :

MaxGen = 50

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes dans le cas de chiffrement de texte :

1) $F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i}$

2) MaxGen = 50

Toutefois, elle englobe les deux conditions suivantes dans le cas de chiffrement d'images :

1) $F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n$

2) MaxGen = 50

g. Déchiffrement

Dans cette phase représentant l'opération inverse du chiffrement, une clé de session est générée suivant le même mécanisme utilisé pour nos deux précédents algorithmes. Toutefois, elle est de taille fixe vu que toutes les données texte ont une représentation chromosomale fixe englobant 1393 gènes et toutes les données images ont une représentation chromosomale fixe regroupant 768 gènes.

h. Réglage des paramètres et résultats

Afin de construire un jugement objectif à propos de l'algorithme proposé et de pouvoir noter le maximum de remarques et de comprendre les détails, une large panoplie d'images tests de différentes dimensions a été utilisée. Nous présentons ci-dessous et à titre d'exemple les résultats obtenus pour les mêmes données test précédemment utilisées dans la présentation de nos deux algorithmes développés et précédemment présentés PosESecL1 et PosESecL2 : "Texte 1" de taille 1084 caractères, "Texte 2" de taille 1151 caractères, "Lena" de taille 131X131 pixels et "Logo" de taille 420X395 pixels.

- **Réglage de paramètres**

L'algorithme décrit ci-dessus possède différents paramètres, dont les valeurs influent fortement sur sa qualité. Cette section présente une étude empirique du réglage de ces paramètres afin d'obtenir de bonnes performances, ainsi que le comportement général de l'algorithme qui en résulte.

Les figures III.25 et III.26 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation.

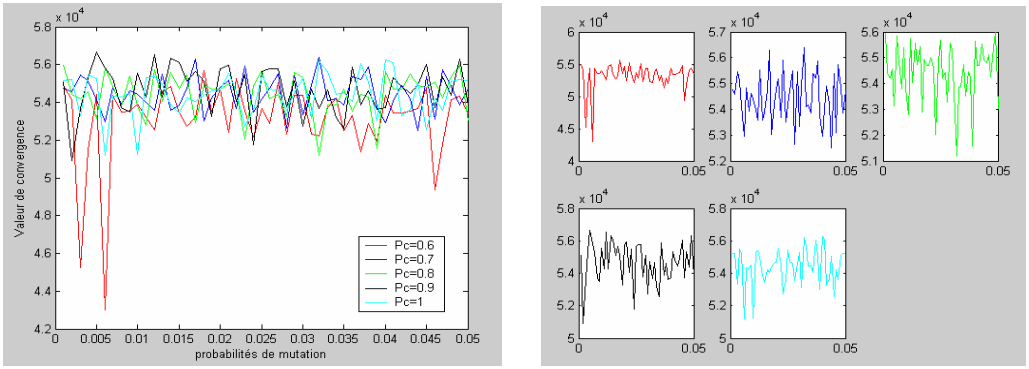


Figure III.25. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

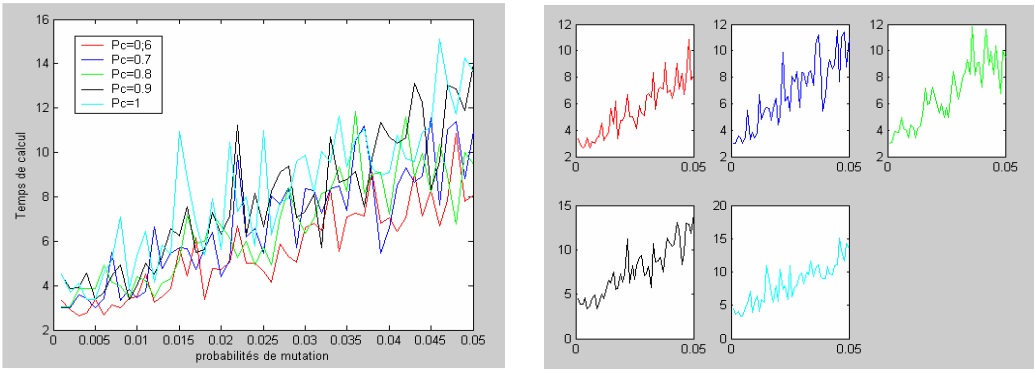


Figure III.26. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Pour choisir les valeurs des paramètres de la probabilité de croisement et celui de la probabilité de mutation, le niveau de confusion n'est pas la seule exigence à satisfaire ; le temps de calcul est aussi important. Ainsi, les valeurs qui nous semblent les plus adaptées sont : *Pc=0.9* et *Pm=0.005* assurant un haut niveau de confusion (VC=56674) en un temps de calcul très raisonnable (T=3.37s).

Après avoir fixé les paramètres de la probabilité de croisement et de mutation, la taille de la population est un autre paramètre à régler expérimentalement aussi. Les valeurs testées sont résumées à travers les figures III.27 et III.28 en indiquant pour chacune d'elles les résultats de chiffrement obtenus en terme de degré de confusion (représenté par la valeur de convergence) et de temps de réponse.

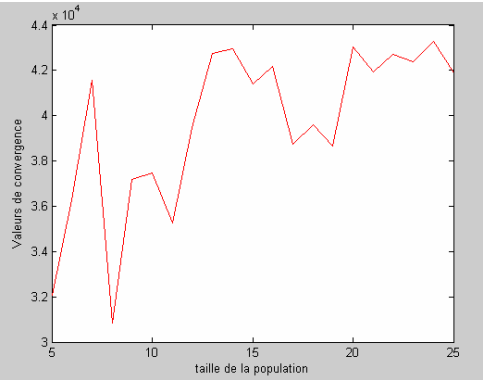


Figure III.27. *Evolution des valeurs de convergence en fonction de la taille de population.*

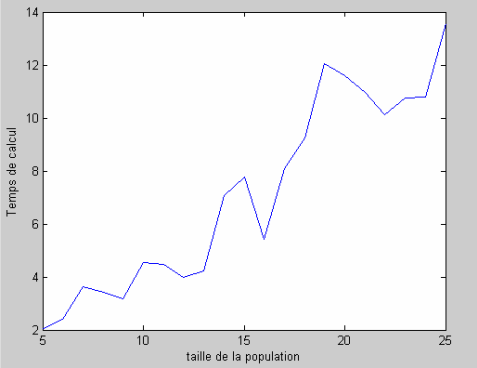


Figure III.28. *Evolution du temps de réponse en fonction de la taille de population.*

D'après la variation de l'influence de la taille de population sur la valeur de convergence et le temps de calcul résumé à travers les deux figures III.27 et III.28, nous optons pour régler le paramètre relatif à la taille de population en lui attribuant la valeur 13 (VC = 42739, T = 4.24s).

Le même principe a été utilisé pour fixer le paramètre relatif au nombre de générations. Donc, et suite à une série de tests sur un ensemble de données soumises au chiffrement et en utilisant la fonction fitness proposée, nous avons retenu certaines valeurs de paramètres qui permettent d'avoir un résultat exploitable en un temps de calcul très raisonnable (de l'ordre de 5 secondes). Ces valeurs sont données par le tableau III.5.

	Valeurs des paramètres de OEEA
Taille de la population	13
Nombre de générations	50
P_c	0.9
P_m	0.005

Tableau III.5. *Valeurs adoptées pour les paramètres d'OEEA.*

- **Résultats**

Après avoir décrit en détail notre méthode nous rapportons, dans cette section, les résultats de l'application de notre troisième algorithme proposé, OEEA, sur les données test citées précédemment (Texte 1, Texte 2, Lena et Logo) en adoptant le paramétrage fixé ci-dessus.

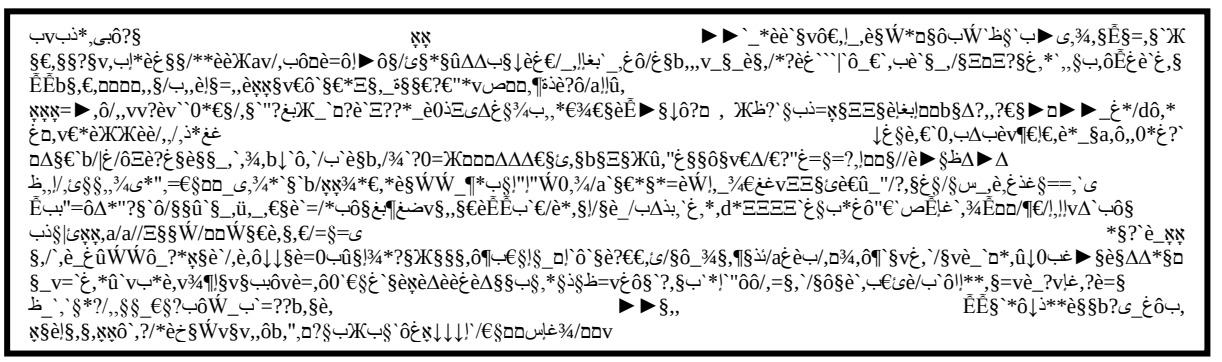
Nous constatons que la taille des clés est indépendante de la taille des données à chiffrer et elle est fixe pour des données de même nature (15323 bits pour les données texte et 7680 pour les données images).

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	15323	2173	6.54
	Texte 2	1151	15323	2284	6.73
Données images	Lena	131 X 131	7680	46034	3.37
	Logo	420 X 395	7680	431340	3.85

Tableau III.6. *Résultats obtenus par OEEA.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)

Figure III.29. ***(a)*** *Donnée originale Texte1,* ***(b)*** *Donnée chiffrée correspondante.*

Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

78 72 75 95 104 96 110 91 108 163 135 87 160 144 129 116 126 83 142
123 102 157 114 82 156 101 164 36 141 103 81 137 151 107 7 111 9 86
100 140 145 153 84 128 133 162 1053 1054 1055 1056 1057 136 120 1089 1090 1091 1092
1093 124 105 80 12 94 134 159 147 127 139 150 155 98 161 90 119 106 154
125 85 152 92 143 148 130 99 132 1281 1282 1283 1284 97 149 131 109 113 93
354 570 347 922 437 923 243 662 465 3 849 213 206 910 924 832 823 456 583
480 925 758 340 568 926 551 369 927 498 791 777 594 638 704 8 394 505 5
789 432 840 690 739 350 813 449 928 796 911 357 525 282 782 604 541 929 810
487 59 698 366 812 462 453 731 403 596 799 930 931 932 225 281 569 933 648
934 820 1386 1387 935 500 476 682 683 921 497 936 261 768 649 904 937 639 447
622 177 938 769 727 496 900 412 200 264 410 914 563 746 901 783 318 642 625
436 778 284 451 459 328 833 43 185 827 186 319 939 592 441 847 502 220 517
455 218 839 603 556 609 286 772 51 850 940 906 664 316 368 209 786 629 589
352 941 878 23 657 273 942 515 869 183 331 790 346 279 861 247 593 943 445
400 558 30 418 344 856 256 595 795 676 624 187 471 944 807 170 757 892 566
64 945 265 779 879 27 269 637 315 168 946 655 607 644 853 742 857 666 564
417 477 947 272 781 305 872 495 806 522 19 580 587 514 640 948 949 876 735
228 920 528 588 950 240 211 1231 1232 1233 915 458 33 287 312 951 545 863 860
571 667 490 952 953 954 888 916 913 299 391 905 671 378 442 669 575 591 726
811 865 202 398 358 399 955 956 314 454 172 341 234 621 546 214 643 737 230
957 223 788 891 259 577 736 958 416 181 263 755 486 50 289 618 959 292 846
864 174 960 519 433 395 434 20 902 961 907 962 1389 1390 1391 1392 1393 275 39
359 964 965 173 966 687 653 337 473 756 700 842 463 329 895 345 818 539 236
787 601 221 793 599 333 919 773 899 504 69 300 701 802 581 58 483 297 967
464 201 721 635 309 255 4 1 800 808 590 605 334 968 548 815 324 969 970
37 203 508 195 835 443 402 870 831 882 65 190 971 250 392 972 307 406 22
401 973 801 387 974 501 237 540 610 222 822 450 798 843 470 765 561 301 439
975 608 482 976 260 351 467 660 885 555 977 552 320 409 267 348 868 373 189
251 205 734 559 14 978 229 431 24 375 659 628 837 612 466 825 979 227 452
619 841 257 178 509 805 980 311 656 871 204 981 623 407 193 1060 1061 1062 1063
754 364 985 385 986 761 520 530 198 288 29 41 485 280 987 1230 1231 1232 1233
1234 290 764 988 989 469 1337 1338 1339 1340 1341 524 507 646 239 627 991 884 526
858 852 681 661 317 1197 1198 1199 1200 1201 826 992 322 699 353 1127 1128 1129 1130
475 993 47 994 887 1140 1141 1142 1143 877 672 70 430 384 684 192 995 527 996
303 997 674 383 828 1049 1050 1051 1052 342 2 326 60 678 491 866 283 26 404
792 1293 1294 1295 650 422 31 886 338 549 68 48 277 484 531 207 585 536 49
1000 705 258 600 651 521 1001 1002 523 438 363 460 15 293 513 381 747 444 550
753 310 1213 1214 1215 1216 1217 1003 898 335 248 245 1004 912 1005 386 274 529 634
630 1006 216 785 503 1257 1258 1259 584 356 620 1007 547 745 271 743 304 617 729
217 285 855 397 330 1008 1009 1010 557 516 875 750 771 355 512 728 296 472 492
1011 361 1012 880 582 809 166 427 446 803 420 494 1013 1371 1372 1373 1378 1379 1380
1381 167 576 784 1014 474 688 733 759 567 468 829 343 1271 1272 670 636 543 626
867 370 598 327 308 606 380 28 658 1015 774 295 844 665 61 560 188 268 732
821 1035 1036 1037 1038 1039 1040 874 371 360 673 663 196 1016 692 184 597 298 1017
614 415 16 1374 1375 723 1376 1377 55 752 715 244 685 215 1018 199 253 848 362
419 212 52 171 616 613 1019 917 425 390 534 1020 918 889 738 897 565 232 270
481 1021 252 488 535 641 396 838 893 797 578 896 766 1022 249 424 1023 1376 1377
1378 1379 388 834 1024 372 224 633 242 794 276 1025 854 631 1026 336 393 775 1356
1357 1358 1359 1360 691 532 675 197 235 499 194 572 730 1027 776 54 414 654 883
862 573 1028 506 349 909 679 586 56 780 873 493 332 428 231 602 377 632 542
824 423 816 53 749 1029 562 554 717 763 11 696 408 894 210 1030 179 702 645
751 804 680 1031 323 718 448 819 457 175 489 325 851 238 1032 306 814 741 374
233 611 461 1033 313 510 668 376 1034 57 748 724 44 1041 1042 1043 1044 1045 695
1046 1277 1278 1279 421 246 294 740 1058 1059 710 1064 1065 1066 1067 1068 719 1069 1070
1071 1072 1073 1074 1075 1076 1077 1078 1079 706 1080 77 73 76 74 79 117 146 1081
1082 1083 1084 63 1085 1086 1087 1088 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104
1105 1106 1107 1108 712 1109 1110 1111 1112 1113 25 169 881 413 382 1114 1115 708 1116
1117 1118 1119 1120 1121 1122 1123 714 1124 1125 1126 615 770 1131 720 1132 1133 1134 1135
1136 1137 1138 1139 18 1144 1145 1146 365 208 440 429 1147 1148 1149 1150 1151 1152 1153
1154 1155 35 1156 1157 1158 1159 1160 1161 903 537 1162 1163 1164 1165 1166 1167 1168 1169
1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 709 1180 1181 982 983 405 984 845 226
1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 66 1202 1203 1204
1205 713 1206 1207 1208 1209 1210 1211 1212 115 138 118 112 122 1218 1219 1220 1221 1222
1223 17 1224 21 1225 1226 1227 1228 1229 998 999 6 836 817 180 1235 1236 1237 1238
1239 1240 1241 1242 711 703 1243 1244 707 722 1245 1246 1247 1248 1047 1048 367 1249 1250
10 62 697 1251 1252 1253 1254 1255 1256 254 389 176 339 191 1260 716 1261 1262 1263
1264 1265 13 1266 1267 1268 1269 34 182 574 1270 1273 1274 693 1275 1276 1280 1285 1286
1287 725 1288 1289 1290 1291 1292 1296 1297 744 278 219 1298 1299 1300 1301 1302 1303 1304
1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 694 1320 32 1321
1322 158 88 89 121 40 1323 1324 1325 1326 1327 1328 1329 45 1330 1331 1332 1333 1334
1335 1336 1342 1343 67 1344 1345 1346 1347 1348 1349 1350 1351 46 1352 1353 42 1354 1355
544 426 291 647 762 262 963 411 1361 1362 1363 1364 1365 990 302 38 579 479 435

1366 1367 1368 1369 1370 538 859 686 241 908 760 533 165 830 511 266 553 1382 1383
1384 1385 518 677 890 1388 689 652 478

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسيًا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال

(a)

gé¥¥§&◄¥c:¥~x┼Ч‌x;>ش╪♫♫⇑¦;>}&◄◄¦¥"<;g§§q5⇑ ¥¥;}¬;§¦<<<î{xcx;<¥¥§xקקt;:g¥¥;ى";<xx¥:§¥§}§;¦;¥¥x;¥<
³x۞۞⇑۞rgt¦¥xrxg◄t¬g¦x4và¦x¥x>§_⇑¦ƒîx¦;<ƒv*§r¥ىx§{¥x¥xqx;⇑ ↔↔îsxî§⇑;¥§<{Ч<j³:&r'קק¥╪xr:
ى۞q¬xƒ>♫¦¥<:q¸↔~®¥xé§g§۞۞;ش;g&x؟♫§xq;
۞¬{î<¥l§;c§¥xî¥§§x{xî۞0>¦◄хњхњхîàњ¦{¥x_:q§Ч§&r¥x¥étsЧЧ{&>¥x;;¦x╪ى;¥x&4§¥:&³³<"îx§ЧЧЧé§xq{¬<ى&{□¥à&x¦¥⇑⇑&>;<
¥î§§¥§ît¥¥rr§;î&;;>g{x¥x¥¥q;<x§;+ ┼┼┼>g§§{¥x;qs;îrxgxƒ¥⇑&x§¥{_î<³¦cx¥ƒ{x>;x۞x5¥⇑xrg;§x&¥¬xx: ◄x§ƒ³{xj¥¥q{î¥;¥j◄5&{g;;
њњq¥¥x&é;g;/>:;l¶‌x³xg¥î;¦;îxx۞۞q<¥:&&♫;¦p¦:/v¥xxxקp¥&§jx;x¥q◄◄¥>;{↔۞î~_::{;x_§xњ&{ﺔ¬¥gƒ¥éî>¥¥§§tl۞;¦x;x,x¶‌¥¦l:g{
qsq¥¥x<¦¶‌Ч;ºx¦>¦x;&q¥:§; ◄xàtgƒ{¦§¥⇑cq¥ggî³¦xî♫x<t{§>;{¥t¬q>¬╪g<§q:>®&_§;<ىr↔xx>¥¦¥Ч‌g<¥àtt~¥éxà5¥ƒ;_<<¥
♫¬xrîx¥§¥¥g~¥¦j¥{x¥↔↔¥¥g{¥§~§¥§¥¥⇑¥;;¥gƒ!lx;:¬;{gњњњx§╪╪xx&¦g↔↔{¥q۞x¥¥xx¥r{t{<®ggî;□۞!gx_ﺔ{_:x{t¦¶‌Чt>◄◄◄
4g¥x<<¥¥q;à¥⇑qg³x¥;¥ƒs<_và¥r§♫xàx;³ §>xƒƒ◄◄x۞tq¥{>îxt⇑x¥¥éxxs;_xىs;{>
קà'xq{ƒ<<קקx¶‌xrgsrv<xњњ4¥;x<x³sg;§ج¦x>¦t♫§§&g{<{;;¥qקx³gњњxƒ¥;ىxéj¥s۞њsss¥¥¥¥קxq_x;¥>¦<§î¥qstc5;sj³³¥s§>;:_ﺔé~g¥¥<x:4<
۞۞{¥ts§<{s§xxrq¥<¥{;:¥¥>&¥;x¦g♫:s;:®¥gsx;¥<v¦s/¬}>ى¥¸x⇑⇑s~sx;w{◄x:ىº¦ig§jx_¶‌Ч&;g_؟>s{&◄۞¥t¦5;s®{{♫⇑¥ش§¥¥>!x
<۞¥x<x;0>¦<>xt;¥î♫{x¥⇑îjs<¦;↔x¦¥><;&xrקקקs¥¬g&<x⇑:x{î³³x:îЧЧ_'{g§◄r§¥;<x/¥<⇑⇑{;¥{{{:□~¥x۞

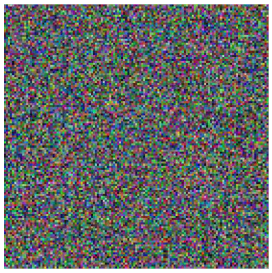
(b)

Figure III.30. **(*a*)** *Donnée originale Texte2,* **(*b*)** *Donnée chiffrée correspondante.*

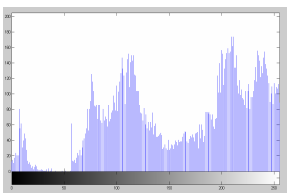
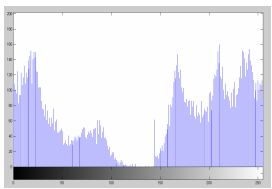
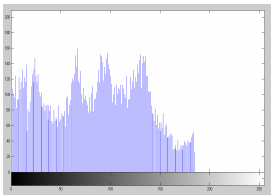
Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

59 1091 1 96 52 8 21 1200 47 54 71 16 43 14 1366 1369 25 64 26
1271 30 32 89 15 65 68 53 28 1303 1305 42 84 81 78 1090 34 24 62
20 58 74 56 39 1117 4 85 99 1324 18 98 1306 29 72 55 66 5 45
49 17 40 23 73 80 97 50 608 102 848 575 286 1220 1223 167 980 869 803
925 1201 134 310 1278 622 866 190 472 727 1118 535 818 1367 114 377 326 835 553
1368 166 488 100 1270 874 206 981 307 232 873 880 676 538 712 578 982 305 860
1325 899 933 842 332 865 315 713 239 445 619 203 939 983 785 192 1302 163 749
872 1327 297 605 513 722 270 940 269 814 198 943 201 570 195 422 746 897 802
402 904 945 250 794 560 691 1276 217 693 514 1304 852 935 804 666 518 1222 133
467 915 156 910 407 1221 170 257 738 542 692 912 984 508 443 245 815 671 372
63 61 67 985 1277 174 128 477 930 639 986 800 987 667 258 988 456 309 989
817 856 688 607 559 520 459 577 188 439 369 325 643 314 172 363 796 592 312
105 193 124 157 148 582 669 822 566 390 779 806 397 1187 1188 573 849 901 917
813 149 317 285 696 180 337 725 824 634 1057 223 990 829 554 953 466 732 242
733 905 647 308 168 101 487 958 1326 793 572 812 454 517 449 991 831 366 540
704 868 356 212 913 847 321 408 1059 280 623 909 484 271 137 662 220 433 324
881 1375 841 1333 896 837 624 992 891 1334 649 724 581 161 502 1275 545 685 924
843 1058 473 585 1374 828 833 275 993 382 450 176 107 1285 375 427 291 929 795
299 362 599 1283 863 178 736 515 531 937 331 292 994 742 888 1332 313 596 126
381 648 359 411 489 431 376 234 120 934 1284 175 714 522 955 534 247 614 598
652 1378 995 143 410 601 996 997 1273 453 883 158 208 719 298 430 689 446 462
357 104 840 928 184 413 303 657 364 399 870 256 396 636 424 159 391 1384 1385
115 1319 702 277 279 419 241 329 1000 734 698 627 469 871 521 690 602 262 140
342 687 1376 1001 464 887 350 116 268 1272 1274 895 720 465 267 197 1002 588 541
468 361 808 132 204 141 1377 287 444 547 438 150 761 300 221 425 921 1320 348
403 737 611 496 304 862 171 586 709 792 820 834 127 406 274 229 145 194 1003
886 546 392 922 288 673 731 448 205 276 561 263 1064 956 625 1004 186 227 492
1005 442 254 593 1006 1007 395 584 260 902 1065 530 968 590 334 389 1008 558 708
877 393 412 475 678 646 526 1203 1204 273 533 832 470 243 1009 579 884 474 451
786 850 908 723 1066 405 501 1010 1011 117 677 1012 498 658 1382 699 765 819 1013
214 1014 1015 233 423 135 1016 650 164 504 615 721 222 457 900 892 911 455 954
1017 642 1018 351 151 838 138 500 1381 505 1019 437 458 226 436 385 580 946 1020
179 131 400 653 173 1101 306 481 213 218 240 686 707 106 626 380 1021 511 358
333 146 1063 620 674 610 111 811 567 1022 185 507 603 878 338 706 1023 283 482
1100 266 379 323 799 926 432 1098 906 893 951 845 1024 388 668 568 718 855 764
110 701 949 224 112 864 756 551 125 920 680 1206 209 142 409 1025 207 612 844
670 633 587 367 645 302 130 591 1312 805 383 621 1026 169 679 103 349 1027 660
655 781 571 544 861 641 730 480 1028 322 1029 628 705 255 682 493 1030 1031 162

890 211 919 1099 478 927 1032 661 355 1205 768 932 1033 853 1067 236 1034 340 681
556 527 801 537 486 182 327 613 189 606 604 703 265 384 1035 617 231 1193 246
165 942 1036 728 640 821 875 200 597 659 261 717 441 415 854 1037 108 1192 569
823 374 281 894 539 923 807 452 202 154 1359 483 440 177 882 485 248 562 319
495 160 387 695 259 697 122 876 401 1190 651 576 414 879 1038 230 966 635 426
519 916 600 950 210 352 118 751 237 918 563 386 963 282 630 656 754 365 797
1182 1183 609 716 295 740 339 476 370 1039 330 378 675 1055 228 885 557 851 113
715 272 594 741 747 826 238 858 672 574 109 497 1191 447 152 353 121 663 494
760 532 499 129 898 404 1040 583 1184 524 565 638 344 294 416 253 644 629 119
938 284 429 1056 320 360 479 555 798 346 825 183 710 947 1041 936 1080 318 595
249 632 827 199 743 1180 336 637 683 435 421 191 729 684 944 810 460 512 509
503 857 1042 418 1094 739 654 1043 1044 525 529 839 1093 809 510 1181 664 771 244
394 311 434 859 1095 931 368 589 316 123 941 957 948 181 139 354 1045 1046 776
784 1047 1048 1049 1050 830 398 264 335 1051 1052 775 1053 1054 420 543 75 76 1060
979 1061 1062 1068 1069 1070 1071 1072 762 1073 1074 196 506 1075 1076 1077 969 1078 1079
11 9 1123 1124 1081 1082 293 373 1083 1084 973 965 1085 1086 296 564 1087 1088 1089
83 36 90 77 6 92 3 91 10 95 33 1092 371 219 1096 1097 726 251 1102
252 903 1103 1104 1105 1106 1107 147 889 552 1108 1109 774 1110 1111 1112 790 1113 773
1114 735 289 1115 1116 37 69 1119 1120 1121 1122 1125 788 975 1126 1127 461 780 1128
960 1129 1130 341 215 1131 1132 1133 971 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143
1144 1145 1146 976 1147 1148 1149 1150 769 1392 1393 1151 1152 1153 962 1154 1155 1156 225
694 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 216 914 153 1347
301 1171 1172 1173 1174 1175 752 1176 1177 1311 290 836 1178 1179 1185 1186 490 471 550
1189 1353 1194 766 1195 782 1196 1313 1197 1198 1199 665 428 1202 907 616 763 1207 977
1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 767 379 321 1224 758 1225 1226
1227 1228 1229 523 278 1230 1234 1235 770 1236 1237 1238 1239 748 1240 1241 1242 22 31
44 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1310 1254 1255 967 1256 1257 1258
1259 1260 753 964 1261 1262 816 1265 1266 618 516 1267 1268 1269 789 1279 1280 13 12
48 41 1281 1282 783 463 1286 1287 1288 1289 1290 959 1291 1292 1293 1294 70 7 1295
1296 1297 548 187 1298 1299 1300 1301 136 536 1307 1308 1309 970 744 1314 1315 757 777
1316 1317 1318 791 1321 1322 1323 27 35 1328 1329 144 345 952 1330 1331 961 1335 1336
1337 51 79 1338 1339 1340 711 235 1341 1342 1343 974 1344 745 1345 1263 1346 94 86
1348 1349 1350 867 155 1351 1352 528 750 1354 998 328 999 1355 1356 60 88 1357 82
46 38 87 57 1358 772 1360 1361 1362 343 549 491 417 1363 1364 1365 1370 1371 767
1372 1373 347 700 1379 1380 778 631 1383 759 846 1264 1386 1387 93 2 1388 1389 755
1390 978 787 972 1391

(a) ***(b)***

(c)

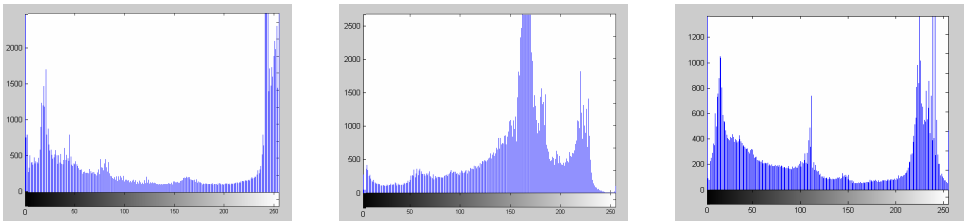
Figure III.31. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 0,9375 K octets) :

369 423 529 171 397 412 474 15 689 64 112 390 267 21 118 304 6 28 574
465 356 338 717 638 339 110 243 458 508 450 361 608 635 115 593 749 151 742
575 716 428 109 486 584 294 722 521 141 87 695 373 225 452 602 679 13 208
85 468 634 709 298 70 297 514 601 439 359 320 251 222 292 319 765 456 614
142 645 625 288 383 217 125 476 718 274 236 729 483 567 114 487 492 303 542

426	628	675	713	296	505	370	730	409	395	764	641	396	669	460	337	264	230	204
623	358	293	418	150	520	134	328	469	762	440	533	651	551	535	286	438	462	103
355	347	63	158	16	561	205	410	643	703	432	519	693	247	620	346	40	736	366
129	652	164	327	371	506	363	154	241	447	434	624	585	163	748	376	380	144	407
59	143	667	252	10	536	548	766	360	335	455	311	497	200	466	249	751	35	119
165	341	36	741	398	448	656	18	65	233	734	17	531	512	595	660	104	586	81
435	299	101	92	309	454	436	710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579	352	739	698	306	253	577	248
735	515	342	285	61	219	138	568	705	658	427	511	532	210	357	307	153	633	711
98	433	619	691	60	20	137	733	485	324	659	419	719	684	269	392	754	68	259
284	467	295	48	300	224	541	600	715	630	613	172	66	425	558	604	38	692	34
557	263	258	384	51	317	375	111	325	146	701	472	388	607	289	94	644	377	745
580	500	598	496	240	443	594	364	554	74	637	187	712	655	747	576	145	47	501
417	490	622	162	491	626	578	699	750	14	128	493	282	226	459	351	589	209	582
545	740	596	147	275	121	362	470	53	193	291	333	290	179	523	527	42	30	402
738	525	113	374	664	646	507	609	685	756	189	700	148	260	654	464	191	350	378
176	257	538	599	603	11	702	372	571	237	12	96	170	192	653	88	416	540	25
180	522	642	166	72	246	194	707	680	33	639	159	215	75	728	227	516	681	344
90	99	559	127	513	265	7	23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555	69	708	345	494	271	238	726
393	534	132	385	587	686	647	368	203	3	481	482	332	199	167	687	560	539	495
743	44	583	91	610	690	336	31	261	403	379	255	725	498	323	386	198	572	220
484	73	322	677	606	340	272	343	58	503	381	26	421	517	761	79	676	334	348
760	49	663	235	597	566	313	627	39	414	618	518	479	55	752	186	694	563	649
175	223	133	549	420	277	510	727	305	391	228	632	404	196	206	84	52	184	755
212	581	763	429	107	697	724	590	77	737	445	71	682	502	509	120	254	413	758
640	149	616	43	195	242	239	211	611	453	757	218	86	19	310	387	2	105	117
424	244	504	116	449	591	8	229	605	399	629	182	57	461	314	588	177	234	190
102	106	714	122	617	612	216	673	312	431	326	621	80	382	721	93	45	156	665
553	767	475	174	329	394	415	547	188	331	543	365	308	46	67	670	330	157	354
302	389	480	367	232	768	27	207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631	41	471	131	201	245	54	202
444	556	544	720	270	32	688	759	442	279	478	136	197	123	273	671	437	262	457
672	268	349	95	50	488	706	546	183	221	178	160	528	696	62	9	214	615	573
489	430	97	744	22	753	524	446	213	746	731	181	231	283	661	570	401	276	78
400	4	562	526	683	301	89	477											

(a) ***(b)***



(c)

Figure III.32. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session (Taille$_{Clé}$ = 0,9375 K octets) :

503	463	515	233	412	71	181	502	153	365	302	449	225	362	500	33	523	198	519
471	404	521	497	145	496	504	494	205	138	505	6	522	214	149	451	422	498	146
520	63	453	320	516	510	350	287	495	39	387	385	20	115	80	361	472	69	513
213	204	105	425	68	311	507	506	208	508	24	41	499	511	237	54	415	509	37
518	517	52	458	384	493	390	8	191	78	501	40	163	514	229	512	99	574	372
579	351	396	36	389	17	370	312	626	436	5	1	284	118	629	334	594	625	174
704	648	28	534	469	657	342	411	479	401	45	533	234	675	379	23	235	231	223
77	324	729	634	548	217	555	142	673	159	346	568	551	85	195	343	333	408	158
10	563	532	373	264	73	242	359	179	280	473	709	605	128	327	82	756	308	638
588	152	31	766	272	285	166	667	714	51	613	157	439	265	561	271	314	277	491
286	32	125	2	332	194	547	552	241	156	443	661	143	455	431	120	215	484	578
434	526	647	344	257	371	744	168	199	584	260	707	486	763	369	244	524	298	540
689	405	764	196	595	538	345	478	636	464	95	216	29	567	46	735	57	331	397
752	426	377	288	133	114	134	62	424	668	720	399	200	336	696	459	141	553	173
702	169	148	171	460	622	210	35	409	201	570	209	444	14	624	558	299	483	66
366	184	224	600	100	671	738	293	677	549	192	193	291	554	230	76	481	639	67
124	545	620	226	93	529	303	136	164	470	603	150	687	419	304	683	414	581	74
565	127	245	144	577	337	760	475	557	705	301	49	418	137	44	110	7	79	306
207	13	635	380	276	758	420	278	248	238	292	630	457	47	437	583	445	375	723
354	492	684	255	679	81	263	119	221	61	682	751	669	652	104	681	659	290	92
121	454	188	172	564	355	754	708	489	251	394	220	12	60	97	126	566	604	403
601	15	395	429	330	715	645	546	490	759	108	154	423	34	441	87	741	575	643
666	101	19	693	686	718	111	631	326	542	617	674	706	84	178	543	381	50	167
733	402	716	654	462	614	239	300	254	725	262	619	232	428	305	593	335	161	427
83	250	341	180	374	632	465	406	719	183	378	623	382	243	147	740	765	296	649
658	736	582	252	165	712	745	89	413	569	160	106	633	329	177	608	340	461	589
537	261	456	38	103	607	640	627	98	4	30	597	416	281	596	599	536	253	734
26	742	477	642	266	438	135	386	573	295	339	721	711	190	487	753	591	539	162
737	610	726	313	140	197	70	16	525	621	637	86	749	182	572	11	474	139	612
730	347	598	43	556	541	319	48	761	187	664	748	275	646	672	258	151	219	710
259	256	236	348	450	609	109	325	186	571	580	678	703	435	430	694	606	376	274
240	328	206	393	175	466	615	315	587	75	383	602	616	527	25	273	762	448	697
113	452	132	155	724	364	739	294	485	728	202	750	476	650	544	116	480	688	42
338	530	352	123	743	247	641	676	368	211	176	107	467	698	322	55	560	56	731
685	270	170	767	227	655	690	279	310	746	644	203	732	680	663	662	417	64	528
651	433	691	96	59	246	585	757	112	318	283	618	628	22	297	665	592	535	94
65	58	222	102	586	699	3	695	358	228	717	212	392	576	531	289	130	88	356
388	316	122	768	660	421	446	185	611	268	670	410	398	468	249	321	755	72	9
18	692	700	117	323	722	447	307	353	360	131	90	407	129	363	562	432	488	590
27	367	400	559	349	317	391	440	656	653	727	218	442	482	282	713	53	747	91
269	309	189	267	21	701	550	357											

III.5. Discussion et évaluation des résultats

Un bon crypto-système doit satisfaire les six points clés de Kerckhoffs [Kerc, 1883] et ceux de Shannon [Shan, 1949]. Plus explicitement, la qualité d'un crypto-système se mesure principalement par sa résistibilité face aux différents types d'attaques. La technique de cryptage proposée à travers nos trois algorithmes proposés PosESecL1, PosESecL2 et OEEA et qui consiste en une perturbation évolutionnaire de la donnée originale, assure une bonne sécurisation contre les attaques les plus destructives. Les critères utilisés pour évaluer leurs sécurités sont les suivants : la vitesse de l'algorithme, l'attaque statistique, l'attaque différentielle, l'attaque exhaustive et l'analyse de l'espace des clefs.

III.5.1 Vitesse des algorithmes

En plus du paramétrage, la vitesse d'un algorithme évolutionnaire dépend étroitement du codage utilisé. Un mauvais codage affecte non seulement la qualité de l'algorithme en termes d'optimalité de solutions mais aussi en termes de temps de convergence. Dans le cas de nos deux premiers algorithmes proposés, PosESecL1 et PosESecL2, le codage dépend de la taille de la donnée à chiffrer ce qui affectera la vitesse des algorithmes puisque le temps de traitement des données de grandes tailles sera beaucoup plus grand que celui de traitement des données de petites tailles, chose due à la taille des chromosomes manipulés durant les générations de l'évolution. Ceci se voit clairement à travers les résultats obtenus par ces deux algorithmes où le temps de convergence de PosESecL1 est meilleurs que celui de PosESecL2

du fait que la taille des chromosomes codant une donnée pour le PosESecL2 est trois fois plus grande que celle des chromosomes codant la même donnée pour le PosESecL1. Toutefois, le codage proposé à travers OEEA unifie la taille des chromosomes codant des données de même nature (textes ou images) et de différentes tailles. Ainsi, le temps de traitement est indépendant de la taille des données traitées. En plus et d'après les résultats expérimentaux présentés précédemment, nous constatons que le temps de convergence de ce dernier algorithme est très raisonnable (de l'ordre de 4 secondes).

Comparons, maintenant, la vitesse de nos trois algorithmes développés avec celle des principaux crypto-systèmes (AES du coté des crypto-systèmes symétrique, RSA du coté des crypto-systèmes asymétrique et SEC qui est un algorithme de chiffrement exploitant les algorithmes génétiques). Le tableau 7 et la figure 33 présentent un résumé des temps de chiffrement et de déchiffrement de chacun des algorithmes cités.

	PosESecL1	**PosESecL2**	**OEEA**	**AES**	**RSA**	**SEC**
Temps de chiffrement (s)	24.5	46.15	3.37	0.5	42.8	1.8
Temps de déchiffrement (s)	0.26	0.75	0.12	0.08	42.5	0.06

Tableau III.7. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

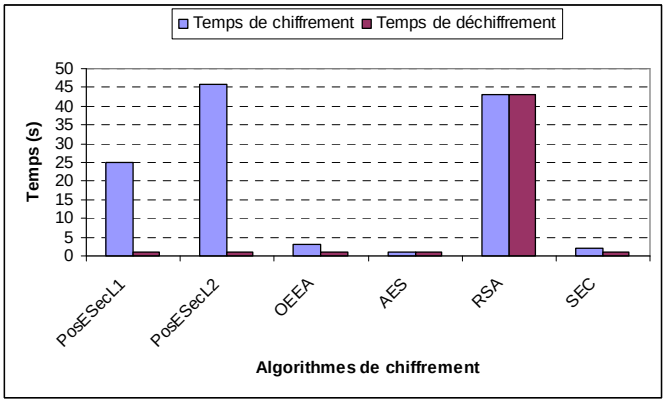


Figure III.33. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

D'après le tableau et la figure, ci-dessus, présentant le temps de chiffrement et celui de déchiffrement de nos algorithmes en comparaison des algorithmes SEC [Omar, 2007], RSA [Mene, 1996] et AES [www4], nous remarquons que :

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement beaucoup plus petit que celui de RSA. De son coté PosESecL1 possède aussi un temps de chiffrement/déchiffrement meilleur que celui de RSA. Toutefois, PosESecL2 a un temps de chiffrement de l'ordre de celui de RSA et un temps de déchiffrement meilleur que celui de RSA.
- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin de celui d'AES. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes PosESecL1 et PosESecL2 sont plus grands que celui d'AES.

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin que celui de SEC. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes sont plus grands que celui de SEC.

De même et d'après les résultats expérimentaux obtenus et présentés précédemment, nous constatons que le temps de calcul de OEEA est indépendant de la taille de la donnée à chiffrer du fait de l'unicité de taille de codage de toutes les données de même nature soumises au chiffrement. Toutefois, ce n'est pas le cas pour PosESecL1 et PosESecL2 où le temps de calcul des deux algorithmes est strictement dépendant de la taille de la donnée à chiffrer. Ainsi, OEEA est l'algorithme ayant un temps moyen de calcul le plus adapté.

III.5.2 Niveau de confusion

La confusion est l'une des notions essentielles énoncées par *Shannon* sur lesquelles se base un algorithme de cryptage. Elle sert à cacher la relation entre la donnée en clair (originale) et la donnée chiffrée correspondante. Donc, elle vise à rendre la donnée aussi peu lisible que possible.

Dans notre cas, les deux modes de chiffrement développés, à savoir le chiffrement à base de positions et le chiffrement à base d'occurrences, sont de niveaux de confusion différents.

Pour le premier, les deux algorithmes proposés dans ce mode sont aussi de niveaux de confusion différents. En effet, le niveau de confusion de PosESecL2 est meilleur que celui de PosESecL1 du fait que PosESecL2 exploite les positions des éléments codant les différents composants de la donnée, considéré chacun comme unité séparée (gène) du codage final de la donnée (chromosome). Ainsi, la donnée chiffrée aura de grande chance de contenir de nouveaux éléments ou composants ne figurant pas dans la donnée originale. Cette chose complique considérablement, voire même, pénalise toute cryptanalyse possible comme nous allons le démontrer dans les sections qui suivent. Toutefois, PosESecL1 est de niveau de confusion strictement dépendant de la taille de la donnée à chiffrer puisqu'il exploite les positions des éléments de la donnée. Donc dans le cas de données de petites ou de moyennes tailles, une attaque exhaustive finira de trouver la bonne combinaison initiale de positions et, ainsi de casser l'algorithme.

Pour le deuxième mode de chiffrement développé, il offre un très bon niveau de confusion grâce à l'utilisation d'un codage bâti autour de la notion de nombres d'occurrences. Ce mode ne présente ni information utile pouvant être exploitée par un cryptanalyse ni possibilité de reproduction de la donnée en se basant uniquement sur cette donnée surtout que le codage adopté assure que les composants de la donnée (caractères ou pixels) changent dans la donnée chiffrée par rapport à la donnée originale. En plus, le niveau de confusion de OEEA, l'algorithme développé dans ce mode, est meilleur que celui de PosESecL2 et ainsi de celui de PosESecL1 du fait que l'espace des nouveaux caractères qui peuvent apparaître dans une donnée chiffrée par rapport à la donnée originale correspondante est beaucoup plus grand dans OEEA que dans PosESecL2.

III.5.3 Attaque statistique

Ce type d'attaque considère le crypto-système comme une boite noire, il analyse statistiquement les entrées et les sorties de ce système. Pour ce faire, nous avons utilisé les mesures suivantes dans le but d'évaluer ou de quantifier la différence entre l'image originale et l'image cryptée correspondante :

Le facteur *NPCR* (*Number of Pixels Change Rate*) donné par l'expression (III.39), l'erreur absolue moyenne (*MAE : Mean Absolute Error*) donnée par l'expression (III.41) et l'erreur

quadratique moyenne (*MSE :Mean Square Error*) donnée par l'expression (III.42). Le tableau 8 résume les valeurs des différentes mesures obtenues après les tests qui ont été effectués sur l'image Lena et l'image chiffrée correspondante obtenue dans le cas de nos différents algorithmes proposés.

	NPCR	**MAE**	**MSE**
PosESecL1	0.6314	0.91	0.58
PosESecL2	0.9073	0.97	0.61
OEEA	0.9955	1.03	0.75

Tableau III.8. *Niveaux de confusion.*

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \tag{III.39}$$

$$D(i,j) = \begin{cases} 0 Si\ \mathrm{Im}_O(i,j) = \mathrm{Im}_C(i,j) \\ 1 Si\ \mathrm{Im}_O(i,j) \neq \mathrm{Im}_C(i,j) \end{cases} \tag{III.40}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \tag{III.41}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \tag{III.42}$$

D'après les résultats d'application des mesures quantifiant la différence entre l'image originale et ses versions chiffrées calculées par nos algorithmes, il est clair que la majorité des pixels formant l'image originale ont changé soit de positions (dans le cas d'application de PosESecL1) ou de contenu (dans le cas d'application de PosESecL2 ou de OEEA) pour, ainsi, calculer l'image chiffrée. En effet cela se voit surtout à travers les valeurs de NPCR qui calcul le niveau de différence entre les deux images où les résultats obtenus sont proches de la valeur optimale qui vaut un (01). Toutefois, ces valeurs ne sont pas les mêmes pour les trois algorithmes, chose qui est tout à fait normale vu que les algorithmes sont de niveau de confusion différent (voir section précédente). Et comme nos algorithmes présentent l'avantage d'être non déterministes, donc chaque nouvelle application d'un même algorithme donnera lieu à des résultats différents ce qui entraîne que l'application des mesures NPCR, MAE et MSE donnera lieu, de son tour, à de nouvelles valeurs. Ainsi, aucune propriété statistique ne peut être déduite d'une donnée chiffrée et l'attaque statistique ne possédera aucune chance de casser nos algorithmes.

III.5.4 Attaque différentielle

Cette attaque essaye de tirer des conclusions sur le fonctionnement d'un algorithme de chiffrement en comparant des versions chiffrées de plusieurs données originales et dans un meilleur cas des versions chiffrées de plusieurs blocs formant une même donnée. Ainsi, dans le cas des algorithmes de chiffrement par blocs (DES [Stin, 1996], [Biha, 1991], [Mats, 1994], 3DES [Stin, 1996], [Gant, 2001], AES [Lepr, 2000], etc), quand la donnée (surtout pour le cas d'images) contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement, ce qui fait que la donnée cryptée contiendra des zones texturées ; mais vu que nos algorithmes traitent la donnée en une seule passe, c'est-à-dire ils opèrent sur la donnée totale et non pas des blocs de la donnée ; alors la cryptanalyse

différentielle sera mise à l'écart. De ce fait aussi, nos algorithmes présenteront une certaine robustesse au bruit par opposition aux algorithmes de chiffrement par blocs, et qui se traduit par le fait qu'une erreur sur un bit chiffré ne va pas propager des erreurs importantes dans tout le bloc courant et par la suite dans toute la donnée lors de la recombinaison des blocs.

Même si le cryptanalyste pense à essayer de tirer des observations à partir des résultats de cryptage de plusieurs données pour essayer d'imiter le fonctionnement des algorithmes, l'approche évolutionnaire pseudo-aléatoire complique considérablement la tâche de ce type d'attaque. Une sensibilité à la donnée originale est aussi un point avantageux des algorithmes proposés vu que la donnée cryptée correspondante à une certaine donnée originale change d'une instanciation du problème à une autre. Autrement dit, nos algorithmes développés sont des algorithmes non déterministes à l'inverse du RSA, par exemple, qui a nécessité au préalable de randomiser les données soumises au chiffrement pour éviter les attaques exploitant les modèles connus des données en clairs [Ster, 2004].

III.5.5 Attaque exhaustive

L'attaque exhaustive ou encore dite attaque à force brute est une attaque qui demeure fatale pour les crypto-systèmes utilisant des clés de petites tailles. À ce stade, nos trois algorithmes proposés présentent des niveaux de sécurité différents. La robustesse de PosESecL1 est strictement dépendante de la taille de la donnée originale. PosESecL2 présente un niveau de sécurité plus élevé que celui de PosESecL1 du fait que sa clé soit de taille quatre fois plus élevée que celle de PosESecL1 dans le cas de manipulation de données textes et de trois fois plus grande dans le cas de manipulation des données images. Ainsi, et vu qu'à l'heure actuelle la taille de la clé assurant une résistibilité face à une attaque exhaustive est de 512 bits, alors le cryptage par PosESecL1 de toute donnée possédant une taille inférieure à 512/le nombre de bits nécessaires pour coder la taille de la donnée (nombre de caractères ou de pixels), sera exposée au risque d'une attaque réussite par force brute. C'est le cas aussi pour les données cryptées par PosESecL2 et qui sont de taille inférieure à : 4*(512/le nombre de bits nécessaires pour coder la taille de la donnée texte) ou de 3*(512/le nombre de bits nécessaires pour coder la taille de la donnée image). Toutefois, OEEA ne peut en aucun cas être cassé par une telle attaque vue que la taille de la clé utilisée est largement sécuritaire. Ainsi, OEEA est l'algorithme le plus sûr parmi nos trois algorithmes développés.

Considérons le texte à chiffrer « **there is only one God and Mohamed his prophet** », nous reportons dans le tableau suivant (tableau 9) les tailles de clés de chiffrement de nos trois algorithmes développés en comparaison de DES, 3DES, AES et SEC, ainsi que le nombre d'opérations nécessaires pour générer toutes les combinaisons de bits formant les clés pour chaque algorithme. Ceci donnera une idée sur la complexité de l'attaque exhaustive appliquée contre chacun des algorithmes où, il se voit clairement que l'attaque exhaustive menée contre nos algorithmes est la plus complexe.

	Taille de clé (bits)	**Nombre d'opérations**
DES	56	2^{56}
3DES	128	2^{128}
AES	256	2^{256}
SEC	240	2^{240}
PosESecL1	352	2^{352}
PosESecL2	1408	2^{1408}
OEEA	1393	2^{1393}

Tableau III.9. *Complexité de l'attaque exhaustive.*

III.5.6 Analyse de l'espace des clés

Ici, nous testons la sensibilité du processus cryptographique proposé aux clés utilisées. Dans notre cas, la clé générée est une clé calculée à partir de la donnée originale et de la donnée chiffrée correspondante, donc elle change d'une instanciation du problème à une autre. Ainsi, le cryptage successif d'une même donnée donne lieu à un ensemble de données chiffrées différentes ce qui entraînera, à chaque fois, la génération d'une clé de session différente pour le chiffrement d'une même donnée originale. Cela dotera notre méthode de cryptage d'une très grande sensibilité aux clés puisque toute clé interceptée d'une manière illégale ne servira que pour le déchiffrement d'une seule version chiffrée d'une même donnée donc elle ne sera plus utile par la suite.

Le problème rencontré dans le cas des crypto-systèmes symétriques, et par conséquent dans le cas de nos algorithmes développés qui sont des algorithmes symétriques, est la communication de la clé secrète au destinataire en vue de l'utiliser dans l'extraction de l'information originale (le déchiffrement). Pour ce faire, certains concepteurs ont proposé de combiner les fonctionnalités de la cryptographie symétrique et asymétrique. C'est le cas du PGP par exemple. Il crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ; puis, il crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire. Ainsi, le même principe peut être adopté pour sécuriser le transfert de nos clés en leurs chiffrant par un crypto-système asymétrique. Dans ce cas un schéma hybride de transmission sécurisée peut être envisagé comme le montre la figure suivante :

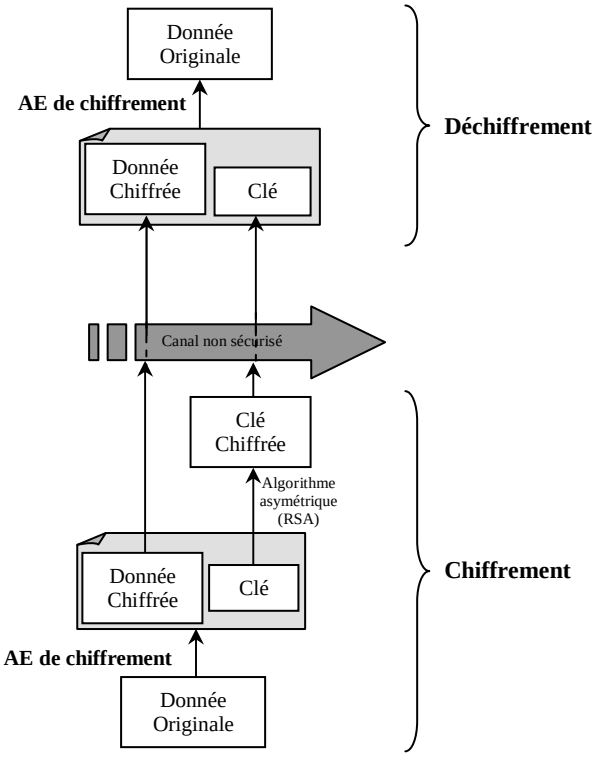


Figure III.34. *Schéma hybride de transmission sécurisée.*

Or, pour nos algorithmes PosESecL1 et PosESecL2, il est clair que la taille de la clé dépend strictement de la taille de la donnée. Le problème qui se pose lors du chiffrement d'une donnée de grande taille, c'est que la taille des clés générées elle aussi devient grande ce qui consommera un grand temps de chiffrement de clés du fait que les crypto-systèmes publiques sont déjà lents. Le problème s'aggrave de plus en plus lorsque la taille de la clé

devient égale ou dépasse la taille du codage de la donnée en termes de bits. Dans ce cas, l'utilisation de ces deux algorithmes devient inutile et il sera plus intéressant d'utiliser un crypto-système publique pour chiffrer la donnée au lieu de la clé générée malgré que la sécurité de tels crypto-systèmes soit grandement affectée par les choix paramétriques (p et q dans le cas de RSA [Zimm, 2005]). Pour remédier à cette anomalie, nous proposons de coder (ou bien comprimer) la clé générée par un système de codage sans perte, tel que le codage de Huffman par exemple [Huff, 1952], en vue de réduire sa taille avant de la chiffrée par un crypto-système asymétrique, puis d'envoyer au destinataire le package englobant la donnée chiffrée et la clé codée chiffrée. Lors de la réception, on procède inversement. On déchiffre la clé codée en utilisant la clé publique du destinataire, puis on la décode pour obtenir la clé proprement générée par PosESecL1 ou PosESecL2 qui va permettre de reproduire la donnée originale. Ainsi, le schéma du processus de sécurisation englobant cette dernière fonctionnalité sera le suivant :

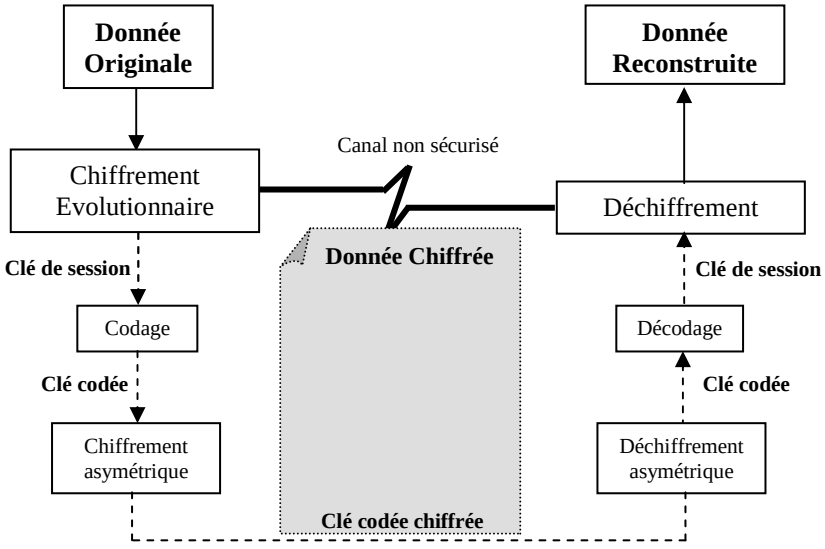


Figure III.35. *Schéma général du processus de chiffrement et de transmission sécurisée de la clé générée par chiffrement publique.*

Une deuxième solution de transmission sécurisée de la clé de session générée peut être envisagée dans le cas de manipulation de données images. Il s'agit de la technique de tatouage. Après avoir codé la clé générée par notre algorithme de la même manière proposée dans le cas de la première solution de transmission sécurisée de la clé (en utilisant un codage de Huffman par exemple), nous proposons, ici, de marquer l'image chiffrée par la clé de session codée. Ainsi, le schéma du processus de sécurisation englobant cette deuxième solution de transmission de la clé peut être résumée par le schéma de la figure III.36.

Cette deuxième solution de transmission est conditionnée par la taille de la clé codée pour satisfaire le compromis Robustesse-Capacité-Invisibilité lors du tatouage [Katz, 2000], [Barn, 2004], [Koba, 1990].

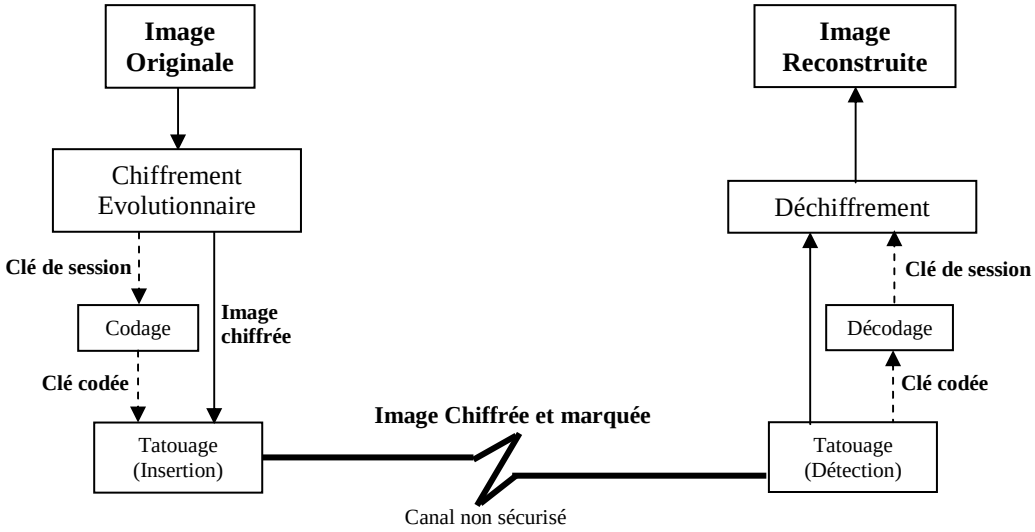


Figure III.36. *Schéma général du processus de chiffrement d'images et de transmission sécurisée de la clé générée par tatouage.*

III.6. Conclusion

À travers le présent travail, nous avons démontré que les algorithmes évolutionnaires, inspirés fondamentalement de la sélection naturelle des espèces, peuvent être également appliqués au cryptage de données. En effet, avec ces algorithmes, nous avons ramené le problème de chiffrement en un problème d'optimisation.

L'application de ces algorithmes requiert le réglage d'un certain nombre de paramètres pour la phase de chiffrement. La taille de la population initiale, le taux d'exploration et le nombre de générations affectent le temps de convergence des algorithmes, quant aux probabilités *Pc* , *Pm*, elles possèdent, théoriquement, une influence directe sur la qualité du résultat. Or, dans notre problème, les opérateurs génétiques (croisement et mutation) n'agissent pas sur l'information traitée (caractères du texte ou pixels de l'image) mais sur leurs coordonnées ; ils sont utilisés comme un outil pour explorer le reste de la population. De ce fait, les paramètres P_c , P_m affectent également la convergence des algorithmes.

Dans la pratique, les paramètres des algorithmes évolutionnaires sont réglés approximativement par tâtonnement [Gold, 1989]. Suite à une série de tests sur diverses données et en utilisant les fonctions fitness proposées, nous avons retenu certaines valeurs qui permettent d'avoir un résultat exploitable (donnée chiffrée) en un temps de calcul variant d'un algorithme à l'autre.

Les résultats obtenus montrent que les schémas proposés présentent des aptitudes dans la confusion et dans la sensibilité à la donnée originale qui les rend loin des attaques différentielles. De même, et pour pénaliser ou plus ou moins compliquer la tâche de l'attaque exhaustive, une deuxième alternative (PosESecL2) a été proposée en augmentant de trois ou de quatre fois la taille de la clé générée par la première alternative (PosESecL1). Une troisième alternative de cryptage robuste et sûre a été également définie. Il s'agit d'OEEA opérant suivant un mode de chiffrement (chiffrement à base d'occurrences) inexploitable par aucune attaque avancée.

Et pour résumer l'efficacité des deux modes de cryptage exploités, le cryptage à base de positions et le cryptage à base d'occurrences, le plus correcte est de dire que le deuxième

mode est plus important que le premier. En effet, ce mode est bâti autour de la notion d'occurrences, chose qui ne peut en aucun cas guider ou être exploitée par un cryptanalyste puisqu'elle ne présente aucune information utile pour aucune attaque possible. Ce qui n'est pas le cas pour le premier mode de cryptage à base de positions où une recherche exhaustive finira, un jour, par retrouver la bonne combinaison de positions d'éléments d'une donnée et ainsi de remettre en cause l'algorithme de cryptage utilisé.

Ainsi c'est les qualités du mode de chiffrement à base d'occurrences qui nous a motivées à le tester à travers d'autres métaheuristiques qu'elles soient à population ou à une seule solution. D'ailleurs le prochain chapitre présente une métaheuristique de colonie de fourmis traitant le problème de cryptage en opérant suivant ce mode de cryptage proposé.

Finalement, il sera utile de rappeler que l'objectif en cryptage est d'assurer la sécurité des données pendant leur durée de validité, donc, toute attaque tardive n'aura aucune importance.

CHAPITRE IV

CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

IV.1. Introduction

L'auto-organisation est un phénomène décrit dans plusieurs disciplines, notamment en biologie dans la branche qu'est l'éthologie (étude des comportements des espèces animales dans leurs environnements). Une définition a été proposée par Deneubourg [Dene, 1977] **:** *«l'auto-organisation est un processus dans lequel, un modèle de niveau global émerge uniquement, d'un grand nombre d'interactions entre les composants de bas niveau du système. De plus, les règles spécifiant les interactions entre ces composants sont suivies en utilisant uniquement des informations locales, sans références au modèle global ».*

Autrement dit, l'auto-organisation explique l'émergence d'un comportement collectif macroscopique par des interactions simples au niveau microscopique. L'auto-organisation n'exclut pas la complexité au niveau individuel. Elle suppose simplement qu'à un certain niveau, les individus se comportent comme des entités simples [Monm, 2000].

Depuis longtemps déjà, des comportements auto-organisés ont été découverts dans la nature. Par exemple, dans une colonie de fourmis, on peut observer différentes castes spécialisées dans un certain nombre de tâches : élevage de couvain, recherche de nourriture, construction de nid, etc. Aussi, le nid est construit sans que les insectes soient dirigés, ils répondent à un certain nombre de stimuli provenant de leur environnement.

D'importantes recherches ont eu pour intérêt principal l'étude de ces comportements intelligents afin de savoir comment ces populations interagissaient, accomplissaient des tâches et évoluaient. Ces recherches ont abouti à des modélisations inspirées du principe qu'un groupe d'entités plutôt simples et obéissant à des règles locales de coordination, sont capables d'engendrer des comportements globaux beaucoup plus complexes. Ces modélisations caractérisent ce que l'on dénomme « l'intelligence collective » [Bona, 1994], [Monm, 2000].

Dans ce chapitre, nous nous intéressons à l'algorithme de colonies de fourmis. Nous l'avons étudié, adapté et appliqué au problème de cryptage de données texte ou images. Le vocabulaire employé par cet algorithme est directement calqué sur celui de l'éthologie, nous parlerons donc, de fourmis, de connaissance collective, de phéromone, d'évaporation, etc.

Dans la première section de ce chapitre, nous présentons notre approche utilisée pour la conception du système de chiffrement dont l'objectif principal est l'obtention d'un système de

chiffrement symétrique non déterministe dont la clé secrète est générée pendant l'application de l'algorithme ; donc elle n'est pas connue à l'avance ce qui rend difficile le travail du cryptanalyste. Pour ce faire et étant donné que dans le précédent chapitre le problème de cryptage a été ramené en un problème d'optimisation, il a fallu, tout d'abord, coder le problème d'une manière adéquate permettant, ainsi, d'effectuer les opérations propres aux fourmis tel que : le dépôt, l'incrémentation et l'évaporation de phéromone, etc. Ainsi, le codage adopté est celui validé par la première métaheuristique d'AEs testée qui est le codage à base d'occurrences. Ce dernier permet la concrétisation des déplacements de fourmis par exploitation de la notion de permutation.

Ensuite, nous avons bâti notre algorithme en définissant soigneusement : la fonction d'évaluation cherchant à maximiser la différence entre l'information initiale et celle codée, le mécanisme de sélection adopté pour le déplacement des fourmis, le dépôt et l'incrémentation de phéromone, l'ensemble de la connaissance collective, etc.

IV.2. Motivation

Les algorithmes de colonies de fourmis forment une classe des métaheuristiques récemment proposée pour les problèmes d'optimisation difficile. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. Une colonie de fourmis communiquent indirectement via des modifications dynamiques de leur environnement (les pistes de phéromone) et construisent ainsi une solution à un problème, en s'appuyant sur leur expérience collective. C'est d'ailleurs cette dernière notion que nous cherchons à exploiter à travers l'utilisation de la métaheuristique de colonies de fourmis pour la résolution du problème de cryptage.

Ainsi, les objectifs que nous nous sommes fixés consistent à comprendre et isoler les mécanismes intéressants de cette métaheuristique, utiliser l'approche de chiffrement à base d'occurrences validée dans le précédent chapitre, suivant cette optique et appliquer l'algorithme ainsi conçu au problème de chiffrement de données texte ou images.

D'une manière générale, notre objectif est de développer un algorithme de chiffrement à base de colonies de fourmis par du constat simple que les colonies de fourmis résolvent des problèmes complexes, bien que l'intelligence d'une fourmi soit limitée ce qui est équivalent au fait de dire que, l'intelligence du système entier est plus grande que celle de la simple somme de ses parties.

IV.3. Algorithme proposé

Afin de bien comprendre notre algorithme de chiffrement proposé, AntCrypt, une description complète des différentes étapes de l'algorithme profondément inspirées de la métaheuristique *ACO (Ant Colony Optimisation)*, sera présentée. Des retouches ont été apportées sur l'algorithme de base pour l'adapter au problème étudié qui est celui de cryptage. Ainsi, le schéma de l'algorithme adopté est le suivant :

Algorithme IV.1 *AntCrypt*

Entrées
Codage de la donnée originale
Ensemble de paramètres (*m* : taille de la population, ρ taux d'évaporation)
DEBUT
1) Initialiser la connaissance collective // connaissance collective $\leftarrow \Phi$;
2) Créer la population initiale comportant *m* solutions initiales (*m* : nombre de fourmis) ;
Répéter
3) Construire les solutions;
4) Enrichir la connaissance collective ;
5) Evaluation des solutions de la connaissance collective ;
6) Sélection ;
7) Mise à jour de phéromone ;
Jusqu'à Satisfaction du critère d'arrêt
Retourner la solution trouvée ;
FIN

IV.3.1. Codage adopté

Pour prouver encore une fois l'efficacité de l'approche proposée et précédemment présentée exploitant un chiffrement bâti autour de la notion de nombres d'occurrences d'éléments de la donnée à chiffrer, le codage adopté par AntCrypt sera le même que celui d'OEEA.

La figure IV.1 rappelle le codage de données à utiliser.

$O(e_1)$	$O(e_2)$	.....	$O(e_i)$	.....	$O(e_{l-1})$	$O(e_l)$

Figure IV.1. *Codage d'une solution sous AntCrypt.*

Où : $O(e_i)$ est le nombre d'occurrences de l'élément e_i dans la donnée, l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Ainsi, notre algorithme AntCrypt cherche à changer itérativement la répartition des nombres d'occurrences $O(e_i)$ sur les différents éléments d'une certaine donnée avec génération aléatoire de positions dans la solution afin de créer le maximum de désordre dans leurs positions. Cela donne, aussi, l'avantage de produire des éléments (caractères/pixels) inexistants dans la donnée initiale.

IV.3.2. Création de la solution initiale

Initialement la connaissance collective (le savoir partagé) est vide. Dans cette étape on crée la population initiale comportant *m* solutions (*m* : le nombre de fourmis).

Nous désignons par *DonneeInitiale* le codage de la donnée originale. Donc, il est représenté par un vecteur dont les éléments contiennent les nombres d'occurrences des 1393 caractères (respectivement des 256 valeurs possibles de chacune des composantes R, V et B codant les pixels) dans le texte (respectivement dans l'image). Ces éléments sont notés :

$$O(e_1), O(e_2), ..., O(e_n) / n = \begin{cases} 1393 : Texte \\ 768 : \text{Im}age \end{cases}$$

Nous générons ensuite *m* fourmis susceptibles de créer, chacune, une nouvelle solution. Pour ce faire, chacune des fourmis est positionnée aléatoirement sur un élément de *DonneeInitiale*. Nous avons choisi de noter les différentes positions courantes par *PosCourante*. Ensuite, chaque fourmi va se déplacer un certain nombre de fois vers d'autres éléments (d'autres nœuds) notés chacun *PosSuivante* par application de la roulette aléatoire, et elle permute lors de chaque déplacement le nombre d'occurrence de l'élément « *DonneeInitiale[PosCourante]* » avec celui de l'élément « *DonneeInitiale[PosSuivante]* ». Une fois une fourmi s'arrête de se déplacer, nous obtenons une solution qui sera ajoutée à l'ensemble de connaissance collective. Alors à la fin de cette étape, la connaissance collective sera augmentée de *m* nouvelles solutions (chaque solution est obtenue par une fourmi).

IV.3.3. Construction de solutions

Après l'étape d'initialisation et dans le but de construire de nouvelles solutions sur une itération donnée de l'algorithme, les fourmis vont choisir à partir de la connaissance collective cette fois-ci leurs points de départ pour construire de nouvelles solutions. Ces points représentent les solutions ayant les plus grandes quantités de phéromone, c.-à-d. celles qui perturbent le plus la donnée initiale par rapport à une version cryptée représentée par la solution choisie. Donc le mécanisme permettant la construction d'une solution est le suivant :

1) Choisir les *m* meilleures solutions de la connaissance collective ;
Pour i = 1 : *m* ***faire***
    2) Choisir une solution *Sol* parmi les *m* solutions choisies en (1) ;
    Pour j = 1 : NbrMaxDép ***faire***
        3) Positionner aléatoirement la fourmi sur un élément *Sol[PosCourante]*;
        4) Déplacer la fourmi suivant la méthode de roulette aléatoire vers une autre position *PosSuivante* ;
        5) Permuter les deux éléments *Sol[PosCourante]* et *Sol[PosSuivante]* ;
    Fin pour
    6) Ajouter *Sol*, la solution qui vient d'être construite, à la connaissance collective.
Fin pour

IV.3.4. Evaluation

La fonction d'évaluation (ou d'adaptation) quantifie la qualité des solutions calculées. Celle que nous avons définie pour associer des valeurs d'adaptation à nos solutions calculées est la suivante :

$$F(Sol_j) = \sum_{i=1}^{n} \left| O_j(e_i) - O_0(e_i) \right| \quad \text{(IV.1)}$$

Tels que :

Sol_j : Solution j,

n : Nombre des éléments du vecteur *DonneeInitiale*,

$O_j(e_i)$: Nombre d'occurrences de l'élément (caractère / pixel) i dans la solution j,

$O_0(e_i)$: Nombre de répétitions de l'élément (caractère / pixel) i dans la donnée initiale.

IV.3.5. Sélection

Le rôle de la sélection est de distinguer les solutions sur la base de leur qualité, en particulier, pour permettre aux meilleures solutions d'une génération donnée d'être copiées dans la génération suivante.

Dans notre cas, lors du passage d'une génération g à la génération qui suit (g+1), nous favorisons les m meilleures solutions parmi celles de la connaissance collective globale y compris celles construites durant la génération g.

IV.3.6. Manipulation de phéromone

IV.3.6.1. Incrémentation de phéromone

Suivant la fonction F donnée en (IV.1), nous devons choisir la quantité de phéromone déposée pour chaque solution. Le mode de dépôt sélectionné peut fortement modifier le mode de convergence de l'algorithme. On pourrait choisir de déposer la même quantité à chaque solution. Cependant, on peut aussi décider de donner une puissance plus forte aux phéromones déposées sur les solutions améliorant l'altitude. C'est cette seconde solution qui a été appliquée dans notre algorithme.

Pour chaque solution *Sol*, nous calculons leur efficacité selon la formule (IV.1). Si cette solution existe déjà dans la connaissance collective on incrémente leur quantité de phéromone comme suit sans la réinsérer :

$$Phéromone(Sol_i) = Phéromone(Sol_i) + \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.2)}$$

Tel que EffMax est l'efficacité de l'une des meilleures solutions calculées de manière déterministe en permutant dans un ordre choisi le $i^{ème}$ élément avec le meilleur des l-i éléments restants (en maximisant la différence entre les deux éléments en question).

Sinon, si la solution n'existe pas, elle sera ajoutée à la connaissance collective en lui attribuant une quantité de phéromone selon la formule suivante :

$$Phéromone(Sol_i) = \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.3)}$$

IV.3.6.2. Évaporation de phéromone

Comme dans la nature, les phéromones sont des substances chimiques qui s'évaporent au fil du temps. Donc, il sera nécessaire que l'algorithme imitant ce coté de la nature représente ce phénomène afin d'éviter aux fourmis de converger vers un maximum local.

$$Phéromone(Sol_i) = Phéromone(Sol_i) \times (1 - \rho) \qquad \text{(IV.4)}$$

Où ρ est le taux d'évaporation.

L'évaporation de phéromone de toutes les solutions après une période de temps permettra de redonner de l'intérêt à d'autres solutions qui peuvent nous amener à la meilleure solution. Le taux d'évaporation ρ doit être bien choisi, puisque s'il est très faible, les phéromones s'accumulent et atteignent la borne max et nous risquons alors de s'enfermer dans un maximum local, et s'il est trop grand, les phéromones atteignent rapidement la borne min et beaucoup de solutions n'auront pas la chance d'être choisies par les fourmis.

Dans notre application, ce taux doit aussi dépendre des valeurs des éléments du codage. Si la différence entre deux éléments est grande, c.à.d. la fonction F va prendre de grandes valeurs correspondantes à de grandes quantités de phéromones, donc le taux d'évaporation peut prendre une valeur importante. De même, il dépend du nombre de fourmis travaillant sur une même génération : beaucoup de fourmis dans une génération augmentent la possibilité d'avoir deux solutions identiques, c.-à-d. la phéromone sera incrémentée plusieurs fois de suite, alors le taux d'évaporation sera important.

Ainsi et après un certain nombre de générations fixé expérimentalement, nous avons varié un pourcentage d'évaporation de 0% jusqu'à 0.5% en s'inspirant, au début, de l'application de l'algorithme ACO pour la résolution du problème de TSP [Barg, 2005]. Ce choix a été, ensuite, validé par expérimentation.

IV.3.7. Critère d'arrêt

Après un certain nombre de générations fixé expérimentalement, l'algorithme converge vers la meilleure solution trouvée. Reste de générer ou de reconstituer la donnée cryptée à partir du codage de la solution proposée. Ceci correspond à l'opération inverse de l'opération de codage. Il s'agit d'une opération de décodage qui procède comme suit : à partir du codage de la solution proposée qui est un vecteur de nombres d'occurrences des éléments de la donnée chiffrée, pour chaque élément *i* du vecteur *Sol*, on génère *Sol[i]* positions aléatoires du caractère/pixel correspondant à l'élément *i* dans la donnée cryptée.

IV.3.8. Déchiffrement

Nous arrivons maintenant au processus inverse qui permet de rendre la donnée à nouveau intelligible sans aucune perte d'information ; il s'agit du processus de déchiffrement.

Notre algorithme proposé est un algorithme symétrique, donc la clé générée doit être maintenue secrète. Cette clé s'obtient au fur et à mesure du calcul de l'information chiffrée au fil des générations. Sa valeur finale correspond aux permutations des positions des nombres d'occurrences des éléments de la donnée chiffrée pour obtenir celles des nombres d'occurrences des éléments de la donnée en clair. Le processus de déchiffrement consiste donc à replacer les éléments dans leurs positions initiales suite à l'introduction de la bonne clé.

IV.3.9. Réglage des paramètres et résultats

Cette section est consacrée d'une part, à l'optimisation ou la fixation du jeu paramétrique et d'autre part, à l'application de l'algorithme proposé *AntCrypt* sur les mêmes données de test utilisées pour l'évaluation des AEs proposés et présentés dans le précédent chapitre. Ils s'agissent des données faisant l'objet de la figure III.1. Dans un premier temps, nous allons exposer notre stratégie pour établir et fixer le choix paramétrique de notre algorithme. Puis nous allons présenter les résultats d'application d'*AntCrypt* sur les données modèles (textes et images), à savoir la donnée chiffrée, la clé générée, la quantité de phéromone, l'efficacité, le temps de chiffrement et de déchiffrement.

IV.3.9.1. Réglage des paramètres

Les figures IV.2 et IV.3 présentent les résultats des différents tests effectués en termes d'efficacité et de temps de chiffrement. Ces valeurs représentent la moyenne de dix (10) exécutions successives appliquées sur l'image de test Lena pour différentes valeurs de nombre de générations et de fourmis.

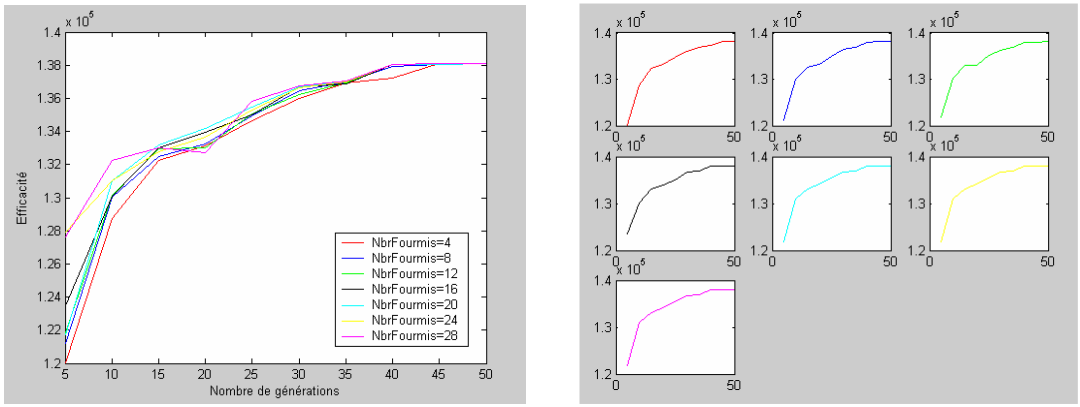


Figure IV.2. *Influence du nombre de générations et du nombre de fourmis sur l'efficacité.*

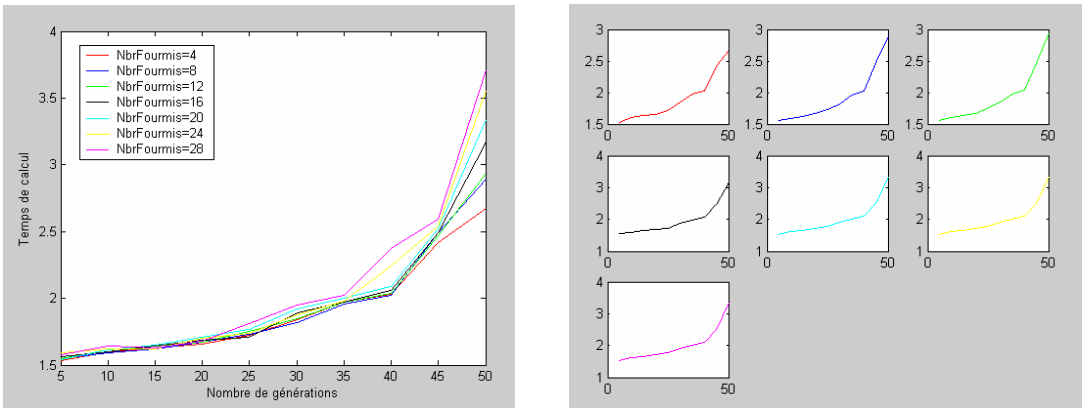


Figure IV.3. *Influence du nombre de générations et du nombre de fourmis sur le temps de calcul.*

D'après les résultats résumés à travers les figures IV.2 et IV.3, nous constatons que la majorité des résultats des tests ont des valeurs d'efficacité assez proches. La meilleure valeur d'efficacité est obtenue au bout de 50 générations à 20 fourmis (Efficacité = 138077) mais avec un temps de chiffrement assez important par rapport à l'ensemble de test (3,17 s). Cependant, le meilleur temps de chiffrement est obtenu pour le test qui a été déroulé sur 5 générations et 4 fourmis (1,53 s) mais malheureusement avec la plus faible valeur d'efficacité (Efficacité = 120009).

Les tests déroulés sur 45 générations avec 4, 20, 24 ou 28 fourmis, et 50 générations avec 4, 8, 12, 20, 24 ou 28 fourmis ont donné de très bonnes valeurs d'efficacité (138073, 138070, 138073 et 138074, 138075, 138074, 138076, 138073, 138072, 138075 respectivement) mais après de longs temps de calcul (2.42 s, 2.52 s, 2.54 s, 2.59 s et 2.67 s, 2.94 s, 3.34 s, 3.56 s et 3.71 s).

D'autres tests déroulés sur 5 et 10 générations ont montré, cette fois-ci, de bons temps de calcul (compris entre 1.53 s et 1.64 s) mais de mauvaises valeurs d'efficacité (comprises entre 120009 et 132263).

Donc, notre objectif sera de réaliser un compromis entre la valeur d'efficacité et le temps de calcul. En effet, le test déroulé sur 40 générations avec 12 fourmis semble le plus satisfaisant. Il permet d'atteindre une bonne valeur d'efficacité (Efficacité = 138063) en un temps de calcul très raisonnable (2.04 s). Ca sera le paramétrage à adopter pour *AntCrypt*.

IV.3.9.2. Résultats

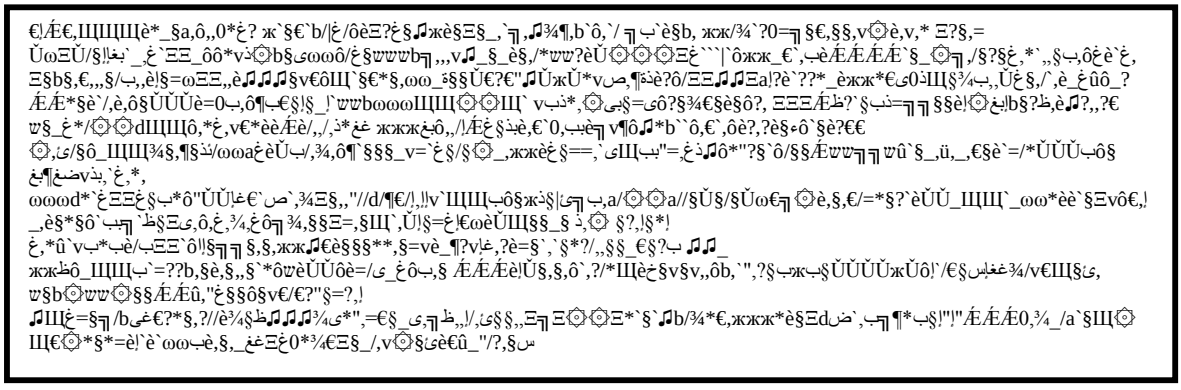
Ci-dessous, nous présentons les résultats d'application d'*AntCrypt* sur les mêmes données modèles utilisées pour tester nos trois algorithmes présentés dans le précédent chapitre : Texte1, Texte2, image Lena et image Logo suivant le paramétrage fixé dans la section précédente (résumé à travers le tableau IV.1). Toutefois, et pour démontrer l'aspect non déterministe de notre approche de chiffrement par exploitation de métaheuristiques, dont l'ACO fera l'objet ici, nous présentons, cette fois-ci, pour chaque donnée originale deux versions chiffrées en deux instances différentes accompagnées des clés de chiffrement générées dans chaque cas.

	Valeurs des paramètres d'*AntCrypt*
Nombre de fourmis	12
Nombre de générations	40

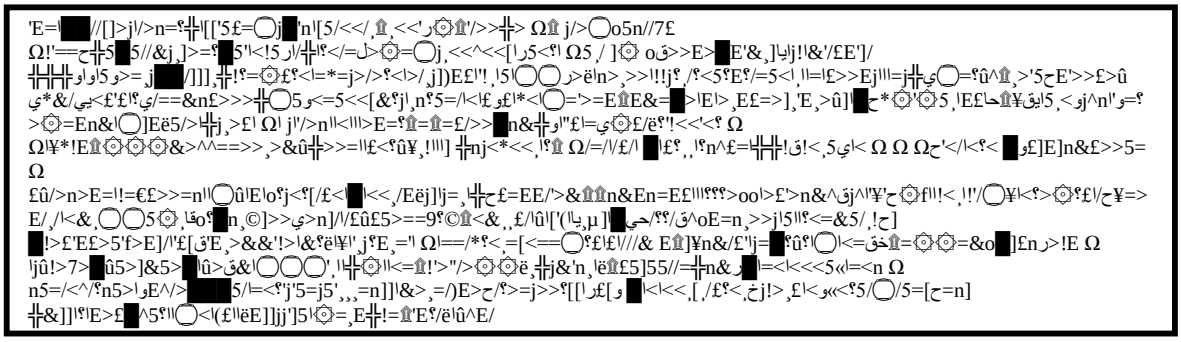
Tableau IV.1. *Valeurs adoptées pour les paramètres de AntCrypt.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure IV.4. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets) :

124	398	407	770	921	266	896	19	794	90	918	245
843	914	614	547	809	904	560	80	375	908	879	886
901	816	830	543	573	900	156	196	98	254	906	314
230	443	917	497	903	418	288	907	289	680	147	861
820	642	902	259	916	840	30	898	905	102	822	635
317	161	920	465	292	458	546	91	911	246	910	833
111	912	155	193	640	919	923	643	357	681	121	725
915	892	581	899	922	216	909	913	789	549	179	578
28	552	3	650	1070	379	850	996	934	933	1104	285
459	997	1124	1242	1026	897	1171	1075	8	521	64	281
1017	955	985	872	262	485	774	16	85	709	279	97
628	120	374	87	286	1112	277	862	518	47	255	1097
1203	45	163	60	599	12	461	5	848	1240	92	1032

331	174	888	83	1073	20	676	342	494	280	422	487
607	1020	958	61	1154	601	95	1100	112	386	1238	1196
1185	1119	393	669	1	756	1172	1012	1102	274	394	363
237	755	267	6	219	389	893	187	1225	437	175	683
834	659	1215	1085	1138	819	447	688	436	764	1158	117
206	1024	675	695	264	319	345	76	127	1183	484	1006
1191	540	1205	210	384	689	655	991	71	928	1130	416
1129	324	761	1134	1089	1029	449	1200	704	231	1013	466
474	1166	813	302	506	1050	490	1204	1092	382	194	744
1066	682	164	72	630	972	15	1213	738	309	1199	531
1149	1084	1202	971	535	395	1031	731	988	737	41	1188
1142	584	336	839	113	818	1245	306	723	657	620	403
252	858	994	423	1036	563	1217	1137	135	108	1133	1088
1160	810	829	946	1058	1068	952	944	751	673	1110	79
1232	835	733	557	1193	982	1179	298	1052	1101	198	1144
225	232	949	400	229	981	545	846	954	432	778	411
634	1161	1042	457	18	250	68	1123	866	462	1243	1051
1120	768	960	990	1189	1182	491	827	185	1145	1148	714
1115	428	574	995	1067	653	1091	978	27	415	132	940
145	409	665	1234	956	622	727	1227	786	402	1009	859
1236	1224	870	1039	387	239	1028	565	1043	594	176	856
787	424	323	588	1033	715	945	1195	359	1162	1165	793
791	930	684	299	716	1076	605	101	1241	149	826	412
1181	480	364	1187	1004	14	1226	579	438	1222	542	613
125	664	1176	197	202	1208	593	520	183	1207	248	604
434	1139	1127	1014	211	891	169	747	968	55	1190	339
1178	625	758	698	330	1038	446	425	611	668	646	1170
1174	144	166	283	1244	857	1231	1131	366	48	31	34
1107	52	814	315	24	414	983	1135	860	11	452	937
539	337	812	935	639	507	863	378	637	301	1055	1219
1077	1198	440	943	451	658	478	361	392	82	103	586
693	881	1210	652	1235	37	1001	970	1156	1175	808	184
278	662	799	341	1041	244	736	1030	109	730	984	195
1019	678	728	420	294	1212	796	1218	303	1192	627	645
1079	1109	1046	481	272	488	1116	1096	222	238	21	349
1034	2	1128	1011	130	157	445	167	705	1087	1201	33
470	489	10	598	453	128	39	1061	817	1209	1216	741
926	1184	1008	942	524	969	372	308	118	228	473	815
312	623	1122	516	1078	962	1143	929	1233	989	316	602
941	1125	1223	648	1072	1136	58	1062	224	931	571	1173
987	649	158	1040	612	998	233	864	241	140	9	621
177	88	1047	321	57	1000	1037	1117	340	1151	961	1221
595	973	138	152	959	483	213	479	207	477	191	287
740	522	977	654	284	227	332	141	234	105	842	305
1080	957	572	355	310	1140	660	258	78	322	831	1054
201	1229	467	1045	273	353	629	441	618	63	1074	849
527	924	43	173	710	295	948	708	1057	876	606	36
967	1025	1081	1147	1132	868	979	1082	966	844	1237	1010
1239	845	772	51	391	1168	855	865	823	65	62	874
275	276	159	885	182	1105	1118	1002	603	592	999	759
1035	762	1069	1083	1007	1005	932	1094	824	745	110	828
154	677	638	703	851	204	1180	806	208	760	884	771
610	1230	25	1211	439	925	1022	765	1155	670	672	519
469	356	548	679	1056	1015	236	1167	964	803	1106	218
1194	591	663	587	260	50	69	1141	953	1071	975	701
212	396	550	307	1063	203	811	743	510	84	333	1153
739	566	807	1159	1018	162	1157	385	1023	35	1114	22
644	1108	261	1093	711	963	296	328	706	122	358	597
938	965	504	1197	249	788	1098	790	325	1214	1228	401
1027	525	1086	951	624	529	1099	992	1150	136	667	380
1126	1111	199	1048	1206	580	404	697	251	719	475	40
585	1044	1095	1053	686	350	575	1163	1003	1060	1064	223
950	146	986	1177	752	304	726	123	853	742	168	492
454	766	804	1021	533	247	46	939	42	1121	221	555
226	980	327	17	180	974	798	927	209	1152	692	554
370	126	151	129	596	377	800	1169	1146	408	463	417
569	1103	615	498	326	59	456	1186	1049	137	139	291
427	632	685	1113	433	1220	153	1164	722	354	189	976
523	100	763	509	1016	444	399	369	1065	936	825	220
360	265	947	1059	749	568	1090	838	852	631	365	993

329	268	651	1330	390	500	1331	1332	460	320	468	729
1333	1274	583	617	753	564	56	782	1306	661	1334	263
165	1335	1318	795	713	371	367	1336	134	1305	271	748
1337	81	532	1299	343	486	1338	1339	472	1264	1302	413
1340	362	769	801	633	1273	517	534	867	94	335	732
1247	1290	1316	1260	89	781	718	1341	450	1323	551	1342
1343	1344	894	1292	687	515	1345	608	1346	1347	1246	351
368	767	1248	570	430	1322	735	235	1348	1349	1284	1350
1288	1278	802	847	257	499	746	805	1267	114	1295	514
1351	188	785	558	1317	1352	784	70	537	832	1353	448
181	1354	297	99	44	442	217	609	501	410	1355	837
1356	66	1314	313	38	1357	777	541	559	1358	1320	426
170	1312	1319	1279	186	32	882	171	429	776	1359	1360
880	720	671	1275	406	215	431	1253	347	792	717	1361
200	1362	1313	1363	54	754	1251	476	511	1364	435	1325
1365	1291	338	1257	1315	699	1282	641	712	381	96	1327
482	883	1280	421	1366	513	1367	26	1368	1250	779	1276
1326	242	143	1307	300	1249	1369	508	23	721	690	1298
773	895	887	1270	875	53	576	67	589	455	890	636
626	1328	405	536	647	582	119	656	696	1370	282	290
334	619	1371	1372	512	503	1373	1374	854	544	877	1375
205	493	106	1308	783	77	577	293	1376	1254	344	1252
148	526	256	1377	464	616	214	878	1294	1303	1321	1378
1297	1379	1272	1311	1300	1266	240	590	1380	530	243	1287
567	1258	1293	160	495	707	150	1381	666	702	873	505
528	674	1382	562	694	86	889	397	1277	93	1296	1285
1256	104	131	270	1383	311	49	190	1281	841	388	734
1283	1265	346	775	115	600	1263	1255	1384	352	1310	172
1259	1309	561	869	13	691	142	133	871	373	821	836
269	502	75	1385	253	1386	116	318	1301	376	1387	556
348	74	178	1268	724	107	1324	553	7	1388	1286	192
1389	29	73	780	1329	750	538	797	471	1262	1390	419
757	1391	1261	1392	1304	1289	496	4	700	1269	383	1271
1393											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets) :

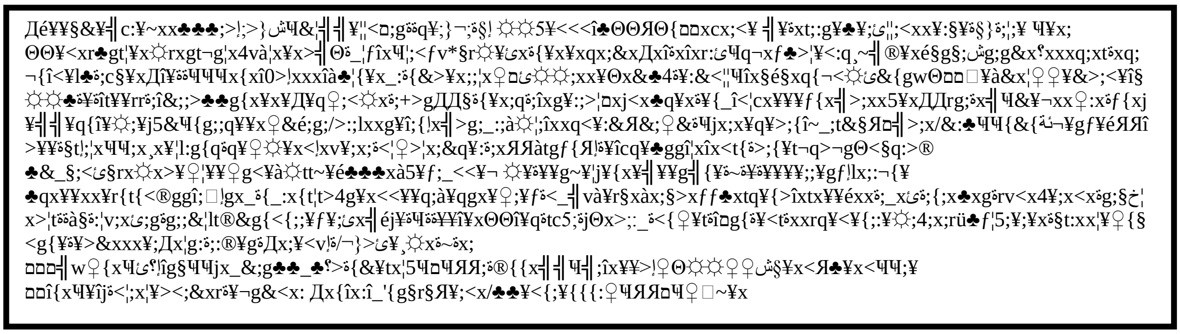
80	34	87	94	51	64	91	46	18	32	55	81
44	2	63	86	54	72	45	16	88	84	20	75
82	97	39	13	6	36	67	95	12	53	17	92
24	98	89	14	74	60	25	41	40	76	10	57
66	93	5	11	9	70	3	31	73	61	42	43
23	27	59	28	78	37	52	69	7	71	77	8
99	50	85	96	30	21	29	22	58	19	90	68
26	47	79	15	35	48	38	878	603	723	164	621
401	997	320	274	4	126	364	628	908	150	286	287
600	815	170	482	424	159	307	220	139	213	596	967
217	133	405	644	511	861	909	334	802	964	597	691
449	942	485	721	538	945	862	870	495	513	505	957
623	883	229	506	955	419	767	751	917	56	143	237
831	972	799	62	324	224	479	750	864	255	771	330
378	316	474	564	161	403	172	748	461	223	978	778
166	318	904	540	985	167	741	202	467	569	49	434
793	283	765	568	248	102	250	653	816	359	297	602
232	249	207	196	365	190	754	947	455	490	338	931
788	454	698	649	774	308	588	702	708	631	33	444
557	463	503	913	886	1	715	116	550	516	189	849
244	685	458	584	285	764	975	418	589	877	921	487
309	632	144	701	850	317	932	563	239	643	610	192
348	574	533	732	993	212	809	299	417	790	501	464
922	567	867	331	650	231	466	722	895	666	168	891
529	541	558	762	279	730	635	824	486	177	313	520
581	560	607	140	977	719	859	339	638	599	380	491
662	777	532	890	758	169	745	382	83	580	900	965
534	987	968	395	104	892	107	826	225	670	222	200
893	387	992	292	545	499	961	353	590	919	869	291
772	575	191	160	605	305	180	811	142	410	832	236
450	389	868	314	659	699	163	247	507	335	481	293
110	982	673	214	634	363	675	842	65	995	343	576
814	546	175	488	425	280	910	677	680	265	718	438
497	716	875	414	366	692	710	976	344	548	970	376

851	421	781	837	927	792	388	874	860	836	798	570
627	743	912	937	204	310	803	203	509	686	306	665
131	776	728	808	682	938	657	197	843	735	779	549
739	593	179	640	846	185	717	370	753	379	178	360
857	465	579	565	899	756	158	452	606	737	132	326
629	729	498	784	295	429	221	648	925	375	500	385
408	182	235	616	431	654	768	171	478	346	694	300
609	994	797	125	357	926	766	863	103	257	145	789
924	420	696	321	528	329	646	476	981	148	402	123
227	818	374	396	954	833	749	121	233	198	935	795
462	720	105	153	897	724	705	416	604	230	536	655
539	866	228	697	246	773	998	526	951	426	352	974
591	988	852	647	573	806	642	127	427	100	619	404
713	459	782	328	783	827	433	496	551	208	349	880
807	369	146	371	118	523	181	521	928	613	277	439
939	661	645	946	510	393	484	633	578	601	106	906
787	887	819	278	907	902	553	276	151	770	272	134
394	681	165	358	432	668	810	556	399	129	820	350
136	117	205	812	622	262	817	652	984	109	260	397
916	448	618	559	898	530	742	780	973	689	885	264
847	245	514	687	630	747	524	986	493	303	259	477
270	267	195	442	958	991	215	547	347	269	914	162
667	114	571	537	733	663	124	865	377	586	583	587
761	184	936	135	744	940	368	296	684	941	712	592
137	113	206	706	412	933	241	594	271	688	822	543
561	980	216	332	844	660	535	304	996	949	894	731
423	341	755	734	284	356	918	183	201	504	830	392
390	585	956	457	446	440	492	775	174	651	582	841
456	615	199	149	554	522	238	362	266	813	796	111
959	989	614	876	953	608	119	409	282	544	428	834
639	351	354	664	637	855	301	873	517	519	625	896
963	725	400	736	355	209	595	290	671	508	835	658
381	256	430	979	345	871	786	901	176	714	839	288
437	120	319	243	840	598	983	251	157	226	752	342
700	669	821	435	856	470	273	407	315	726	234	888
406	727	252	704	340	473	443	929	542	823	740	337
785	641	384	693	460	872	218	445	469	155	193	853
386	828	138	298	999	854	881	943	483	101	759	769
147	930	611	678	112	672	624	709	311	141	577	518
791	422	626	480	336	612	950	905	948	617	327	920
471	128	829	562	695	884	240	858	281	122	254	323
367	707	683	453	794	512	711	889	944	475	263	302
372	186	656	294	911	531	845	969	253	738	801	805
966	489	472	383	289	990	210	258	441	879	391	451
525	322	188	763	173	312	413	261	154	962	515	156
800	971	411	882	219	436	398	527	415	555	333	760
674	757	447	620	572	268	952	242	115	502	494	187
194	152	552	804	703	468	108	903	838	636	130	275
960	325	848	676	923	825	679	690	361	211	746	566
934	915	373	1097	1243	1324	1351	1271	1352	1040	1080	1241
1277	1136	1353	1089	1051	1354	1355	1039	1077	1342	1258	1320
1037	1064	1202	1242	1312	1281	1123	1054	1170	1267	1285	1222
1340	1349	1129	1111	1296	1174	1252	1345	1356	1166	1357	1115
1275	1328	1250	1128	1263	1171	1098	1284	1033	1231	1276	1005
1058	1313	1311	1059	1124	1232	1157	1013	1100	1132	1019	1099
1103	1023	1112	1225	1240	1304	1194	1215	1358	1247	1198	1334
1335	1122	1018	1210	1164	1341	1156	1127	1106	1006	1300	1323
1114	1359	1245	1113	1066	1154	1150	1036	1344	1289	1360	1238
1298	1034	1035	1093	1048	1008	1318	1253	1309	1028	1331	1361
1016	1195	1286	1362	1363	1050	1090	1038	1137	1119	1287	1079
1163	1182	1151	1125	1043	1303	1025	1325	1133	1042	1364	1002
1234	1075	1347	1088	1365	1212	1117	1230	1301	1294	1071	1214
1200	1307	1049	1264	1190	1060	1147	1299	1332	1160	1165	1366
1109	1180	1343	1305	1315	1065	1187	1055	1052	1191	1367	1110
1196	1368	1074	1205	1257	1262	1168	1336	1108	1308	1045	1369
1330	1278	1239	1134	1221	1149	1235	1104	1209	1370	1053	1265
1282	1107	1094	1233	1069	1153	1213	1244	1327	1266	1146	1229
1091	1141	1105	1346	1226	1041	1070	1186	1142	1056	1302	1063
1199	1321	1162	1193	1086	1101	1270	1083	1310	1017	1192	1177
1179	1095	1248	1184	1219	1121	1022	1273	1172	1371	1046	1372

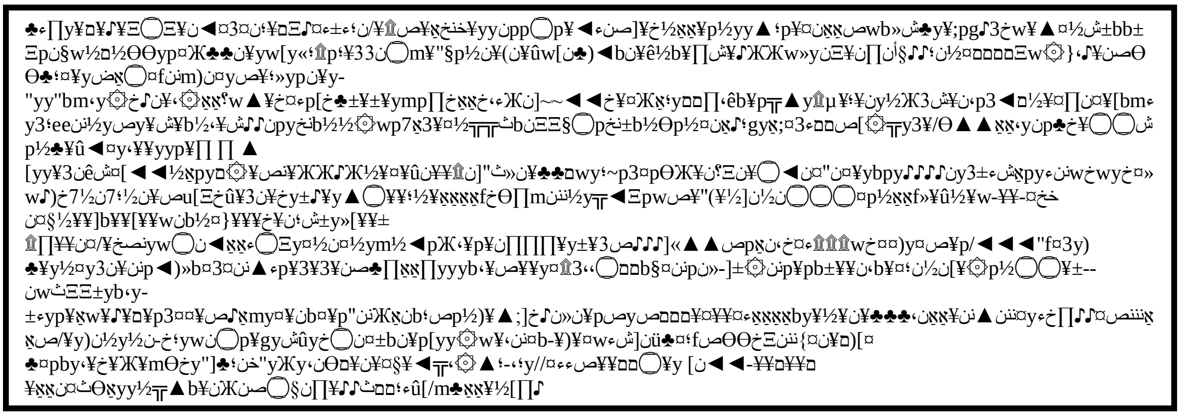
1044	1178	1096	1208	1224	1183	1003	1084	1012	1131	1130	1348
1062	1316	1159	1188	1021	1373	1009	1135	1374	1375	1203	1211
1228	1010	1087	1268	1376	1333	1081	1377	1322	1220	1145	1072
1057	1288	1061	1047	1249	1255	1120	1030	1306	1261	1189	1246
1314	1144	1378	1254	1379	1126	1295	1138	1176	1319	1152	1337
1216	1201	1236	1380	1293	1076	1185	1291	1158	1143	1381	1029
1297	1067	1259	1382	1140	1155	1068	1269	1031	1383	1384	1118
1339	1260	1290	1102	1385	1386	1082	1280	1167	1000	1078	1085
1073	1387	1197	1272	1007	1181	1227	1011	1001	1161	1206	1251
1116	1204	1388	1173	1032	1350	1024	1389	1207	1292	1217	1326
1237	1390	1223	1218	1148	1391	1392	1279	1274	1026	1027	1256
1092	1014	1015	1393	1317	1139	1338	1329	1169	1004	1175	1283
1020											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)



(b)



(c)

Figure IV.5. (a) *Donnée originale Texte2,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

700	340	911	855	661	6	451	910	17	3	151	267
937	143	936	914	475	880	745	874	939	897	155	935
281	730	209	938	173	638	777	191	329	930	384	421
673	131	485	462	512	259	736	644	457	728	932	593
933	65	920	924	926	918	349	196	232	309	553	782
917	369	919	927	455	878	255	921	913	929	156	931
912	759	400	922	616	915	594	691	243	934	592	916
819	361	128	923	486	24	925	159	748	301	928	140
411	379	1093	16	599	1169	1108	1079	1040	985	1021	141
799	1006	1067	1107	82	572	1014	325	977	498	9	943
974	670	394	805	270	49	178	658	808	604	392	1229
642	456	1058	80	1184	78	669	472	1045	1214	445	1011
195	492	242	952	389	1069	814	839	1060	591	983	895
285	248	67	947	1194	460	380	38	46	584	482	430
631	1236	478	1012	1004	297	570	1240	228	332	390	1127
1153	40	992	991	1151	625	44	1106	47	1189	99	823
986	706	395	85	559	205	28	641	887	197	319	979
1094	554	882	1066	822	1166	531	564	949	305	398	654
694	544	200	160	801	1017	189	1031	1083	1143	476	1196
1002	198	894	348	1052	619	69	214	1192	1099	120	1134
318	300	534	73	862	104	425	328	1092	1141	677	296
692	1212	466	612	1190	144	960	686	1188	53	404	668
683	381	106	672	523	1145	688	42	1053	342	102	779
292	757	1055	1183	852	514	34	890	702	525	1233	863
870	79	444	858	1241	851	904	1250	948	346	224	31
807	754	528	942	794	1140	511	1101	347	295	1198	289
436	408	587	308	269	403	247	1074	25	1148	27	264
194	489	1160	163	1056	1225	298	322	965	1202	563	237
1035	1179	941	56	1051	1057	1064	434	978	1015	532	413
602	1049	71	772	162	1123	483	1142	231	1152	844	1078
1235	1159	356	418	1047	1082	737	273	1137	676	1144	263
372	1039	530	1098	440	284	1191	1081	94	1205	29	783
1216	603	630	859	877	598	473	4	274	944	674	681
709	1154	565	1008	837	662	549	671	1003	215	710	1228
1243	758	1016	962	1238	175	1217	1158	518	57	1172	628
764	876	1223	606	7	253	657	968	1245	766	999	542
324	775	399	738	989	1097	1009	515	238	406	770	245
993	499	1167	30	217	230	1077	299	787	74	537	1029
784	802	645	76	1147	1197	415	452	1126	45	517	629
1227	1000	940	569	632	618	164	172	101	18	88	216
1135	950	397	377	879	809	1226	1176	527	227	1203	431
833	276	970	87	1027	256	480	843	409	637	286	1013
1248	142	725	605	743	1114	567	643	791	640	576	1090
1244	150	689	1018	650	589	311	516	995	1178	519	1175
611	828	621	857	1122	1020	1065	294	474	1230	959	376
869	501	898	370	545	1117	816	1071	1110	43	526	623
115	836	655	680	26	494	1195	447	1023	385	513	548
969	161	250	1185	1034	110	111	860	15	561	793	1170
961	820	133	432	1116	97	114	279	93	193	495	422
1246	1059	997	112	212	1210	1130	33	219	60	469	698
1231	174	812	479	763	1207	1218	277	50	450	416	41
954	582	116	488	127	1128	552	697	1209	1080	306	490
1085	1129	539	981	953	337	449	211	726	1061	1124	510
715	1091	861	83	1007	827	477	330	800	849	546	8
747	419	10	186	454	500	66	583	899	357	401	1001
971	659	856	226	1234	240	233	91	761	343	1046	1146
595	107	1186	1237	1103	1139	1076	1048	647	579	722	353
608	386	660	1115	596	1042	610	636	405	718	290	529
1138	234	433	1112	1164	964	945	742	967	773	20	77
1070	1221	1072	804	1224	663	1161	721	1120	1118	458	797
1193	707	1119	326	453	35	505	1030	533	126	1041	64
769	753	14	63	1024	1100	387	149	744	239	1200	703
1037	504	522	32	956	875	536	723	998	345	1171	435
1068	1155	366	832	1213	491	1028	1173	829	429	23	374
1181	976	11	973	393	1187	988	1232	135	1102	169	493
278	664	134	90	821	885	1163	762	741	627	108	892
1199	996	367	958	982	1165	417	1174	95	1150	557	1033
740	333	891	382	129	972	1247	1109	1121	266	204	1206

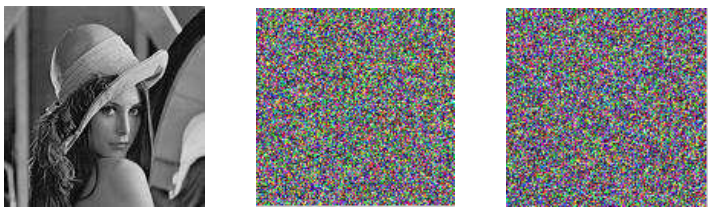
1010	388	187	966	1204	955	987	70	790	1239	503	352
957	262	1084	272	59	866	414	1168	1219	1201	1222	907
1111	317	963	1113	1026	497	487	818	810	109	442	439
190	780	423	896	130	363	1086	798	617	1050	1087	994
1105	208	1211	580	796	89	291	68	1025	251	1182	1005
682	1104	378	980	538	646	118	323	699	1157	359	1136
236	1156	1131	1075	1043	1149	853	96	229	351	179	1220
180	138	145	424	177	1019	1032	560	975	136	1132	1125
261	774	685	568	886	675	315	202	158	1036	168	1133
713	601	1215	600	176	1054	1208	817	865	678	739	864
789	1095	1162	509	1022	614	327	1249	946	607	806	760
185	1062	221	667	902	566	316	635	665	39	813	717
786	951	834	624	438	1088	58	1177	984	1044	62	719
1251	900	1096	100	1073	166	990	695	1180	463	2	334
124	1242	1038	123	148	889	615	121	1089	651	407	481
749	1063	1271	470	86	597	465	1332	1307	260	1342	1275
881	368	371	48	1343	1301	1270	1344	838	1313	1345	666
1346	132	265	871	302	840	84	502	842	1347	222	765
573	252	1348	1322	732	246	1349	835	1334	13	1319	1339
464	1350	778	771	1351	355	268	590	845	1311	210	223
1289	577	184	1282	1352	735	1269	1293	1353	1278	459	1263
241	207	1274	1354	652	1325	1298	446	12	588	1268	873
884	1355	622	1252	125	344	551	213	304	1356	1265	1357
1296	609	1330	908	693	98	5	868	555	1320	225	428
687	1331	383	541	1308	81	188	756	1267	558	154	1272
1358	1359	341	1258	52	847	467	1360	776	724	1361	22
1279	103	280	36	220	1362	795	634	578	461	571	484
181	303	321	690	1363	734	905	1285	574	19	733	412
396	313	402	1364	1365	331	139	550	51	1366	1281	365
335	746	1367	586	257	336	613	287	1253	803	883	792
653	850	72	585	310	831	1329	360	729	192	1368	218
426	21	496	727	1273	815	750	320	1326	1318	1369	1327
1370	1371	639	684	1262	358	1372	903	1373	1257	581	825
312	906	117	1374	1341	649	1375	767	1376	1264	556	1377
373	235	1378	826	1379	1336	731	872	1316	203	468	1304
1380	848	708	167	543	1291	1261	626	1256	888	712	206
1340	307	1294	854	535	338	1337	1315	1259	909	410	37
244	620	1277	443	1338	362	75	1287	471	867	153	768
575	1381	811	824	1314	171	1323	105	1254	1290	1328	271
1303	1266	846	183	441	1300	648	521	1324	1255	146	893
375	1382	1383	508	354	1283	696	54	254	282	283	656
785	788	448	841	1310	506	137	1317	1295	165	830	1333
339	1286	701	714	679	751	1299	540	704	249	147	182
1260	1384	364	1302	157	293	1306	1385	633	92	420	1309
427	288	716	1335	391	1321	55	258	1280	1288	201	61
1386	152	1387	1388	901	705	350	1389	752	1390	1	1391
720	711	755	113	562	1284	1292	1392	1297	314	437	547
1393	507	1305	520	1312	199	119	524	781	170	122	275
1276											

Clé de chiffrement de la deuxième version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

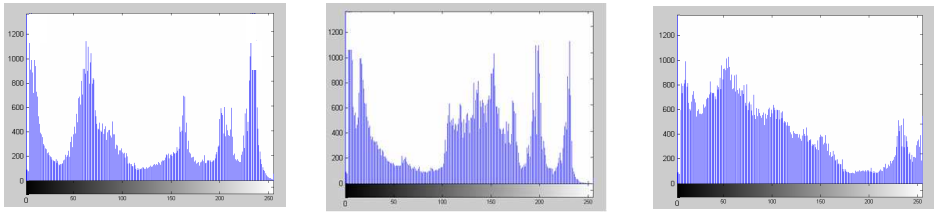
6	162	2	1	279	439	507	352	360	804	758	567
472	465	78	992	500	327	830	580	948	101	535	173
756	991	805	522	928	10	147	416	461	606	769	803
827	692	341	982	129	12	244	944	197	477	831	139
684	658	255	593	356	49	36	172	504	108	272	375
248	575	302	299	81	358	496	773	607	476	852	562
382	726	134	846	667	641	98	73	735	230	548	540
902	484	64	727	643	542	640	380	556	632	723	
835	436	748	109	721	654	619	212	277	894	211	475
705	335	257	844	856	781	938	462	961	11	623	62
651	254	74	4	746	265	525	130	943	185	52	473
457	160	251	860	59	939	447	121	946	511	757	934
258	136	974	350	35	50	483	491	965	32	941	453
339	344	514	929	137	470	27	502	724	131	184	512
362	677	213	871	802	981	39	5	570	345	848	76
328	884	412	238	68	391	376	488	325	292	882	71
142	970	810	196	220	796	730	132	296	622	550	490
590	574	3	560	87	826	158	8	518	817	973	635

819	409	689	498	206	353	526	890	41	617	440	647
396	915	935	797	96	242	975	297	285	608	177	264
782	563	99	203	659	7	544	104	761	629	719	271
333	329	877	581	332	838	778	611	704	762	674	156
972	899	610	930	366	785	927	976	956	31	263	117
312	315	664	775	752	337	413	996	878	853	963	46
637	866	111	538	310	105	30	418	445	179	814	933
151	867	597	274	86	467	609	984	273	152	834	43
408	303	594	44	505	696	419	516	631	732	794	712
284	588	682	435	874	521	858	589	969	906	33	138
398	722	698	529	372	524	93	378	122	486	554	494
448	655	103	833	547	808	9	148	924	855	728	713
386	691	751	226	468	239	311	688	482	628	28	737
753	566	373	602	543	168	118	318	326	813	14	307
309	275	485	818	942	870	653	481	868	950	993	861
690	499	766	648	443	349	679	280	493	673	824	56
424	199	321	889	621	962	166	218	977	714	487	444
336	582	281	402	624	553	387	747	441	390	546	261
743	306	492	603	58	616	267	235	89	528	703	630
811	364	497	381	715	800	250	665	893	94	379	361
913	627	532	262	530	585	980	295	388	21	697	241
903	828	971	851	995	686	672	469	888	615	519	905
595	908	501	832	169	433	897	718	63	508	228	80
816	385	922	115	862	91	423	428	95	559	427	873
786	779	676	909	15	604	979	710	354	680	370	578
601	739	150	245	157	907	342	128	953	417	454	404
881	223	240	82	17	270	646	792	125	153	940	896
266	694	571	552	135	127	898	986	901	669	772	229
842	837	916	346	371	523	47	661	464	771	783	839
790	18	308	460	577	702	154	289	770	990	57	463
114	69	24	642	400	120	671	232	650	97	48	249
774	895	605	178	919	314	863	442	926	53	558	634
921	989	458	29	587	568	741	406	434	145	657	561
791	920	369	340	246	645	821	749	429	614	403	539
330	509	144	945	225	918	541	394	112	537	706	850
405	60	879	900	845	222	374	789	474	625	549	425
209	584	276	317	421	652	426	67	843	599	198	355
947	338	557	305	37	431	319	678	290	876	155	638
596	806	451	331	124	744	809	54	411	51	182	883
912	825	170	807	16	983	857	904	666	836	955	660
988	61	175	420	159	395	742	951	515	133	26	869
815	268	410	205	219	176	191	446	729	347	252	864
534	958	849	207	466	234	65	531	90	269	75	478
384	221	181	795	183	551	22	968	459	911	479	399
414	286	734	670	192	685	925	959	545	777	300	287
174	415	357	106	88	840	960	987	42	859	636	780
471	745	294	701	592	164	600	750	733	377	764	320
949	208	422	716	681	291	140	301	449	711	83	393
165	668	343	952	84	917	700	892	126	113	639	195
954	759	966	573	85	644	102	231	224	92	885	914
180	143	613	40	141	994	887	348	875	200	20	367
188	316	456	110	720	489	503	569	450	260	626	822
119	38	383	793	767	708	186	171	167	579	736	812
572	72	784	163	77	365	555	392	243	765	452	146
283	187	322	282	107	760	683	886	407	693	662	216
841	23	707	363	564	210	359	55	999	964	401	738
217	324	847	506	598	116	201	13	586	253	576	695
823	798	865	591	699	323	256	190	389	872	788	620
288	829	533	763	247	717	313	298	19	985	854	149
278	520	931	998	455	100	687	820	34	351	768	193
923	967	740	755	480	432	25	227	725	161	799	932
189	675	937	663	204	66	957	237	565	334	891	583
618	787	233	304	259	397	997	536	656	612	437	527
495	202	368	936	510	438	633	649	801	754	910	776
880	123	194	214	236	513	978	293	215	430	517	709
731	1165	1126	1054	1145	1227	1122	1168	1335	1190	1338	1157
1224	1003	1310	1038	45	1301	1153	1295	1316	1032	1318	1329
1167	1115	70	1004	1339	1340	1341	1081	1150	1294	1342	1343
1319	1086	1125	1018	1269	1118	1012	1214	1344	1129	79	1332
1061	1213	1111	1205	1305	1130	1252	1093	1315	1057	1037	1299

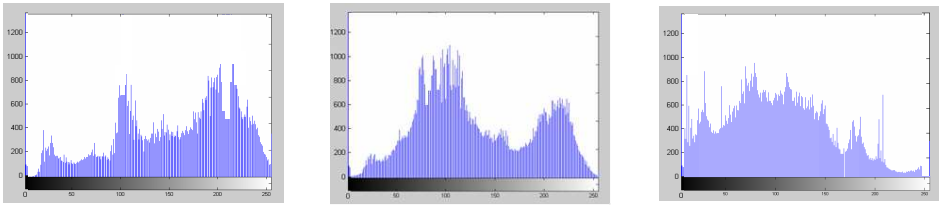
1083	1277	1062	1219	1345	1238	1346	1096	1221	1022	1103	1195
1257	1142	1050	1347	1274	1077	1088	1006	1058	1220	1046	1020
1171	1268	1116	1348	1278	1013	1188	1291	1237	1314	1296	1349
1350	1351	1109	1192	1352	1353	1189	1267	1229	1005	1155	1107
1243	1060	1008	1091	1128	1064	1023	1041	1293	1245	1248	1102
1113	1354	1263	1131	1355	1163	1356	1015	1039	1357	1247	1000
1053	1311	1358	1283	1049	1124	1065	1359	1360	1361	1281	1173
1254	1181	1114	1276	1191	1336	1104	1223	1184	1362	1079	1123
1226	1259	1137	1273	1112	1070	1363	1287	1035	1197	1364	1365
1140	1366	1367	1368	1127	1288	1369	1199	1244	1010	1133	1105
1024	1370	1371	1048	1180	1312	1317	1045	1139	1330	1177	1019
1040	1029	1055	1230	1225	1285	1275	1334	1372	1085	1036	1076
1206	1258	1208	1333	1028	1002	1154	1320	1186	1166	1161	1030
1135	1261	1095	1101	1067	1280	1136	1194	1097	1302	1260	1099
1034	1373	1172	1063	1374	1132	1094	1080	1337	1234	1265	1175
1066	1169	1047	1087	1246	1121	1375	1193	1068	1292	1376	1377
1138	1378	1325	1075	1326	1232	1203	1249	1092	1073	1379	1297
1152	1001	1071	1306	1026	1256	1242	1304	1380	1148	1156	1321
1201	1303	1021	1239	1210	1222	1381	1290	1251	1204	1025	1264
1382	1383	1084	1187	1228	1271	1146	1185	1078	1324	1384	1385
1215	1216	1176	1170	1100	1098	1212	1016	1031	1218	1209	1159
1196	1217	1178	1233	1241	1106	1162	1266	1141	1143	1090	1240
1250	1298	1328	1108	1307	1386	1262	1089	1387	1074	1027	1313
1388	1389	1056	1179	1009	1151	1119	1044	1255	1331	1134	1286
1272	1117	1033	1147	1231	1052	1149	1207	1309	1200	1164	1282
1390	1007	1160	1198	1082	1072	1308	1284	1253	1174	1069	1391
1202	1392	1182	1017	1270	1235	1236	1042	1043	1211	1322	1051
1183	1011	1120	1323	1158	1289	1279	1327	1110	1393	1014	1300
1059	1144										



(a) ***(b)*** ***(c)***



(d)



(e)

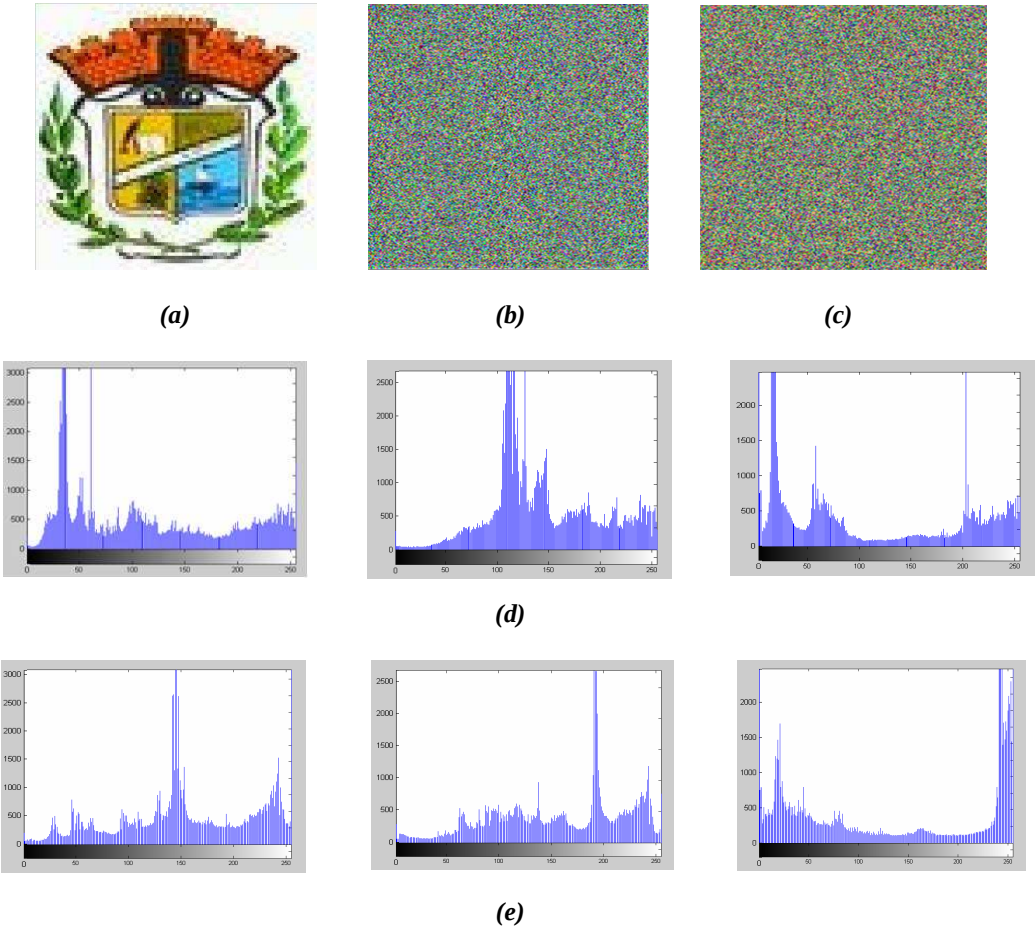
Figure IV.6. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

5	623	186	3	733	275	321	55	398	496	437	199
363	266	63	526	685	316	438	200	324	525	595	106
194	114	109	386	198	126	674	374	751	755	176	273
541	467	369	279	516	661	197	609	447	4	86	679
716	196	117	422	736	430	129	474	261	762	527	630
759	737	615	362	694	367	687	583	581	646	19	676
234	645	203	574	284	580	332	389	517	613	743	637
732	272	209	431	453	408	495	557	651	394	30	242
337	735	551	566	473	477	391	104	50	79	401	11
535	514	500	269	352	143	372	216	664	267	341	499
184	713	543	553	458	300	22	51	718	325	712	68
532	704	512	501	1	7	547	690	763	256	522	536
130	381	215	511	224	371	647	94	485	70	78	494
761	220	436	622	21	577	149	725	582	443	322	334
191	576	424	697	513	10	296	96	550	345	492	88
154	585	506	239	549	456	519	590	702	399	426	505
212	339	120	727	631	692	166	159	201	67	684	742
102	440	537	225	29	655	709	562	567	308	171	644
539	452	264	502	592	416	510	274	111	753	621	561
235	208	226	45	468	370	175	471	410	747	349	42
304	387	677	230	439	157	534	121	170	657	8	552
617	213	691	455	150	43	223	678	484	95	301	36
442	503	598	579	449	610	731	156	635	563	90	249
103	247	432	524	206	633	6	533	584	160	564	721
754	750	91	538	162	722	508	285	250	353	254	744
726	717	597	283	185	559	625	393	570	669	69	504
174	421	457	329	355	87	268	616	136	195	639	336
100	720	668	518	54	116	214	605	319	303	190	58
23	405	342	548	428	376	231	470	373	34	320	670
49	475	132	181	588	271	418	314	167	703	243	404
364	123	419	297	227	48	291	624	461	81	528	738
192	105	766	619	62	25	187	330	115	636	441	450
152	545	760	498	365	221	71	383	153	530	134	660
662	189	757	396	348	368	602	596	238	714	205	671
749	614	435	493	317	395	672	307	182	481	17	47
415	338	27	57	682	640	648	765	60	219	356	689
155	173	673	629	683	529	600	658	448	32	292	384
125	172	327	681	113	740	351	603	24	312	675	148
40	643	15	309	402	294	93	217	232	122	434	135
326	183	281	315	593	480	118	53	288	358	89	142
128	39	699	131	241	41	521	66	343	282	741	734
218	277	420	202	540	13	642	486	589	204	723	478
730	715	340	710	764	80	653	472	211	445	483	411
459	464	433	18	392	138	618	406	377	571	193	591
360	366	158	680	252	469	382	145	52	35	462	72
407	278	133	73	101	412	26	165	745	344	575	12
599	400	739	695	748	310	568	139	293	140	608	403
375	708	587	768	298	666	76	554	251	606	414	572
245	97	222	74	270	507	388	44	601	696	546	594
229	107	706	168	311	350	487	112	667	255	33	444
361	177	454	169	164	479	509	82	686	463	515	228
663	244	357	649	711	259	628	265	397	707	728	423
665	2	632	626	460	654	210	290	124	767	163	333
280	429	295	489	451	698	565	127	37	641	491	688
161	531	302	719	179	75	729	724	604	482	693	347
240	413	61	490	746	378	586	258	84	627	427	248
263	335	16	542	523	233	556	31	144	257	38	64
331	92	544	701	650	466	56	573	188	659	306	578
253	287	756	236	425	299	119	611	65	9	555	323
141	417	110	318	359	20	146	77	656	758	652	207
289	151	237	98	385	620	700	85	137	634	147	260
354	180	276	560	59	380	465	313	569	346	409	638
558	108	328	612	476	488	46	446	262	286	246	178
607	83	705	752	390	497	99	305	28	379	14	520

Clé de chiffrement de la deuxième version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

81	506	289	165	107	184	553	751	570	17	729	724
457	204	65	760	510	256	116	708	314	296	315	381
538	6	734	302	480	33	139	345	54	144	113	57
393	424	399	163	470	71	599	733	51	653	151	30
175	743	321	556	187	232	112	323	363	707	588	183
235	260	128	37	477	453	462	267	88	304	351	56
308	359	762	464	121	374	294	382	715	392	478	609
647	205	431	253	89	594	666	262	622	592	224	322
311	164	277	29	341	291	154	646	699	522	495	74
625	47	320	395	644	77	43	100	494	723	402	706
702	105	152	244	283	737	404	340	750	629	396	764
125	412	49	387	22	364	484	643	337	242	131	389
342	690	120	572	567	435	745	176	466	13	209	403
517	508	279	265	4	252	117	132	475	587	391	310
635	331	492	502	360	372	380	456	710	104	96	669
565	416	257	531	3	326	612	171	568	1	563	444
434	507	41	259	505	577	84	192	147	447	597	747
347	711	335	744	383	421	540	367	394	660	143	281
657	178	339	418	562	436	301	766	27	94	516	66
703	161	91	313	58	63	136	610	203	45	162	292
195	483	317	705	228	62	196	476	18	539	64	368
141	451	503	39	169	307	229	297	672	350	670	626
596	309	9	19	740	353	519	160	673	134	614	400
541	75	649	605	748	223	82	137	388	525	667	655
535	640	90	585	659	448	243	295	170	48	299	237
754	533	726	452	548	528	231	230	590	700	24	586
182	678	732	101	573	290	156	272	207	631	607	701
177	445	155	365	545	73	38	583	284	80	560	138
591	469	559	97	423	398	119	208	603	527	512	514
767	78	12	173	490	95	324	126	619	349	69	488
665	662	460	693	459	180	420	604	509	720	598	595
529	174	5	50	87	271	683	561	696	422	188	59
245	620	439	482	240	580	361	385	685	593	755	142
358	274	415	497	571	698	677	370	536	679	31	663
641	219	468	736	375	222	189	757	432	158	216	639
471	413	633	455	23	85	642	407	168	651	675	238
206	450	123	146	465	518	276	739	546	319	199	145
611	106	689	636	688	714	461	768	348	632	214	133
442	333	46	129	118	344	654	10	68	722	627	379
581	575	758	730	390	661	411	682	172	249	334	14
652	695	11	336	576	487	278	725	268	384	61	630
191	520	211	521	239	589	756	26	286	409	417	127
227	501	55	298	426	713	735	35	148	159	437	401
186	330	99	638	618	550	410	226	15	44	40	430
742	103	140	458	111	369	449	373	537	202	615	83
270	674	472	600	564	608	397	645	215	234	721	130
261	153	269	515	579	316	405	741	621	727	628	656
254	354	352	122	72	763	76	287	34	752	181	446
544	21	236	566	441	524	617	300	513	79	248	467
115	731	474	408	704	491	305	547	557	377	20	427
719	697	102	357	52	346	526	709	582	185	718	149
542	481	7	716	318	761	606	312	288	201	429	109
549	493	338	285	378	454	86	438	613	419	601	329
303	293	712	110	728	247	433	157	443	282	212	218
197	634	694	746	489	135	558	623	717	680	246	648
264	664	366	574	676	124	362	36	53	8	213	200
530	668	114	555	543	485	650	498	233	42	150	325
166	98	511	343	749	738	414	552	753	251	250	255
356	280	386	266	551	28	273	328	93	499	275	92
486	108	198	534	32	532	194	70	569	217	691	658
406	578	16	263	327	2	221	496	765	463	332	681
616	500	25	428	554	479	306	67	637	584	371	190
692	473	258	60	759	220	425	440	671	523	167	225
210	355	684	602	179	687	376	193	686	624	504	241



(a) ***(b)*** ***(c)***

(d)

(e)

Figure IV.7. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée*
(d) *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

92	39	9	38	13	16	45	76	98	47	24	8
83	40	3	78	80	90	61	26	68	89	53	52
19	31	50	18	12	14	54	64	2	46	58	65
35	71	5	28	74	36	44	73	79	99	96	62
63	94	29	15	37	11	75	10	30	43	7	51
91	55	33	66	57	23	81	41	70	56	77	88
4	95	25	1	69	86	22	82	97	67	42	59
20	87	32	49	72	48	27	425	652	573	441	543
468	129	748	420	738	513	353	453	569	700	185	525
707	143	512	264	237	198	755	658	487	221	741	557
484	647	227	257	708	162	179	579	197	724	60	326
629	757	548	220	413	115	266	258	120	443	218	639
253	669	673	260	712	202	345	430	225	561	284	449
481	109	294	661	248	552	710	354	433	644	372	630
161	684	648	21	476	432	110	489	331	503	403	111
507	363	637	680	17	734	556	588	438	692	666	367
753	723	613	287	34	604	505	168	273	267	421	564
239	542	742	382	642	471	508	278	614	333	359	234
136	764	285	765	545	672	259	114	214	572	170	520
147	458	502	352	244	659	423	688	537	175	386	706
609	678	328	149	272	462	393	567	379	6	144	330
523	145	562	417	119	761	767	226	478	355	726	518
737	722	583	127	440	238	397	429	339	247	173	450
296	635	254	378	341	213	488	134	409	292	731	427

399	494	193	280	314	93	704	123	469	148	634	657
303	344	501	705	404	733	171	627	662	551	311	470
743	510	401	157	554	346	491	340	698	256	240	347
125	611	727	589	358	568	374	325	631	307	361	745
308	756	746	164	369	131	306	448	640	245	565	118
586	668	167	580	313	553	758	189	84	124	350	262
645	703	676	574	540	255	348	575	515	286	128	664
398	366	531	113	217	456	300	681	224	223	593	699
691	422	585	506	402	434	641	270	528	446	426	696
301	656	690	302	529	151	682	204	735	571	416	663
165	108	514	435	444	689	190	290	158	602	140	335
299	582	174	677	516	616	203	736	104	454	536	625
279	577	396	495	312	750	310	349	559	205	315	141
728	269	192	250	208	695	718	709	740	618	687	587
550	283	322	142	209	282	199	187	276	336	653	229
321	176	730	626	714	342	327	219	599	139	608	112
752	566	603	581	390	210	466	373	650	576	152	447
233	163	384	196	242	362	768	180	188	570	461	601
126	216	395	365	497	309	751	201	230	739	595	177
178	249	133	605	766	116	207	394	667	318	281	194
530	600	117	376	235	636	623	646	85	732	598	555
437	241	693	334	222	100	293	612	406	184	492	563
560	547	534	679	407	215	535	351	674	332	475	670
271	212	275	320	558	749	721	418	181	186	633	371
701	591	385	517	533	467	638	686	490	590	364	375
154	606	683	388	762	759	200	474	729	649	405	360
500	166	655	389	343	538	628	493	524	596	483	594
527	172	243	549	754	304	473	415	477	103	597	615
464	295	725	368	412	195	277	231	713	632	297	480
211	442	459	620	532	660	651	319	410	544	485	419
445	744	101	252	191	121	298	486	265	439	182	463
150	452	392	457	400	146	106	747	526	357	370	665
317	482	763	377	720	451	578	621	455	356	324	521
130	288	228	156	323	122	504	643	246	522	496	619
436	697	694	675	431	316	584	232	607	107	381	671
719	261	428	411	408	498	105	159	717	610	539	715
153	268	546	617	387	102	160	414	685	716	465	460
338	511	391	622	289	291	479	138	702	305	711	424
654	155	499	169	329	137	183	251	383	132	592	472
206	263	541	135	760	509	274	624	337	380	519	236

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

528	192	38	202	98	315	156	142	97	339	304	457
569	700	485	538	200	292	180	544	443	684	130	287
352	689	571	468	536	101	169	244	639	353	215	24
735	501	545	266	628	575	359	745	653	1	694	535
411	37	9	342	491	497	146	278	620	88	533	80
151	380	298	137	521	508	409	61	123	191	728	662
558	115	398	514	454	50	323	541	429	410	686	488
228	232	247	559	378	234	181	261	233	345	243	643
758	387	384	33	8	761	412	346	371	695	750	40
308	477	128	179	487	493	294	604	11	504	129	710
68	39	196	739	725	753	43	63	187	73	26	749
401	578	162	176	404	395	699	302	140	270	157	272
563	341	249	81	381	92	652	58	673	289	67	712
547	768	94	322	172	218	241	69	721	438	618	645
242	723	611	556	635	688	297	193	641	474	553	62
654	25	502	310	301	148	269	428	377	70	106	592
672	617	258	439	708	658	34	737	274	407	113	112
279	12	209	119	326	531	207	677	584	42	282	330
199	237	268	648	251	731	447	338	744	685	126	557
216	610	621	483	328	467	205	674	184	259	44	210
649	532	75	600	114	625	473	290	334	83	596	752
256	434	470	82	650	425	141	482	763	534	116	201
354	385	96	549	601	726	576	717	283	173	730	212
707	525	522	373	452	281	348	117	227	318	589	565
208	663	214	505	711	609	276	93	526	31	767	23
636	687	667	372	570	303	581	369	252	510	343	546
540	219	177	418	182	722	760	495	171	656	255	78

329	691	608	293	580	690	229	27	262	513	554	223
296	22	152	432	57	333	159	211	347	516	665	675
727	676	124	450	713	299	564	86	679	4	757	226
311	335	393	402	161	217	702	102	741	54	260	397
637	213	118	548	230	32	364	622	382	111	254	379
568	107	174	165	366	195	417	153	759	512	697	764
367	696	742	76	597	394	435	158	480	361	103	149
120	448	706	755	716	178	166	520	602	537	422	740
682	577	327	357	475	168	560	585	598	436	35	413
110	189	414	250	127	552	358	332	403	190	668	41
587	424	122	588	567	391	612	17	376	416	631	291
355	175	307	222	456	154	340	66	427	586	331	238
542	84	370	484	295	566	183	471	437	236	529	583
593	52	459	319	614	271	449	613	131	698	55	6
313	19	280	60	550	325	349	59	46	714	396	671
453	45	605	245	337	105	85	350	469	167	664	400
669	351	623	660	197	720	591	594	224	16	246	51
421	562	692	2	724	478	388	442	666	53	3	320
419	659	132	91	383	423	235	121	71	462	461	481
543	20	147	160	515	415	316	87	527	203	300	754
286	138	681	288	15	284	579	511	143	704	524	65
509	632	198	574	64	163	29	455	231	139	624	733
638	539	765	275	309	680	472	263	108	321	463	607
640	693	523	135	630	619	701	389	683	627	732	170
709	99	603	420	426	312	441	374	440	392	747	77
719	136	661	317	756	368	406	48	476	489	305	748
109	715	465	615	285	496	375	155	21	446	519	408
561	18	498	220	734	336	5	530	188	762	492	399
464	13	518	555	499	766	433	431	204	458	273	751
248	74	89	479	186	100	705	506	551	500	405	572
28	267	14	743	444	164	150	240	430	362	655	257
616	265	657	185	494	703	644	642	56	134	646	390
445	104	678	647	729	718	363	466	306	599	629	517
360	10	507	746	225	736	595	47	253	651	194	95
582	633	490	486	451	344	606	738	7	634	277	386
503	144	324	30	670	36	365	125	145	356	79	264
314	590	90	626	460	573	206	133	49	239	72	221

Le tableau suivant résume les résultats relatifs au temps de calcul (temps de chiffrement), la taille de clé, la quantité de phéromone et l'efficacité dans le cas de chiffrement des différentes données.

			Taille donnée (éléments)	**Taille clé (bits)**	**Efficacité**	**Temps calcul (s)**	**Phéromone**
Données Texte	Texte1	Version-Chiff1	1084	15323	11784.0	5.63	9,06
		Version-Chiff2			12030.0	6.02	9,13
	Texte2	Version-Chiff1	1151	15323	14498.0	6.7	8,67
		Version-Chiff2			14712.0	6.58	8,65
Données Images	Lena	Version-Chiff1	131 X 131	7680	135362.0	2.16	2.75
		Version-Chiff2			130074.0	2.31	1.17
	Logo	Version-Chiff1	420 X 395	7680	244926.0	2.87	23.11
		Version-Chiff2			226188.0	2.84	21.81

Tableau IV.2. *Résultats obtenus par AntCrypt.*

IV.4. Discussion et évaluation des résultats

La première chose à remarquer, d'après les résultats résumés à travers le tableau IV.2, c'est que le temps de chiffrement des données texte est beaucoup plus grand que celui des données images malgré que la taille des données images est généralement plus grande en comparaison à celle des données texte. Cela revient au fait que la taille du vecteur codant les données texte (1393 éléments) est beaucoup plus grande que celle du vecteur codant les données images (768 éléments), ce qui nécessitera beaucoup plus de temps de calcul lors de l'application du processus de résolution à base de fourmis. Toutefois, le temps de chiffrement est raisonnable pour les deux types de données texte ou images.

Nous remarquons, aussi, que les temps de calcul des différentes versions d'une même donnée sont proches, car ils sont indépendants de la taille de la donnée à chiffrer (une différence apparaît quand la dissemblance entre les tailles de deux données de même type sera importante et qui représente un temps de lecture ou de codage). En effet, la taille du vecteur codant toutes les données de même type et soumises au chiffrement est la même (1393 éléments pour le cas de texte / 768 éléments pour le cas d'images), donc sa manipulation sur l'ensemble des générations prend le même temps. La toute petite différence correspond au temps de codage des données avant le lancement du processus de chiffrement. Toutefois, la légère différence entre le temps de chiffrement des deux versions d'une même donnée revient au caractère aléatoire exploité dans certaines phases de l'algorithme.

Pour le cas de chiffrement des deux données images Lena et Logo, nous constatons que le temps de chiffrement de l'image Logo est plus grand que celui de l'image Lena. La différence est due au fait que le temps de chiffrement est la somme des temps de lecture puis codage de l'image qui varie suivant la taille de l'image, plus le temps de manipulation du vecteur codant les solutions qui n'a aucune relation avec la taille de l'image. Donc, la différence en temps de chiffrement des deux images est justifiée.

La valeur de la phéromone dépend de la valeur d'efficacité maximale et de l'efficacité de la solution calculée. Il est bien clair que pour de grandes valeurs d'efficacité, une amélioration de la quantité de phéromone sera marquée. Prenons l'exemple du premier message où pour la première version chiffrée nous avons obtenu 2166 / 9,06 pour efficacité / phéromone. Cependant, pour la deuxième version chiffrée, nous avons eu 2184 / 9,13. Donc, l'amélioration de l'efficacité est certainement accompagnée d'une amélioration de la quantité de phéromone. Il sera utile de rappeler, ici, que l'évaporation est la cause de diminution de la quantité de phéromone.

En mesurant le taux de différence entre les deux versions chiffrées de l'image Lena et l'image originale (l'image Lena), le tableau suivant récapitule les résultats obtenus en termes de NPCR, MAE et MSE. Il est clair que les deux versions chiffrées sont presque totalement différentes de l'image originale. Autrement dit, les pixels formant les versions chiffrées sont presque tous changés, soit de positions, soit de nombres d'occurrences et de positions ; ce qui donnera lieu à un très bon niveau de confusion.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.9207	1.08	0.73
Lena-version chiffrée 2	0.9082	1.03	0.69

TableauIV.3. *Niveaux de confusion de AntCrypt.*

Maintenant comparant AntCrypt avec le plus rapide de nos trois algorithmes développés et présentés dans le précédent chapitre, bien sûre sur le plan de temps de calcul. Le tableau suivant rappelle les temps de convergence des deux algorithmes OEEA et AntCrypt. Ce dernier a montré un temps de calcul meilleur que celui de OEEA que ça soit pour chiffrer des données textes ou images malgré qu'aucune explication exacte ne peut être donnée du fait que les deux algorithmes utilisent un même mode de chiffrement (chiffrement à base d'occurrences), donc un même codage qui est exploité par les composants de la métaheuristique impliquée (opérateurs génétiques et mécanismes de la sélection naturelle pour OEEA et fourmis et mécanismes de manipulation de phéromone pour AntCrypt). Et vu que le temps de convergence de notre algorithme de chiffrement basé sur les mécanismes évolutionnaires (OEEA) est légèrement plus grand que celui basé sur les fourmis (AntCrypt), donc possible que les premiers mécanismes consomment plus de temps de calcul que les deuxièmes, mais aucune preuve n'est connue !

	Temps calcul (s)	
	Texte 1	Lena
AntCrypt	5.825	2.235
OEEA	6.54	3.37

Tableau IV.4. *Temps de calcul de AntCrypt en comparaison avec le temps de calcul de OEEA.*

Remarque :

Dans le tableau IV.4, les temps de calcul de l'algorithme AntCrypt pour les deux cas de données (Texte 1 et Lena), représentent la moyenne des temps de calcul des deux versions chiffrées (Voir tableau IV.2).

De même et vu que la méthode de chiffrement présentée dans ce chapitre (Cryptage à base de fourmis) et celle présentée dans le précédent chapitre (Cryptage évolutionnaire) appartiennent à une même catégorie mère qui est celle de métaheuristiques, donc toute les deux vont bénéficier de quelques caractéristiques cryptanalytiques en commun. Il s'agit de l'aspect non déterministe démontré à travers les résultats présentés ci-dessus où deux versions chiffrées différemment correspondent toutes les deux à une même donnée originale (voir figures IV.4, IV.5, IV.6 et IV.7), et d'une robustesse contre les types d'attaques les plus avancées (attaque exhaustive, attaque statistique, attaque fréquentielle et attaque différentielle) comme c'était démontré dans la section III.5 du chapitre précédent.

IV.5. Conclusion

ACO est une méthode stochastique qui requiert de plus en plus l'attention de la communauté scientifique. Cette métaheuristique a prouvé sa performance pour plusieurs problèmes d'optimisation combinatoire. En effet, l'utilisation des traces de phéromone permet d'exploiter l'expérience de recherche acquise par les fourmis et ainsi renforcer l'apprentissage pour la construction de solutions dans les itérations futures de l'algorithme. En même temps, l'information heuristique peut guider les fourmis vers les zones prometteuses de l'espace de recherche.

Ainsi et dans ce chapitre, nous avons présenté, interprété et discuté les résultats expérimentaux obtenus par l'application de notre algorithme de chiffrement proposé *AntCrypt* sur des données modèles texte et images. Nous sommes arrivés à fixer nos paramètres par expériences, en choisissant 40 générations et 12 fourmis en essayant de réaliser un compromis entre l'efficacité et le temps de chiffrement.

L'algorithme, ainsi développé et qui a été nommé *AntCrypt*, a été comparé avec nos algorithmes développés à base d'algorithmes évolutionnaires où il a montré de bonnes caractéristiques telles qu'un bon niveau confusionnel, un aspect non déterministe, une taille sécurisée de clés, lui offrant une bonne résistibilité contre les attaques les plus avancées. En plus le temps de convergence de cet algorithme est meilleur que celui du plus rapide de nos trois algorithmes proposés dans le précédent chapitre.

CHAPITRE V

CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

V.1. Introduction

Nous nous sommes intéressés, dans les deux chapitres précédents, à la résolution du problème de cryptage de données texte et images par application de métaheuristiques à populations. Les approches que nous avons proposé sont caractérisées par leur qualité de confusion, leur temps d'exécution réduits pour certains d'eux (OEEA et AntCrypt) et leur robustesse face aux attaques avancées.

Dans ce chapitre, nous nous intéressons à l'exploitation d'une métaheuristique appartenant à une autre catégorie, les métaheuristiques à trajectoire. Il s'agit de la recherche tabou. Cette dernière s'appuie sur une recherche locale combinée à un mécanisme de prévention des cycles, grâce à un système de mémoire des mouvements précédemment appliqués ou des configurations visitées (la liste tabou).

De manière générale, une recherche locale démarre d'une solution initiale possible et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant. Un voisinage d'une certaine solution correspond à des éléments adjacents à cette solution dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration dans la solution courante ne puisse être faite.

Nous présentons, après la description de l'approche proposée, les résultats numériques obtenus afin de les comparer aux résultats obtenus par nos autres méthodes proposées (cryptage évolutionnaire et cryptage par colonies de fourmis) et certaines autres méthodes de la littérature.

V.2. Motivation

La recherche Tabou est une métaheuristique basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

L'originalité de la méthode de recherche tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objective f et l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum

quelconque par acceptation de la dégradation de la fonction objective, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste d'une certaine longueur, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

V.3. Algorithme proposé

À partir d'une solution initiale, le principe général de la recherche tabou est le suivant (voir Algorithme V.1). À chaque itération de la recherche, une partie ou l'ensemble des voisins de la solution est exploré, et un des mouvements parmi ceux de coût minimal est sélectionné. Ce mouvement est appliqué quel que soit son coût, à la condition de ne pas créer un cycle dans le processus de recherche locale à court terme (en menant à une configuration dont les caractéristiques sont stockées dans la liste tabou). Une exception est faite (critère d'aspiration) lorsque le mouvement permet d'atteindre une solution de meilleure qualité que la meilleure solution enregistrée à ce stade. La liste tabou est mise à jour à chaque itération de la recherche en fonction des mouvements choisis. Ce mécanisme permet de sortir des minima locaux en acceptant des mouvements détériorants et en empêchant parallèlement le retour immédiat à la solution de minimum local qui vient d'être quittée.

Algorithme V.1 *Schéma général d'un algorithme tabou*

```
Engendrer une configuration initiale s
s* ← s
T ← ∅ liste tabou
tant que condition d'arrêt non satisfaite faire
    m ← meilleur mouvement (i.e. minimisant f) parmi ceux non tabou
    ou ceux vérifiant un critère d'aspiration
    Modifier s en effectuant le mouvement m
    Mettre T à jour
    si f(s) < f(s*) alors
        s* ← s
    fin
fin
retourner s*
```

La succession des étapes suivantes montre plus de détails du schéma du mécanisme utilisé pour la construction d'une solution au problème étudié :

1) Détermination d'une solution initiale $S_{(0)}$ codant l'image originale.
2) Création puis la mise à jour d'une liste tabou qui mémorisera les déplacements effectués.
3) Choix aléatoire d'un nombre d'emplacements ou d'éléments de la solution courante ($i^{ème}$ solution) à déplacer.
4) Génération aléatoire d'un ensemble de voisins pour chacun des éléments choisis dans l'étape précédente.
5) Choix du meilleur voisin de l'ensemble généré.
6) Calcul tabou de la solution suivante $S_{(i+1)}$.
7) Création et la mise à jour d'une table de hachage qui comportera les dernières meilleures solutions obtenues.
8) Évaluation de la solution calculée $S_{(i+1)}$ obtenue à chaque itération.
9) Validation de la solution calculée $S_{(i+1)}$.
10) Vérification du critère d'arrêt de l'algorithme.

En ce qui suit, nous décrivons les principales étapes de l'algorithme de chiffrement tabou proposé et que nous l'avons appelé *TabuCrypt*.

V.3.1. Création de la solution initiale

La solution initiale exploitée par *TabuCrypt* s'obtient en codant les données originales suivant le même mode de codage adopté par *OEEA* ou *AntCrypt*. C'est d'ailleurs le même codage utilisé pour modéliser toutes les autres solutions.

V.3.2. Choix des éléments à déplacer

Les solutions sont représentées soit, par un vecteur englobant 1393 éléments pour le cas de manipulation de données texte soit, par un autre regroupant 768 éléments pour le cas de manipulation de données images. *TabuCrypt* cherche à réarranger aléatoirement le vecteur correspondant à la solution initiale codant la donnée originale en permutant les éléments entre eux. Toutefois, la taille du vecteur est assez grande (1393 éléments ou même 768 éléments), c'est pourquoi la manipulation un par un de ses éléments sur l'ensemble des itérations de l'algorithme nécessitera un grand temps de calcul. Ainsi, sauf un sous ensemble du vecteur globale sera manipuler pour déplacement. Cela permet, d'un coté, d'optimiser le temps de calcul et, d'un autre coté, de converger petit à petit vers la solution finale en évitant, ainsi, le problème de convergence prématurée.

V.3.3. Génération de voisinage

Le voisinage est le responsable sur le fait de pousser l'algorithme à l'amélioration de la qualité des résultats. Dans notre cas, il est généré de la manière suivante :

Pour chacun des éléments du sous-ensemble construit dans l'étape précédente, nous générons aléatoirement un ensemble de voisins candidats au déplacement avec l'élément en question. Lors de l'élection d'un voisin, les conditions suivantes doivent être vérifiées :

- ✓ Un voisin ne doit pas être élu plus qu'une fois, c'est-à-dire, tous les voisins de l'ensemble de voisins sont différents les uns des autres.
- ✓ Les voisins élus n'appartiennent pas dans la liste tabou.

Une fois l'ensemble des voisins construit, nous choisissons celui qui diffère le plus de l'élément à déplacer. Formellement, soit E_V l'ensemble de n voisins V, et E_i l'élément candidat au déplacement. La sélection du voisin v avec lequel l'élément E_i sera permuté se fait comme suit :

$$v = V_j \,/\, j = \overline{1,n} : Max\left(\left|E_i - V_j\right|\right) \qquad \text{(V.1)}$$

V.3.4. Mise à jour de la liste tabou

Dans notre cas et à une itération donnée, la liste tabou regroupe les voisins qui ont participé à des déplacements au cours des itérations précédentes avec l'un des éléments du sous-ensemble construit dans l'étape 2 décrite dans la section V.3.2. Ainsi, la mise à jour de cette liste (liste tabou) se fait à chaque fois qu'un élément sera permuté avec un voisin choisi suivant l'étape 3 décrite dans la section V.3.3 en ajoutant ce dernier à la liste des éléments tabous. A la fin de chaque itération, le contenu de la liste tabou sera effacé.

La politique utilisée pour rendre tabous certains éléments évite la modification des nombres d'occurrences originaux, puisque le processus de calcul de la solution ne doit pas modifier les éléments du vecteur initial mais plutôt il cherche à changer leur dispersion.

V.3.5. Calcul de solutions

Au cours d'une itération donnée, le calcul de la solution proposée à cette itération se base sur la solution calculée à l'itération précédente. En effet, des éléments du vecteur présentant la solution à l'itération précédente échangeront leurs positions avec celles des voisins désignés pour chacun d'eux. Cette nouvelle dispersion des éléments forme le nouveau vecteur solution. Il est à noter que les éléments non élus pour un déplacement garderont leurs positions sans modification.

V.3.6. Mise à jour de la table de hachage et évaluation des solutions

La table de hachage sert à mémoriser les différents points visités de l'espace de recherche en gardant les différentes solutions calculées sur l'ensemble des itérations. Le but sera d'interdire la recherche à revenir vers des points déjà visités à travers les précédentes itérations. Ainsi, une solution générée à la fin d'une certaine itération sera comparée au contenu de la table de hachage avant qu'elle soit soumise à une validation finale en vu de l'accepter comme intéressante solution à partir de laquelle la recherche peut continuer sur l'ensemble des itérations restantes. Si cette solution représente un nouveau point de l'espace non encore visité, elle sera ajoutée à la table de hachage pour quelle fera l'objet des futurs tests. Cette politique sert à échapper aux minima locaux et à favoriser l'exploration de nouveaux points de l'espace de solutions possibles.

La validation définitive d'une solution S_i calculée à une itération i donnée, consiste tout d'abord, à l'évaluer. Dans notre cas, l'évaluation se fait suivant la fonction d'évaluation illustrée par la formule suivante, notant que S_0 soit la solution initiale (voir étape 1).

$$F(S_i) = \sum_{j=1}^{l} \left| S_{i_j} - S_{0_j} \right| \qquad \text{(V.2)}$$

Où l : représente le nombre de valeurs possibles que peut prendre un élément d'une donnée (égale à 1393 pour le cas de données texte et 768 pour le cas de données images).

Une fois l'évaluation faite, la qualité de la solution en question sera comparée à celle de la solution calculée à l'itération précédente. Si elle est meilleure que la précédente, la recherche à travers la prochaine itération va se continuer à partir du point représenté par cette solution, sinon elle sera rejetée et la recherche continuera à partir de la solution calculée à l'itération précédente.

Ces points peuvent être résumés comme suit : à chaque itération la solution calculée est ajoutée à la table de hachage puis elle sera examinée selon les trois critères suivants :

1) Si la solution S_i obtenue n'est pas une nouvelle solution alors elle sera rejetée et les étapes de 2 à 6 sont à refaire.

2) Si la solution S_i obtenue est une nouvelle solution mais plus mauvaise que celle de l'itération *i-1* (S_{i-1}), de même elle est rejetée, mais sauvegardée dans la table de hachage, et les étapes de 2 à 6 sont à refaire.

3) Si la solution S_i obtenue est une nouvelle solution mais, cette fois ci, est meilleure que celle de l'itération (i-1), alors elle sera retenue pour continuer la recherche à partir de la prochaine itération et, ainsi, elle servira à calculer la solution suivante S_{i+1}. En même temps, elle sera mémorisée dans la table de hachage.

Remarque :
Dans notre algorithme, on considère comme *mouvement global* le passage d'une solution courante S_i vers une solution suivante S_{i+1}, en respectant les conditions citées précédemment. Toutefois, les *mouvements élémentaires* sont les déplacements des éléments et leurs voisins vers leurs positions mutuelles. Ainsi, un mouvement global émerge comme résultat des mouvements élémentaires.

V.3.7. Critère d'arrêt

Pour toute recherche itérative un critère d'arrêt est obligatoire pour éviter le problème de boucles infinies. Ce dernier qui doit être fixé à l'avance permet de déterminer le nombre de fois que la tâche à exécuter sera répétée.

Pour notre algorithme, nous avons choisi de fixer, expérimentalement, le nombre d'itérations. Ce choix influe directement sur longueur du temps de calcul et de la qualité de la solution construite.

V.3.8. Mécanismes avancés en recherche tabou

V.3.8.1. Intensification / Exploitation

L'intensification consiste à approfondir la recherche dans certaines régions de l'espace, identifiées comme susceptibles de contenir un optimum global.
Pour notre cas, les mouvements globaux en plus des mouvements élémentaires serviront à mieux intensifier la recherche. En effet, les mouvements élémentaires traduisent les déplacements des éléments d'un vecteur solution vers les positions de leurs voisins et vise versa. Ces voisins sont les éléments optimaux (les meilleurs voisins) de l'ensemble des voisins candidats au déplacement (voir étape 3). Donc, nous cherchons à exploiter les qualités des éléments. De même, les mouvements globaux cherchent à exploiter les qualités des solutions (les vecteurs d'éléments) en n'acceptant de poursuivre la recherche qu'à partir des solutions améliorantes lors du passage d'une itération à l'itération qui suit.

V.3.8.2. Diversification / Exploration

La diversification vise à utiliser des mouvements encore jamais réalisés afin d'explorer de nouvelles régions de l'espace de recherche et des régions éloignées du voisinage actuel. Dans notre cas, cette caractéristique est accomplie par la sélection aléatoire d'un nouveau voisinage non tabou pour chaque élément.

V.3.8.3. Critère d'aspiration

Le critère d'aspiration autorise un mouvement tabou sous certaines conditions. Il consiste à tester si la solution produite de statut tabou présente un coût inférieur ou de qualité meilleure que ceux de la meilleure solution trouvée jusqu'à présent. Si c'est le cas, le statut tabou de la solution est levé.

Dans notre cas, la notion de tabou est exploitée lors de la construction d'une certaine solution en interdisant les mouvements élémentaires vers les éléments voisins avec lesquels les éléments du sous-ensemble construit comme l'indique l'étape 2 ont échangé de positions. Ainsi, le mécanisme d'aspiration n'a pas de place dans notre méthode du fait que toute libération d'un élément de la liste tabou entraîne une modification des éléments des vecteurs et, par conséquent la modification des caractéristiques servira à calculer ultérieurement la donnée déchiffrée, tandis que notre but est, plutôt, modifier la répartition des éléments des vecteurs.

V.3.9. Déchiffrement

L'opération inverse au chiffrement est le déchiffrement qui permet de regénérer la donnée originale précédemment chiffrée par notre algorithme de chiffrement tabou *TabuCrypt*. Pour ce faire, ce processus exploite une information secrète calculée lors du chiffrement à coté de l'image chiffrée. Il s'agit d'une clé de session secrète. Elle représente les permutations des positions des l éléments (1393 éléments ou 768 éléments) du vecteur codant une certaine donnée à travers les différentes itérations. Une fois elle parviendra sans modification au destinataire approprié, elle servira à calculer la donnée originale.

V.3.10. Réglage des paramètres et résultats

Le schéma général de l'algorithme finalement implémentant les étapes de l'approche de cryptage tabou proposé est le suivant :

Algorithme V.2 *TabuCrypt*

Étape 1 : Générer une solution initiale de gain total G_0.
Étape 2 : Initialiser
- le nombre d'itération iter (à 1),
- la table de Hachage H (ajouter la solution initiale à H),
- la liste tabou T (vide).

Répéter les étapes de 3 à 11 jusqu'à iter = iterMax.
Étape 3 : Initialiser le compteur d'éléments (composants d'une solution)
Répéter les étapes de 4 à 6 jusqu'à
Étape 4 : Sélectionner aléatoirement un ensemble de voisins des éléments non tabous, puis choisir le meilleur de l'ensemble.
Étape 5 : Déclarer comme tabou le voisin choisi (voisin ajouté à T).
Étape 6 : Déplacer l'élément courant et le voisin choisi vers leurs positions mutuelles.
Étape 7 : Evaluer la nouvelle solution calculée pour mesurer son gain G_{iter}.
Étape 8 : Si : la solution calculée est déjà contenue dans H *alors :* Solution ignorée, *Sinon :* Ajouter la solution à H
Étape 9 : Si $G_{iter} > G_{iter-1}$ *alors* continuer le calcul à partir de $Solution_i$ *Sinon* Continuer le calcul à partir de la $solution_{iter-1}$
Étape 10 : Vider T.
Étape 11 : iter + +.

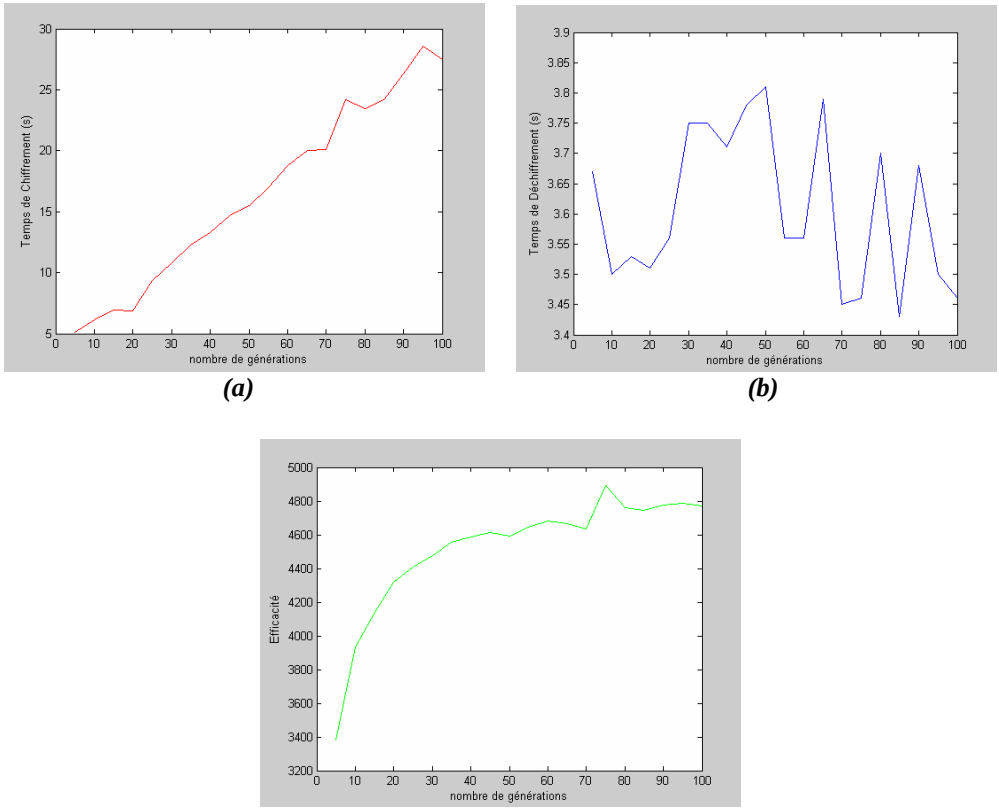
Nous précisons, dans cette section, les valeurs des paramètres de *TabuCrypt* ainsi que les résultats de son application sur les données modèles précédemment présentées (Texte1, Texte2, Image Lena et image Logo).

V.3.10.1. Réglage des paramètres

Les principaux paramètres de *TabuCrypt* concernent le nombre de générations ou d'itérations de l'algorithme et le nombre de voisins seront présentés.

Après *IterMax* itérations, fixée expérimentalement, la procédure de recherche tabou s'arrête et fournit la meilleure solution trouvée. La figure V.1 récapitule la moyenne des différentes valeurs de test et leurs impacts sur le temps de convergence et la qualité de la solution trouvée représentant une version chiffrée de l'image de test Lena.

D'après les résultats résumés dans la figure, ci-dessous, nous constatons que la meilleure efficacité a été obtenue pour un nombre de générations égale à 75 après un temps de chiffrement de 24.20 s et un temps de déchiffrement de 3.46 s. De même, les nombres de générations 80, 85, 90, 95 et 100 ont donné de bonnes valeurs d'efficacité (4760, 4744, 4777, 4789 et 4771, respectivement), mais leurs temps de chiffrement sont assez longs (23.46 s, 24.18 s, 26.29 s, 28.54 s et 27.48 s, respectivement).



(a) ***(b)*** ***(c)***

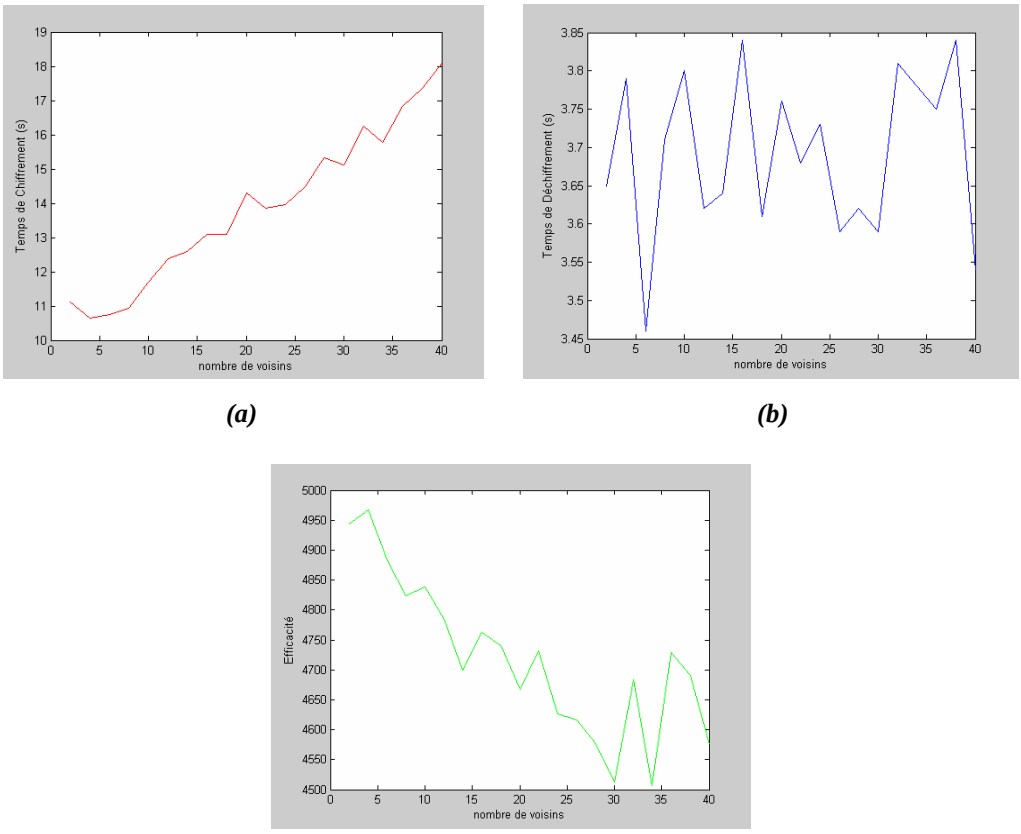
Figure V.1. *Résultats obtenus suivant les différentes valeurs de test de nombre d'itérations (générations) :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Toutefois, les résultats obtenus, pour le test avec 45 générations, sont acceptables que se soit sur le plan efficacité (4612) ou temps de chiffrement (14.70 s). D'autres valeurs de nombre de générations (55, 60, 65 et 70) ont donné des résultats d'efficacité de même ordre que celle du nombre de générations 45, mais après des temps de chiffrement relativement plus longs (29.93, 28.76, 20.04 et 20.12, respectivement). Ainsi, la valeur de *IterMax* a été fixée à 45 générations.

L'autre paramètre à régler est le nombre de voisin. Pour ce faire, nous avons utilisé le même mécanisme que celui utilisé pour régler le paramètre relatif au nombre d'itérations. Les résultats des différents tests sont résumés à travers la figure V.2 en donnant l'influence des différents nombres de voisins testés sur le temps de convergence et l'efficacité de la solution trouvée.

D'après le contenu des figures V.1 et V.2, la meilleure efficacité (4967) a été obtenue avec un nombre de voisins qui vaut 4 après un temps de chiffrement assez raisonnable et représentant le plus petit temps de calcul sur l'ensemble des tests effectués (10.64s).

Ainsi, nous avons opté à régler le paramètre relatif au nombre de voisins en lui attribuant la valeur 4.



(a) ***(b)***

(c)

Figure V.2. *Résultats obtenus suivant les différentes valeurs de test de nombre de voisins :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Le tableau suivant (tableau V.1) résumé les valeurs de paramètres adoptés par *TabuCrypt.*

	Valeurs des paramètres de*TabuCrypt*
Nombre de voisins	4
Nombre de générations	45

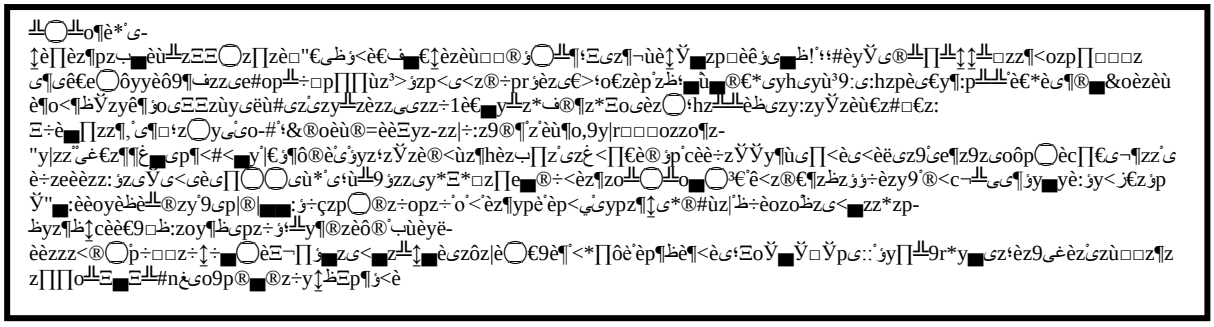
Tableau V.1. *Valeurs adoptées pour les paramètres de TabuCrypt.*

V.3.10.2. Résultats

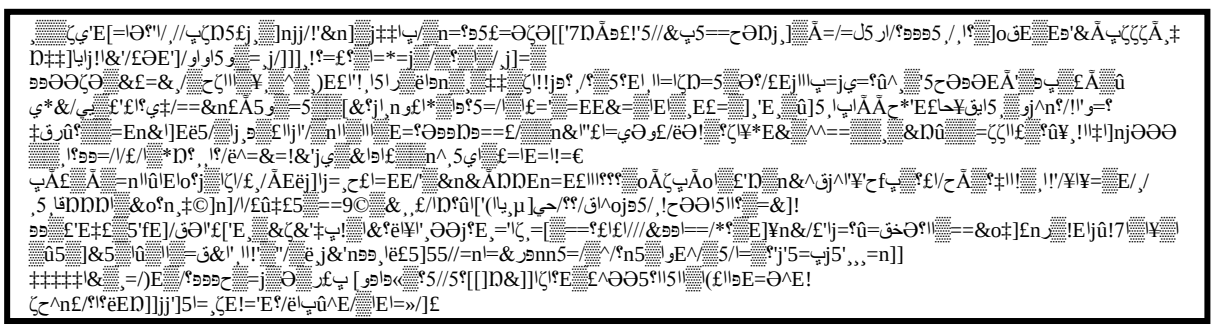
A ce niveau nous présentons les résultats de chiffrement par *TabuCrypt* des données tests. Deux versions chiffrées de chacune des données tests sont présentées accompagnées des clés générées dans chaque cas. Le tableau V.2 résume les temps de chiffrement et de déchiffrement ainsi que l'efficacité de chacun des exemples.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure V.3. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets)

728	898	353	668	901	875	906	437	425	40	739	524
797	890	877	786	357	654	889	258	566	884	883	899
399	721	896	133	905	483	410	895	735	117	900	876
878	303	479	907	706	879	100	813	91	453	152	634
874	186	597	902	480	341	99	42	736	438	456	893
794	344	881	267	887	474	617	191	885	56	892	886
897	377	601	904	34	98	672	880	903	673	496	760
373	882	894	888	347	463	750	891	520	248	465	571
624	116	573	744	866	646	926	68	1228	131	311	532
299	518	139	1134	1120	734	714	290	22	97	541	1047
1234	961	1110	819	52	153	255	1006	560	960	1077	942
1109	154	836	30	412	1119	1202	1037	1003	383	1238	335
392	1035	1241	119	448	409	1115	266	1036	857	971	194
713	912	422	378	500	782	274	1048	1102	651	1129	1091
810	696	1040	994	321	362	46	563	1148	149	339	858
503	328	110	770	1053	773	272	598	77	1087	1187	1121
551	418	225	801	1185	1082	955	557	737	506	464	1070
1240	168	187	843	796	82	867	450	200	764	190	982
1076	967	873	260	695	224	1018	207	1049	277	3	652
792	1014	1213	936	466	621	829	2	400	1085	38	23
1159	570	509	816	973	164	205	1201	112	1078	870	333
364	1212	350	250	102	536	172	629	705	178	381	722
283	1100	823	426	84	122	134	1128	591	361	798	24

1224	1111	179	1143	526	1219	1166	176	834	10	1221	1093
1062	1232	167	587	647	342	490	1041	1114	785	525	1020
473	1233	856	562	423	579	527	427	1194	215	653	242
1237	767	795	1153	862	952	998	644	1081	461	182	232
29	539	1025	1223	434	411	320	932	1204	499	11	546
1112	558	958	1160	486	467	1039	436	1245	1151	421	920
126	692	174	802	581	1066	852	355	1023	180	125	821
1104	129	265	1094	158	1101	824	871	555	1156	106	681
1145	645	803	1208	114	1068	1080	1222	269	996	1074	263
1064	1075	1184	213	699	123	312	1056	1130	175	468	754
380	1063	616	234	1200	768	865	1154	108	1069	631	698
360	1207	397	356	711	540	484	1180	1215	941	1220	596
604	1244	774	916	1050	749	420	169	1029	1242	424	376
1107	1178	820	352	372	118	443	818	701	1136	690	950
1196	93	724	543	1155	476	1034	963	196	978	1086	472
1230	781	610	273	1125	1137	531	120	554	264	969	763
165	407	390	1209	583	49	648	183	331	104	985	130
1126	1152	977	43	917	508	614	992	428	1067	582	983
987	454	497	230	530	285	374	1097	758	308	354	495
664	128	825	441	990	288	951	1131	1008	1004	632	121
752	249	105	682	1195	460	1162	981	746	297	228	700
1116	1095	296	83	1164	1198	528	502	27	762	769	1139
655	863	514	1028	276	811	1177	663	432	12	1002	716
826	96	59	458	1225	1181	136	416	830	1072	937	209
94	986	784	940	146	658	204	976	945	918	185	1012
643	611	1169	1142	1079	1173	793	457	1191	393	212	635
239	984	261	231	1010	1186	1203	576	1124	1140	404	738
991	33	1163	358	586	1098	324	968	1083	642	972	76
1013	1182	1092	751	16	88	817	957	569	135	1117	535
622	670	1009	291	832	329	28	979	72	236	1144	462
298	79	626	772	221	238	487	559	1236	556	946	633
702	69	235	1005	109	140	319	452	1054	155	1019	304
485	756	512	74	398	216	1210	925	1171	911	1243	842
997	478	1239	293	921	429	160	933	1088	431	145	1229
661	295	322	35	1055	1043	1016	201	1150	637	620	482
369	790	253	640	719	491	1061	1042	237	113	914	1122
64	348	171	727	233	254	522	206	1174	81	115	511
1193	928	1218	346	931	405	1	613	1214	394	709	592
359	307	553	708	850	163	999	166	39	1011	370	949
710	1113	1157	935	1022	860	147	78	948	198	1211	177
330	19	771	211	684	993	677	980	1206	927	723	804
1065	970	1044	1099	729	523	442	908	1217	1165	159	9
828	995	150	244	1231	15	325	137	1189	1205	844	691
922	964	564	89	281	243	671	218	657	1000	1172	939
1059	1032	683	649	783	989	757	606	385	965	470	954
389	195	959	806	1060	615	676	630	219	1170	1052	294
257	864	869	371	715	301	840	318	1118	1158	268	1027
627	435	1103	47	1015	1175	975	1192	313	1084	1132	529
67	787	953	1167	1135	924	919	1073	1031	1058	853	271
384	471	812	947	18	1161	765	567	934	217	66	759
545	475	8	1147	505	1001	1045	599	1141	1235	1176	584
284	1216	929	704	1089	943	501	717	445	966	693	605
259	1168	733	743	747	1138	550	336	343	53	814	1133
144	913	439	1051	872	1127	909	1024	459	915	17	504
944	1096	1046	73	726	1105	1026	956	1188	718	549	1146
48	910	494	189	222	1017	827	334	612	585	745	678
208	580	974	938	1123	396	1190	1030	367	55	337	366
962	246	21	4	777	1179	1149	1227	220	930	1226	1021
800	659	1183	731	202	1197	403	1090	988	533	379	1057
1106	1071	515	282	451	1038	574	1033	1199	226	1108	923
1007	280	111	544	406	61	748	1274	402	753	181	1374
143	306	868	1264	6	193	489	618	1351	667	1346	1375
1267	32	732	687	286	688	507	363	1308	859	95	575
855	1376	345	1358	65	1361	1257	565	44	1371	214	703
447	36	561	1329	799	203	625	720	278	1287	845	210
444	1280	1327	1303	414	1268	1273	846	477	446	1253	578
251	1377	822	229	1328	1295	847	419	1320	837	1378	831
1373	1362	538	1379	662	1254	1334	778	1331	58	45	602
87	287	854	839	1292	780	1322	309	619	401	279	588
849	1350	332	791	351	156	1247	1355	639	157	132	1380
1282	1270	1381	641	375	730	1300	498	388	594	519	1382
689	1313	1258	542	595	162	1383	493	275	252	5	1367
1348	368	776	227	1319	675	413	1338	20	1318	188	338
516	548	568	590	1294	415	1343	609	417	1356	54	1315
223	1323	1316	1357	685	1246	1384	433	41	815	1360	1296
755	1345	316	808	650	310	449	1284	623	1347	1302	101

1364	481	1293	142	170	517	680	1314	1324	1281	1385	317
1321	572	833	1261	141	62	697	305	1330	51	173	1312
669	148	1256	656	92	391	161	382	1251	600	665	805
686	1342	589	1369	534	349	1249	838	1332	1336	521	1272
387	1262	1340	1301	70	552	192	1386	1311	603	1277	766
440	789	638	292	37	1279	1291	1372	725	1387	247	851
1299	86	1388	674	848	1310	1359	57	138	1286	124	1285
1266	1248	809	1252	340	289	1389	775	488	71	694	607
31	1349	628	1263	1390	1255	365	1341	300	1275	262	1309
1250	761	25	50	85	408	256	577	835	841	742	1370
707	712	666	1283	1354	455	1317	386	430	1269	492	1339
14	660	1288	314	1368	1306	151	1353	807	90	1265	788
608	510	1391	1335	327	1304	741	7	1365	197	245	861
1297	1259	80	679	1333	75	1326	1352	779	26	199	270
547	537	636	1289	740	323	326	395	1298	60	1344	302
1392	1337	1271	593	127	63	184	13	513	103	1307	469
1290	1393	1260	1366	1305	1325	1278	1276	107	241	315	240
1363											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets)

7	36	33	79	39	24	81	16	5	73	3	1
20	88	30	44	56	41	11	62	65	47	98	26
814	806	645	389	313	988	737	838	346	946	243	890
300	123	778	333	207	229	663	571	127	816	821	355
1220	1344	1025	1270	1147	1345	1020	1218	1338	1037	1085	1041
1016	1226	1114	1073	1346	1190	1195	1253	1227	90	40	70
15	95	6	53	87	34	91	25	82	76	45	13
93	14	55	28	37	17	72	66	85	61	22	64
21	51	8	77	35	78	86	38	10	42	18	99
31	60	2	27	54	92	94	80	19	71	69	29
67	46	84	89	12	9	96	4	57	58	63	49
59	75	50	23	83	48	526	275	178	752	965	810
453	272	436	898	463	245	321	586	113	506	208	419
708	183	640	451	376	195	372	414	942	895	238	302
443	631	474	712	434	917	277	392	187	467	607	557
845	647	688	188	52	97	630	680	865	449	583	164
262	105	634	317	639	119	928	464	714	802	797	932
823	822	953	684	529	694	270	991	482	154	956	978
440	135	197	670	280	351	987	248	881	779	933	327
606	214	662	695	292	456	909	608	158	285	491	877
311	771	395	891	938	803	563	612	561	963	715	501
447	153	498	134	792	220	215	811	874	427	283	418
538	68	32	43	74	687	345	661	924	284	486	359
839	495	115	893	732	103	307	442	305	675	600	136
157	603	106	417	122	918	441	785	930	982	181	466
255	758	765	794	817	271	854	597	494	900	219	198
484	315	194	218	923	382	667	309	901	408	120	558
204	772	336	761	470	266	519	996	240	582	795	969
168	142	541	559	540	853	564	578	936	916	788	882
274	861	527	365	651	367	707	665	162	435	469	929
589	979	939	290	753	755	776	584	656	269	391	416
860	700	786	514	344	566	492	319	993	139	384	790
256	475	252	437	244	145	276	544	374	598	373	828
258	652	870	685	784	318	801	927	353	109	180	324
689	858	910	108	940	508	525	137	749	697	975	296
489	650	503	329	101	570	326	850	716	581	246	627
403	782	375	556	985	949	126	907	422	179	282	221
161	944	981	604	517	349	212	783	138	906	835	873
594	545	880	655	462	967	539	457	203	528	717	952
341	316	366	813	591	836	299	769	926	951	547	904
848	844	840	800	744	576	819	995	412	413	147	588
812	424	356	404	554	869	619	186	390	150	692	825
740	852	343	601	831	431	312	989	535	481	377	702
764	286	855	682	976	261	515	768	750	254	696	883
654	169	304	618	889	727	710	805	362	799	360	621
543	530	288	428	379	118	587	531	423	833	426	210
342	241	905	787	572	439	560	954	748	653	402	773
394	980	448	649	223	488	760	415	368	959	611	913
287	493	569	729	253	767	132	322	536	479	553	476
648	410	815	964	562	458	323	217	385	830	144	234
173	617	643	143	660	628	461	551	473	193	337	257
644	698	673	293	104	872	757	334	843	826	510	452
289	107	573	213	251	459	516	512	110	998	438	242
974	971	314	471	669	550	620	465	148	580	593	777

192	990	155	230	614	504	638	983	915	775	842	149
236	363	759	999	140	745	851	885	736	766	941	294
871	867	994	152	914	407	396	555	690	886	585	896
369	613	731	966	249	774	460	170	505	331	970	721
846	807	892	295	577	298	804	961	191	796	948	159
590	791	674	117	742	720	738	490	405	664	599	184
432	166	509	762	672	986	615	542	306	165	130	622
455	728	683	445	820	658	957	605	177	279	176	335
859	129	364	706	202	879	809	502	348	383	657	477
281	947	832	671	857	868	546	595	112	268	233	724
741	228	713	878	908	167	370	887	925	100	325	704
200	211	499	681	626	574	677	763	433	701	624	446
637	354	111	960	841	977	125	483	444	478	972	711
131	808	430	133	387	380	163	468	911	310	897	864
160	513	227	141	945	518	352	718	984	524	487	425
116	781	411	278	636	876	128	320	532	330	629	725
834	454	250	114	291	668	520	534	849	770	185	358
623	973	709	409	303	146	579	259	955	429	175	747
397	950	659	679	239	856	642	610	609	568	301	958
496	297	533	121	937	480	201	730	552	818	500	225
400	182	521	328	894	922	935	522	686	888	189	931
199	511	754	406	124	232	102	332	920	361	398	693
224	386	399	592	350	739	567	703	265	226	919	263
899	222	472	746	156	635	378	507	206	171	273	827
401	866	565	548	338	789	421	992	209	151	676	641
485	884	699	260	393	523	902	388	726	997	968	190
381	751	633	231	264	756	824	357	420	625	450	743
196	216	602	371	237	962	174	235	735	875	837	172
863	829	666	247	719	912	793	339	847	943	646	734
934	347	798	340	205	575	733	267	537	903	678	308
616	862	596	632	549	691	705	723	497	921	780	722
1229	1311	1126	1074	1105	1179	1155	1223	1023	1161	1175	1129
1184	1134	1305	1319	1088	1082	1347	1079	1281	1152	1144	1149
1288	1176	1186	1289	1254	1283	1348	1017	1159	1189	1349	1217
1131	1039	1350	1272	1064	1192	1188	1228	1115	1095	1264	1310
1058	1201	1198	1022	1014	1043	1108	1071	1351	1352	1027	1165
1151	1059	1028	1353	1141	1200	1143	1354	1247	1100	1029	1112
1005	1355	1257	1083	1194	1046	1356	1035	1024	1307	1101	1091
1278	1287	1092	1139	1120	1250	1007	1246	1225	1116	1051	1357
1244	1086	1209	1096	1090	1284	1358	1106	1127	1318	1309	1290
1193	1216	1076	1052	1259	1359	1117	1360	1361	1239	1047	1034
1003	1057	1277	1111	1093	1332	1301	1009	1183	1328	1178	1061
1099	1265	1215	1240	1362	1213	1269	1298	1049	1312	1363	1171
1263	1252	1364	1197	1077	1243	1365	1366	1145	1214	1142	1018
1315	1123	1146	1107	1060	1367	1113	1006	1160	1337	1249	1326
1011	1279	1368	1208	1196	1275	1056	1170	1110	1234	1335	1261
1294	1069	1031	1203	1063	1267	1258	1300	1202	1157	1241	1030
1150	1280	1206	1369	1181	1180	1021	1370	1158	1121	1097	1205
1128	1012	1293	1089	1341	1237	1187	1317	1102	1010	1068	1304
1371	1291	1372	1148	1153	1065	1256	1303	1306	1124	1295	1053
1013	1162	1268	1044	1373	1078	1032	1166	1177	1137	1026	1255
1374	1375	1133	1236	1191	1172	1048	1245	1282	1119	1296	1062
1130	1308	1376	1207	1299	1036	1377	1333	1136	1199	1378	1379
1292	1164	1045	1248	1343	1380	1314	1235	1321	1066	1232	1182
1271	1336	1320	1260	1251	1381	1382	1168	1286	1383	1384	1316
1322	1385	1042	1262	1173	1212	1038	1386	1040	1274	1154	1327
1273	1185	1387	1313	1210	1388	1019	1132	1033	1001	1156	1389
1122	1070	1098	1211	1055	1329	1323	1285	1072	1054	1325	1004
1302	1118	1231	1219	1204	1087	1222	1221	1125	1080	1342	1224
1015	1340	1390	1050	1330	1233	1094	1081	1067	1104	1174	1297
1391	1331	1075	1392	1339	1238	1230	1135	1324	1276	1163	1334
1103	1084	1002	1138	1167	1109	1242	1140	1008	1266	1169	1000
1393											

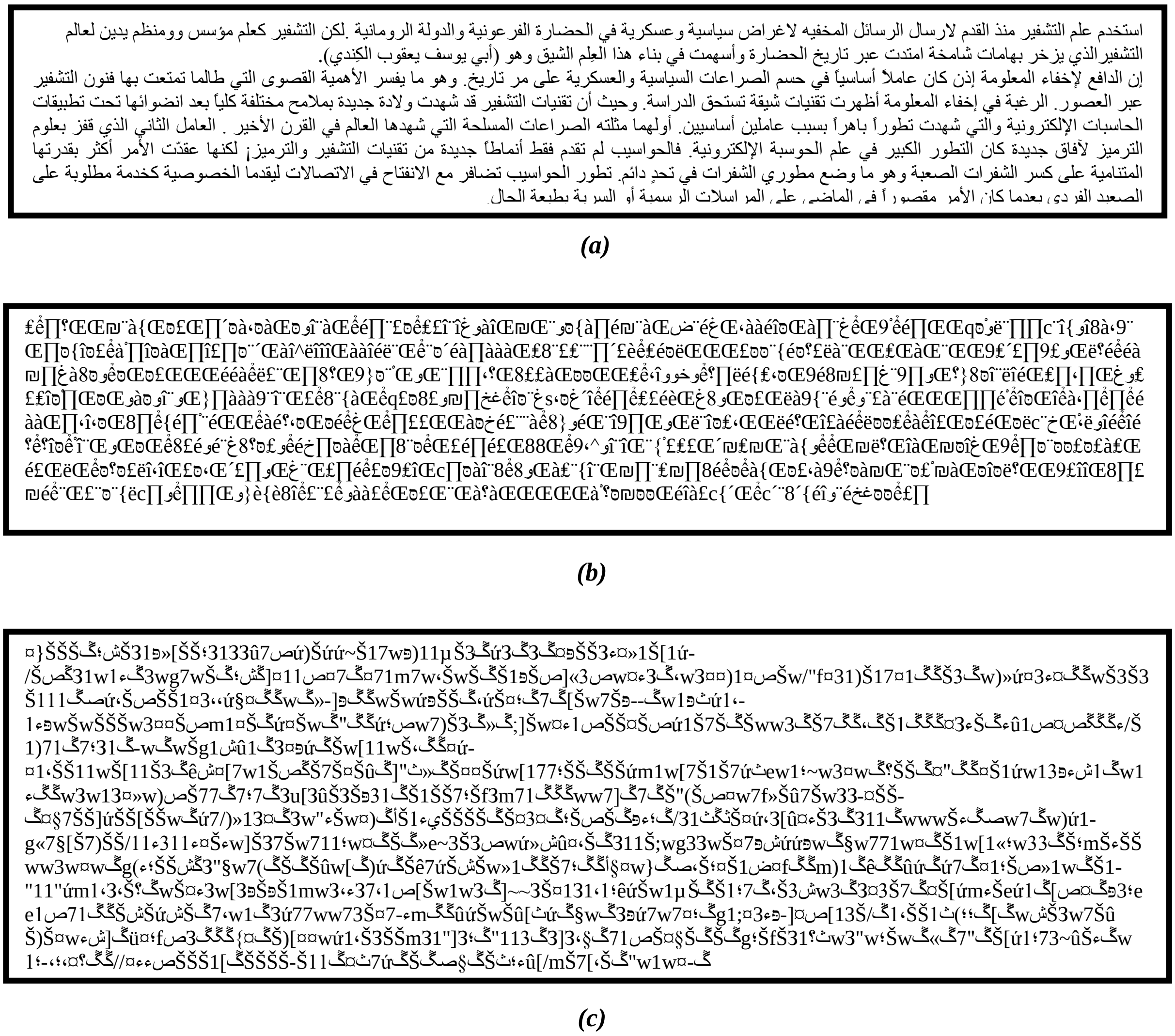


***Figure V.4. (a)** Donnée originale Texte2, **(b)** Première version chiffrée, **(c)** Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets)

8	2	4	7	6	9	1	5	811	910	899	896
878	885	622	892	767	914	357	901	895	297	771	211
70	851	189	890	897	694	880	913	662	78	908	900
906	210	170	889	879	904	833	407	208	300	498	306
259	891	871	858	469	698	543	87	903	368	52	884
266	835	642	886	92	909	905	882	883	405	527	371
915	658	12	893	113	172	457	898	888	633	887	242
877	881	414	376	911	556	912	907	902	163	648	281
894	876	949	701	174	1155	643	192	649	213	466	1103
514	1076	1022	660	624	3	632	430	623	944	925	770
785	796	67	1194	230	707	799	1177	1218	245	508	1007
179	484	1209	1223	549	1118	1037	1167	740	73	1160	248
193	1088	287	1200	304	131	175	544	511	428	1173	176
1176	931	983	815	951	370	32	955	826	53	945	754
168	863	1009	1219	692	532	459	273	1018	356	730	255
636	604	539	11	984	79	930	817	383	581	829	1148
1133	764	961	1068	1127	1140	1225	630	956	403	1143	1235
822	601	755	1217	535	1073	960	1137	203	325	545	1036
745	1112	971	1097	1168	1232	223	437	805	1062	1204	321
272	374	1080	260	748	86	1045	434	429	1157	675	994
145	1142	587	585	1229	1069	1044	267	386	471	1163	147
1102	1182	747	593	798	1054	1019	82	981	978	348	1066
928	669	1192	1065	1038	311	1196	943	1141	597	610	63
741	1085	1075	234	1063	962	850	682	1156	401	843	670
81	1058	84	1000	151	525	60	1110	225	501	969	552
341	446	486	985	1093	580	372	1187	998	517	65	778

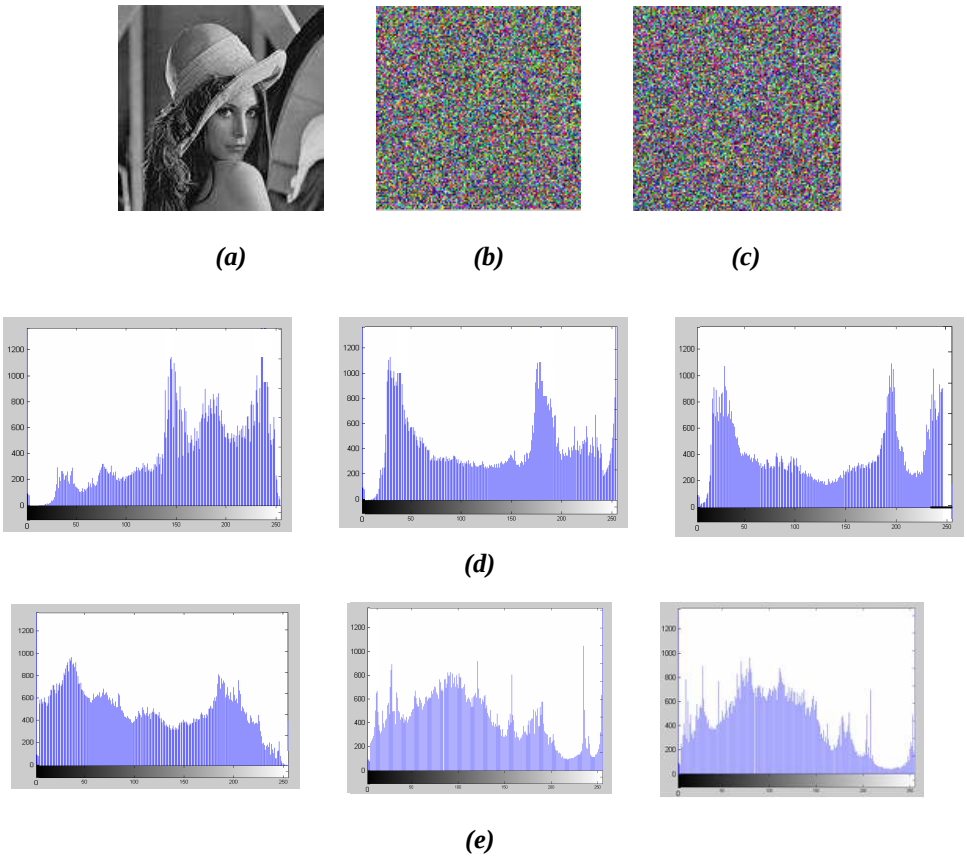
606	137	97	938	180	952	1230	90	35	1067	1084	1083
1014	463	237	814	650	319	935	379	315	309	46	676
1139	1060	1238	240	1152	645	689	846	221	1003	251	116
635	1215	182	28	467	460	171	495	377	674	80	1228
1027	392	452	54	502	957	1064	1079	228	733	932	359
609	997	647	834	1149	1004	751	166	923	830	1166	976
980	1198	1116	975	578	631	75	253	235	1095	1224	860
965	449	1109	43	979	150	763	202	271	252	394	59
516	1205	239	838	874	247	353	685	704	169	83	177
1012	389	114	39	1074	659	924	982	104	565	783	839
705	167	1222	987	656	292	57	93	194	1056	1165	282
861	500	526	564	34	411	922	1189	566	445	506	99
1119	793	257	637	398	1087	1122	1185	105	582	51	494
490	1024	199	207	696	782	596	521	412	138	50	1233
1030	993	1191	258	653	1023	1105	312	757	492	712	419
173	1055	406	948	21	713	119	378	402	929	1186	574
19	654	286	1098	327	246	852	1017	1039	183	592	936
1082	954	332	714	781	328	666	85	518	620	787	1033
20	920	347	750	546	756	1034	1181	1031	62	1035	497
22	522	519	346	1051	591	803	373	1239	1015	927	37
382	966	586	218	140	1124	410	1183	74	595	1180	942
841	1170	872	1125	742	1243	845	316	1104	106	554	159
802	967	1099	710	344	158	196	939	1195	1091	146	614
569	1002	562	1226	641	279	567	416	1178	100	1132	236
823	49	809	296	360	254	575	1013	1048	313	441	801
206	992	1193	157	48	723	1234	280	184	768	615	415
691	824	1213	762	743	753	187	233	404	262	479	1016
422	365	481	1171	661	837	323	963	686	772	605	1227
599	572	777	1072	443	717	1244	1135	1131	735	448	1047
109	657	941	1078	1071	531	865	794	1032	25	107	1100
238	621	120	334	71	795	1077	1001	1159	626	1161	451
342	23	699	1040	919	1145	510	336	715	274	739	468
1231	1172	1237	831	1150	1144	423	14	832	1188	31	1162
68	918	385	338	766	867	1050	528	216	305	577	524
972	709	1081	222	639	399	72	1212	953	1216	124	112
432	1203	1201	126	570	1146	1211	101	426	125	1136	178
209	706	1005	1174	1086	921	926	33	917	358	1011	161
977	450	42	1057	688	728	1111	847	219	456	322	970
197	999	1199	720	366	1175	201	627	1130	320	536	290
1236	995	229	868	478	1214	330	589	1008	625	355	690
968	310	973	337	488	1006	672	77	148	746	808	603
608	1070	343	667	553	354	752	989	946	520	568	628
1061	725	693	1089	110	1117	1151	326	629	1108	1092	294
855	759	16	792	1208	651	391	776	58	1164	256	800
263	916	1120	1179	424	579	547	1115	369	160	974	538
15	89	541	934	947	18	480	1049	1029	465	612	191
958	1028	986	1134	774	548	1138	1210	1197	937	1221	806
1096	542	613	1123	433	678	804	726	76	933	1090	571
1158	1043	959	1094	1041	10	1052	1106	1020	1169	724	1241
862	990	988	269	461	644	711	1153	421	602	677	684
1107	594	1206	1242	164	853	1021	475	277	1113	214	1025
1053	440	491	573	1207	1059	683	134	139	1046	318	864
285	1121	560	1026	1154	397	996	190	509	1190	350	702
1128	417	1042	438	950	673	1010	1114	515	991	665	810
91	869	964	493	188	395	108	844	152	769	335	827
1101	1129	88	435	64	472	1147	807	551	56	1126	719
1202	791	857	136	779	217	1240	444	1184	499	409	1220
668	224	940	299	1343	697	1344	616	442	95	1336	1334
1286	204	646	836	220	489	301	700	1313	288	820	561
773	1297	1345	156	607	473	476	1259	721	133	98	840
1346	351	464	559	1347	384	1348	1304	375	1349	563	1268
436	873	652	439	142	731	1264	314	111	681	393	611
784	1302	558	1270	1350	205	1351	165	13	303	361	1269
1352	734	618	381	732	363	1353	1354	664	557	380	1294
513	94	530	727	1291	244	1355	1338	367	425	454	758
333	1356	1300	1357	1248	339	1342	1341	588	1282	250	317
760	122	1308	512	1358	69	1299	129	102	1251	1359	534
331	788	1298	396	1360	268	264	103	427	482	1325	1307
523	1361	329	1324	1362	1363	1290	1364	859	455	155	663
1340	1333	819	261	1365	1366	1329	1328	786	293	284	1367
154	1331	270	1265	849	132	27	135	729	130	749	505
1368	231	638	1369	390	1370	345	540	1371	1278	1293	276
1372	1250	308	737	17	1373	241	298	362	1262	198	127
447	153	474	1272	1252	825	349	121	1315	1303	215	47
1337	1305	249	1260	186	775	671	1320	387	1374	291	870
289	1375	400	41	227	1289	144	30	55	1339	744	1316

1296	128	1319	504	1376	1245	1377	529	1378	848	431	485
1255	418	1254	708	352	1267	212	232	640	1246	1314	458
1379	722	1327	1380	634	181	118	797	123	590	1280	679
1381	1326	1258	195	1271	507	38	1382	66	1279	453	1256
1309	1383	738	483	340	655	736	533	1312	1384	420	364
842	149	780	1263	1332	1275	1274	283	816	1284	1330	1266
866	1385	117	598	818	703	200	716	408	1285	854	576
295	1386	1292	115	695	1249	1318	40	185	302	1306	1283
1288	45	487	1323	875	143	600	477	583	1281	1253	550
1387	278	619	828	324	1257	813	812	24	790	503	496
36	1388	1389	243	96	1301	1321	718	555	789	1247	388
26	1261	821	1276	162	761	1322	687	275	765	1390	1287
617	1310	141	265	44	1277	226	1391	1392	1311	29	1393
1317	856	584	1295	680	537	462	307	413	1273	61	1335
470											

Clé de chiffrement de la deuxième version chiffrée de Texte2 ($Taille_{Clé}$ = 1,8704 K octets)

681	257	54	272	127	416	212	193	126	445	404	584
108	735	917	617	694	270	386	241	702	562	898	175
381	459	904	737	596	689	130	228	326	836	460	289
37	958	638	703	356	817	741	466	971	858	1	909
688	524	53	13	448	624	634	199	369	807	116	686
102	121	204	490	397	187	670	650	522	163	256	950
868	720	153	510	660	580	68	427	698	546	523	900
621	306	312	331	721	488	314	243	350	313	452	325
844	986	82	499	494	48	12	991	526	453	479	910
978	56	409	607	172	240	620	627	391	779	15	641
174	931	71	90	55	265	963	947	981	60	84	250
95	39	977	513	748	220	235	516	507	916	401	190
360	213	362	725	447	333	105	491	857	78	881	383
89	934	705	999	123	426	231	293	323	91	943	557
803	846	324	945	791	718	826	903	396	258	839	602
713	83	859	38	639	411	400	201	359	545	487	92
138	766	879	799	346	558	928	864	49	961	365	519
149	146	371	18	281	158	430	684	277	885	754	58
374	435	269	318	358	850	335	953	567	443	970	899
169	719	291	789	809	613	432	595	275	882	247	347
62	282	851	685	97	775	151	814	601	384	439	771
980	343	552	598	106	853	541	192	612	993	687	154
271	461	496	125	708	776	948	744	939	375	232	952
286	927	675	671	483	575	373	455	155	305	421	761
731	279	869	288	642	933	788	367	122	676	46	997
35	31	830	902	873	480	736	402	751	477	336	654
450	704	696	295	236	534	244	944	988	629	230	861
341	100	433	906	786	390	750	905	307	42	351	659
714	301	395	34	205	549	77	438	215	285	454	662
871	883	949	884	165	573	935	398	728	112	888	5
985	304	412	440	505	514	219	292	920	131	965	72
349	509	833	287	157	707	308	47	472	810	492	144
339	489	734	139	233	223	474	261	533	209	987	656
912	994	475	911	968	98	772	506	553	214	610	469
132	202	159	568	925	983	938	237	224	669	777	693
538	964	891	746	431	464	604	227	722	756	773	554
50	527	142	252	529	334	171	712	465	437	515	255
874	57	758	540	161	760	733	503	792	27	486	532
822	385	462	234	407	299	583	210	446	88	544	757
436	319	700	109	478	614	393	732	246	599	556	316
682	753	767	70	586	423	796	361	570	793	177	913
73	8	414	30	372	81	709	429	456	79	64	936
508	878	578	63	780	329	442	135	110	457	597	226
870	512	876	458	812	866	266	942	765	768	302	26
330	69	76	537	724	907	3	946	608	500	561	872
4	424	23	535	865	181	120	493	539	315	160	93
589	588	611	701	200	216	661	531	419	115	677	273
399	982	380	188	890	382	24	377	749	655	194	922
673	87	653	823	267	740	85	221	44	582	310	189
813	956	834	695	995	366	410	889	600	353	140	425
590	783	838	908	672	185	819	806	918	501	892	816
955	229	930	128	778	536	542	413	560	484	559	504
975	99	941	186	867	420	984	476	518	66	606	622
405	976	141	937	592	797	378	631	485	211	33	566
666	521	723	28	635	297	957	441	7	683	251	992
626	511	101	591	21	665	716	636	996	551	548	274
585	363	979	332	96	117	609	249	129	923	644	710

637 517 738 51 43 357 969 563 222 203 322 547
470 860 344 798 355 863 248 628 921 845 841 184
847 502 65 565 134 886 848 951 940 471 594 406
774 818 663 45 468 14 645 972 303 959 770 337
855 260 124 752 824 623 619 574 451 781 962 9
825 368 498 640 196 428 877 473 166 197 463 354
415 764 118 815 587 739 276 183 67 321 94 298
564 577 136 785 897 801 444 36 808 364 895 657
820 759 179 311 495 658 147 742 376 691 16 579
167 896 370 434 625 804 467 571 805 745 387 379
664 6 2 690 652 795 967 300 422 726 827 837
572 191 831 61 150 901 225 706 924 327 729 593
320 253 17 152 543 667 954 715 254 893 697 137
263 180 11 143 74 990 19 278 875 763 119 345
605 264 408 22 576 699 802 497 309 854 929 86
680 678 842 168 482 113 394 649 679 727 296 782
835 787 615 618 338 389 25 743 692 52 550 843
114 919 207 828 80 40 284 178 730 10 238 966
242 569 794 832 103 755 392 182 75 481 849 647
821 259 974 104 294 880 418 973 603 674 998 530
616 59 32 262 348 651 195 20 133 340 894 632
156 388 403 643 829 176 581 960 29 245 198 217
762 711 717 41 239 862 747 668 148 914 555 206
449 926 280 352 417 111 932 852 342 784 633 173
887 989 162 107 218 170 328 646 208 790 915 528
290 811 769 630 648 317 840 520 283 145 164 268
525 856 800 1319 1015 1163 1032 1115 1334 1035 1004 1255
1286 1101 1309 1186 1325 1012 1335 1304 1336 1136 1337 1107
1172 1113 1249 1190 1227 1058 1010 1251 1111 1023 1338 1127
1302 1114 1033 1100 1228 1128 1112 1031 1020 1299 1109 1311
1339 1226 1081 1202 1076 1054 1295 1340 1069 1341 1198 1216
1342 1183 1343 1002 1148 1070 1176 1328 1303 1003 1121 1029
1277 1229 1089 1230 1256 1091 1175 1060 1042 1119 1182 1188
1074 1050 1132 1224 1333 1170 1078 1294 1168 1344 1149 1320
1239 1162 1203 1223 1052 1310 1269 1233 1210 1083 1265 1307
1275 1298 1045 1326 1345 1346 1272 1048 1317 1347 1261 1030
1263 1049 1285 1165 1348 1276 1349 1350 1027 1201 1235 1262
1189 1351 1313 1005 1063 1017 1195 1000 1150 1352 1016 1353
1062 1305 1130 1024 1354 1221 1073 1355 1179 1126 1356 1205
1161 1244 1240 1001 1129 1022 1197 1284 1087 1080 1006 1291
1137 1147 1152 1055 1181 1268 1252 1250 1327 1053 1160 1044
1134 1357 1072 1037 1151 1040 1225 1093 1008 1358 1056 1192
1246 1359 1271 1360 1047 1315 1090 1361 1013 1156 1362 1204
1157 1041 1082 1096 1184 1105 1123 1329 1363 1281 1097 1118
1241 1215 1364 1146 1288 1258 1365 1131 1366 1034 1092 1297
1300 1098 1367 1280 1368 1248 1018 1142 1245 1120 1177 1211
1267 1369 1173 1370 1371 1057 1372 1122 1214 1159 1237 1138
1021 1038 1166 1217 1064 1180 1208 1283 1106 1065 1185 1373
1278 1314 1374 1200 1322 1071 1220 1219 1316 1145 1169 1222
1375 1293 1067 1059 1104 1019 1260 1242 1257 1133 1376 1125
1043 1377 1167 1378 1379 1308 1140 1079 1085 1077 1232 1103
1380 1290 1381 1154 1331 1124 1266 1199 1253 1116 1270 1209
1158 1036 1102 1046 1382 1306 1095 1007 1088 1231 1108 1193
1238 1206 1039 1236 1187 1274 1383 1282 1094 1171 1384 1075
1264 1084 1254 1196 1259 1164 1155 1025 1385 1178 1386 1068
1086 1387 1318 1243 1207 1026 1321 1324 1323 1028 1213 1061
1218 1191 1212 1330 1388 1066 1332 1234 1117 1051 1153 1301
1273 1099 1389 1390 1144 1139 1143 1009 1247 1287 1014 1135
1391 1279 1174 1141 1392 1312 1110 1292 1296 1289 1011 1194
1393



(a) ***(b)*** ***(c)***

(d)

(e)

Figure V.5. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

369	423	529	171	397	412	474	15	689	64	112	390
267	21	118	304	6	28	574	465	356	338	717	638
339	110	243	458	508	450	361	608	635	115	593	749
151	742	575	716	428	109	486	584	294	722	521	141
87	695	373	225	452	602	679	13	208	85	468	634
709	298	70	297	514	601	439	359	320	251	222	292
319	765	456	614	142	645	625	288	383	217	125	476
396	718	274	236	729	483	567	114	487	492	303	542
426	628	675	713	296	505	370	730	409	395	764	641
669	460	337	264	230	204	623	358	293	418	150	520
134	328	469	762	440	533	651	551	535	286	438	462
103	355	347	63	158	16	561	205	410	643	703	432
519	693	247	620	346	40	736	366	129	652	164	327
371	506	363	154	241	447	434	624	585	163	748	376
380	144	407	59	143	667	252	10	536	548	766	360
335	455	311	497	200	466	249	751	35	119	165	341
36	741	398	448	656	18	65	233	734	17	531	512
595	660	104	586	81	435	299	101	92	309	454	436
710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579
352	739	698	306	253	577	248	735	515	342	285	61
219	138	568	705	658	427	511	532	210	357	307	153
633	711	98	433	619	691	60	20	137	733	485	324
659	419	719	684	269	392	754	68	259	284	467	295
48	300	224	541	600	715	630	613	172	66	425	558
604	38	692	34	557	263	258	384	51	317	375	111
325	146	701	472	388	607	289	94	644	377	745	580
500	598	496	240	443	594	364	554	74	637	187	712
655	747	576	145	47	501	417	490	622	162	491	626
578	699	750	14	128	493	282	226	459	351	589	209
582	545	740	596	147	275	121	362	470	53	193	291
333	290	179	523	527	42	30	402	738	525	113	374
664	646	507	609	685	756	189	700	148	260	654	464

191	350	378	176	257	538	599	603	11	702	372	571
237	12	96	170	192	653	88	416	540	25	180	522
642	166	72	246	194	707	680	33	639	159	215	75
728	227	516	681	344	90	99	559	127	513	265	7
23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555
69	708	345	494	271	238	726	393	534	132	385	587
686	647	368	203	3	481	482	332	199	167	687	560
539	495	743	44	583	91	610	690	336	31	261	403
379	255	725	498	323	386	198	572	220	484	73	322
677	606	340	272	343	58	503	381	26	421	517	761
79	676	334	348	760	49	663	235	597	566	313	627
39	414	618	518	479	55	752	186	694	563	649	175
223	133	549	420	277	510	727	305	391	228	632	404
196	206	84	52	184	755	212	581	763	429	107	697
724	590	77	737	445	71	682	502	509	120	254	413
758	640	149	616	43	195	242	239	211	611	453	757
218	86	19	310	387	2	105	117	424	244	504	116
449	591	8	229	605	399	629	182	57	461	314	588
177	234	190	102	106	714	122	617	612	216	673	312
431	326	621	80	382	721	93	45	156	665	553	767
475	174	329	394	415	547	188	331	543	365	308	46
67	670	330	157	354	302	389	480	367	232	768	27
207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631
41	471	131	201	245	54	202	444	556	544	720	270
32	688	759	442	279	478	136	197	123	273	671	437
262	457	672	268	349	95	50	488	706	546	183	221
178	160	528	696	62	9	214	615	573	489	430	97
744	22	753	524	446	213	746	731	181	231	283	661
570	401	276	78	400	4	562	526	683	301	89	477

Clé de chiffrement de la deuxième version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

156	453	55	623	217	430	383	574	180	389	257	147
405	685	168	355	721	754	738	227	187	474	143	551
127	578	506	528	477	713	723	319	702	417	384	67
512	575	469	301	86	237	703	204	366	2	268	247
340	452	642	21	12	343	128	214	390	220	262	76
495	281	286	66	397	169	445	170	219	24	48	531
261	338	478	282	635	44	191	209	735	323	410	514
255	234	43	54	516	291	97	344	720	695	129	332
461	117	522	270	89	537	298	161	515	346	758	737
253	35	75	739	240	719	131	27	548	98	455	195
78	683	710	178	595	387	400	403	633	353	350	250
687	37	462	235	119	53	200	385	192	530	179	419
392	488	61	457	45	125	1	576	58	312	552	509
546	706	276	241	617	73	585	681	751	753	517	677
399	91	473	557	701	26	302	637	632	32	565	755
135	539	47	266	110	505	17	631	647	727	303	577
427	411	493	23	743	458	259	122	285	114	489	544
182	72	582	194	264	120	439	279	669	718	292	207
101	407	446	158	475	536	284	630	215	724	25	256
172	622	650	734	87	638	450	521	416	665	762	519
700	251	491	335	287	608	742	34	426	104	370	290
7	315	77	480	330	486	183	619	730	19	331	141
599	555	229	202	359	731	757	273	504	308	167	155
230	542	654	304	165	423	360	733	705	662	648	763
28	433	92	664	694	716	175	760	656	190	171	415
263	627	470	51	100	589	4	707	311	148	88	696
16	466	498	590	111	224	363	468	239	408	162	621
36	333	373	626	561	113	341	70	717	511	379	3
371	109	508	378	463	673	693	221	367	30	388	8
212	750	748	667	374	33	358	661	289	116	447	674
566	173	364	747	314	401	500	185	328	140	634	198
676	449	527	490	260	106	588	79	672	68	759	502
591	142	443	94	420	581	698	108	471	572	296	188
436	38	395	437	545	614	163	525	605	704	121	160
587	318	294	764	307	615	107	761	337	151	422	604
326	69	412	223	300	213	472	249	678	690	580	658
765	11	57	573	99	124	184	523	402	501	226	479
649	361	137	193	618	181	174	39	248	299	579	484
62	670	345	351	636	382	347	454	41	118	644	675
74	620	745	218	252	610	265	438	280	564	60	362
404	503	271	712	418	569	550	406	236	13	600	603

96	82	547	126	568	369	199	534	602	612	485	596
365	31	441	203	682	49	216	729	85	456	722	154
641	258	254	524	668	483	616	435	598	317	532	549
768	613	46	465	628	680	460	225	297	186	246	645
321	133	459	520	196	584	145	413	59	692	231	375
688	102	444	210	293	322	643	714	274	242	339	201
749	65	325	767	691	736	726	424	766	601	144	376
349	159	429	130	583	233	18	529	709	646	625	208
269	380	377	64	594	487	451	244	639	305	176	711
20	112	756	653	71	684	464	157	29	563	189	606
553	651	699	232	50	352	138	728	497	467	152	310
103	10	507	327	715	586	245	434	15	14	746	640
205	81	56	541	526	309	177	393	134	663	562	741
725	368	597	391	394	348	272	288	238	660	40	153
136	533	295	396	571	744	84	146	342	425	164	629
123	689	567	52	320	708	166	740	316	275	559	611
481	686	267	657	93	556	197	752	386	5	336	494
139	554	428	90	83	732	313	592	671	243	448	431
518	132	95	496	277	607	535	80	306	22	540	381
570	372	6	334	652	356	211	659	149	543	499	421
228	278	492	513	357	329	679	432	482	510	115	655
9	624	283	63	150	666	354	324	476	697	409	609
440	414	593	105	442	560	558	206	222	42	538	398

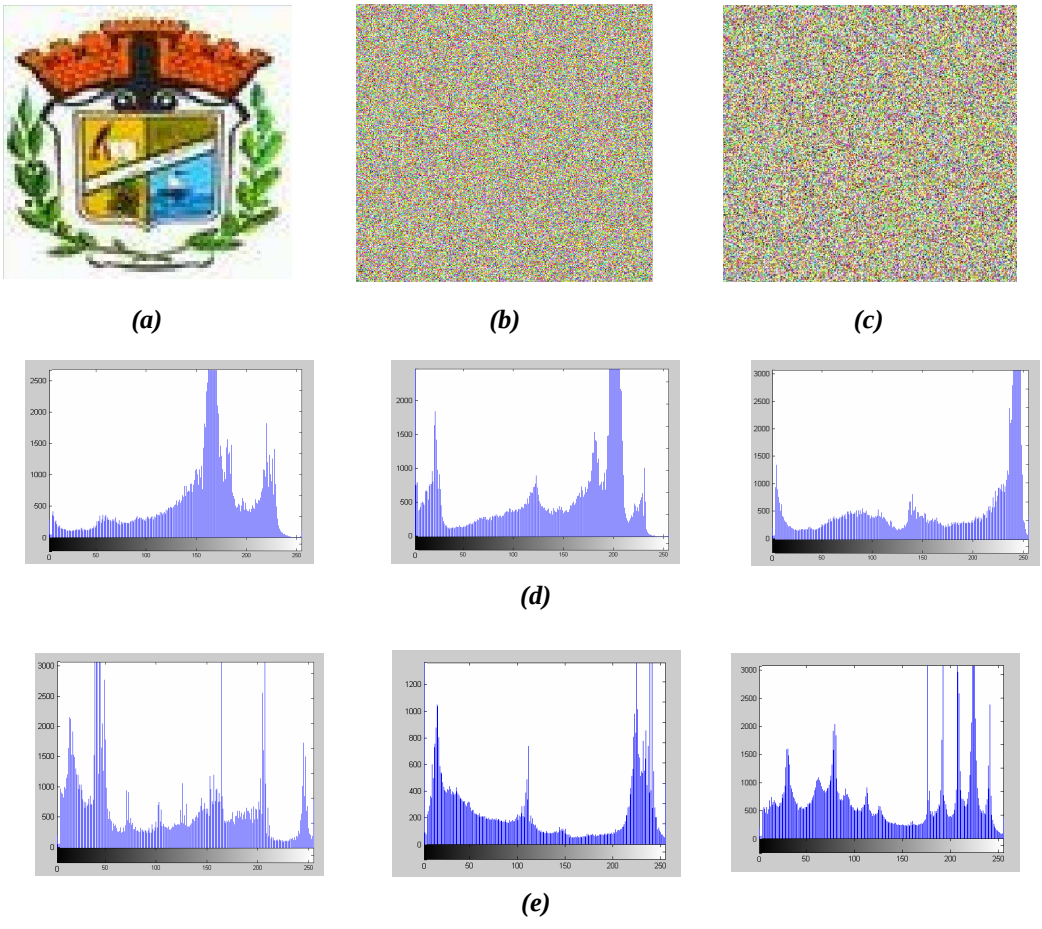


Figure V.6. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

65	9	12	62	15	21	94	7	28	11	81	3
61	36	78	70	91	25	56	97	85	4	30	24
83	32	57	40	54	31	63	8	95	26	75	55
19	96	20	38	49	92	47	44	1	98	52	89
35	67	6	72	69	53	74	60	14	43	84	23
22	82	88	73	90	76	39	13	59	33	58	68
41	50	86	46	5	51	27	99	79	10	71	66
87	48	16	2	45	64	42	80	17	34	77	29
37	93	128	625	414	134	459	456	642	446	447	205
635	230	755	387	503	179	646	371	252	744	662	574
575	475	768	734	494	638	389	441	312	571	445	214
367	308	498	643	659	705	175	676	237	530	674	370
527	626	415	330	333	472	600	238	339	137	730	257
262	144	660	139	217	323	739	329	578	420	618	735
463	199	729	207	340	336	409	551	295	296	477	222
468	276	151	258	725	178	764	245	132	366	525	305
196	717	363	484	658	136	342	397	408	198	598	443
589	455	256	239	287	216	181	290	338	111	334	751
345	392	560	697	355	585	418	18	610	162	534	154
467	241	250	694	464	461	614	500	743	170	244	741
404	259	211	645	318	650	593	442	187	282	639	288
648	267	655	220	573	436	552	344	164	225	727	165
226	667	693	416	116	265	536	532	686	522	471	587
316	260	486	514	488	268	601	628	767	365	382	314
528	195	493	430	108	227	212	145	194	753	716	675
547	152	517	374	182	539	356	677	724	760	377	434
546	119	702	204	590	481	394	114	509	395	301	419
473	490	656	728	168	583	538	426	745	518	149	141
131	279	644	140	580	701	325	348	341	285	203	298
651	263	723	235	462	691	622	722	129	437	354	163
335	398	647	122	469	603	553	737	269	511	101	240
110	361	400	402	317	186	733	277	766	142	264	172
399	613	703	588	231	315	106	669	531	380	233	180
319	504	275	113	425	153	372	748	424	281	157	234
249	581	466	200	289	533	411	630	579	223	221	376
679	569	596	188	421	758	519	146	427	243	123	331
671	508	678	206	304	364	273	440	561	324	297	713
670	274	118	576	715	611	752	620	379	489	369	750
343	278	103	479	621	485	224	526	453	229	604	352
765	653	413	714	499	169	595	762	746	465	410	403
537	100	609	327	117	683	617	570	740	492	373	417
310	505	155	543	246	390	689	102	747	710	594	616
497	381	406	756	563	548	487	632	757	720	512	636
749	474	631	294	433	476	347	661	197	718	393	709
577	174	712	665	592	328	599	435	311	375	719	385
280	480	299	261	292	423	242	159	283	696	542	761
641	349	582	458	565	171	432	431	731	520	732	597
541	605	666	209	368	202	612	337	680	156	272	652
619	633	657	135	125	150	148	506	255	495	104	313
535	540	401	521	412	320	554	634	572	684	654	332
523	627	449	690	183	460	251	682	602	664	742	685
470	309	161	350	515	266	452	640	501	444	567	606
502	429	322	483	711	454	428	726	357	544	378	177
184	291	407	763	624	591	201	218	510	391	556	124
478	293	566	557	623	629	189	721	321	219	270	232
208	302	708	608	586	362	213	388	672	707	248	568
307	271	405	130	115	663	185	253	143	681	698	545
176	422	754	695	450	396	668	109	439	210	607	105
692	673	759	637	112	516	699	121	584	286	736	358
107	147	326	353	562	228	127	513	215	687	555	491
700	649	529	190	738	191	507	138	615	704	383	524
496	193	359	133	384	564	306	550	438	173	236	386
549	167	346	120	448	457	126	300	558	192	254	451
559	482	706	303	284	160	247	688	166	351	158	360

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

498	116	41	239	486	387	497	89	154	262	303	25
129	485	272	506	489	496	446	502	195	127	234	350
240	282	491	338	501	304	256	484	29	209	382	212
238	512	511	494	335	166	495	79	437	381	299	499
480	508	505	481	358	73	482	483	492	392	427	359
507	269	510	236	487	493	243	372	167	231	182	37

94	106	490	87	333	500	71	64	16	504	488	160
503	438	97	227	1	317	336	331	324	76	150	478
509	90	203	716	401	386	194	442	477	328	529	4
762	474	132	3	709	407	735	376	86	596	2	48
651	228	704	200	345	380	201	626	18	5	23	24
385	660	531	732	145	248	562	677	373	663	725	648
736	712	453	84	365	52	569	128	170	183	526	754
43	294	746	146	224	83	112	745	434	448	764	582
400	541	436	635	339	597	390	632	717	26	738	551
383	721	237	93	156	144	463	623	375	627	727	130
152	173	728	731	281	722	220	68	707	656	153	204
193	680	252	54	95	550	429	465	759	70	542	34
55	528	149	414	763	590	232	729	218	646	639	600
458	49	472	57	678	630	664	374	593	533	750	585
190	344	384	117	631	676	539	205	208	300	577	657
697	196	696	394	141	689	131	559	445	513	99	69
669	706	693	578	32	142	756	247	169	768	100	742
217	290	59	264	393	755	330	701	703	332	743	726
266	444	221	549	199	163	659	370	270	250	621	702
268	614	740	552	21	96	435	766	245	622	326	28
278	417	316	752	15	343	733	634	430	618	311	181
340	189	325	532	557	624	765	636	747	9	470	571
140	595	341	538	178	710	308	402	296	573	63	271
125	610	553	449	215	699	20	411	519	263	692	357
690	162	464	439	471	126	666	105	242	720	147	681
604	420	12	523	714	615	739	11	35	155	184	136
192	426	118	318	546	705	368	580	65	253	283	186
682	460	53	670	719	399	724	440	749	364	450	222
45	202	561	121	389	19	109	164	47	261	223	246
77	558	408	82	476	179	423	535	56	616	8	139
38	760	80	406	521	405	527	298	441	249	454	396
723	447	708	748	91	348	684	120	180	715	314	210
617	655	751	658	683	312	391	168	607	61	598	33
404	424	673	570	601	418	295	642	346	613	694	643
159	661	111	257	50	327	287	674	589	289	39	62
219	285	443	143	30	640	274	462	757	362	124	310
397	753	60	259	214	254	216	81	92	412	548	574
555	293	276	534	695	667	671	605	473	302	665	176
433	594	255	265	591	587	516	456	267	206	518	455
369	319	306	98	431	603	292	395	291	378	602	103
191	638	107	688	40	174	581	356	85	102	469	679
711	347	46	213	409	119	138	566	547	371	342	377
211	628	641	165	554	609	425	78	421	413	108	575
525	612	568	10	280	114	349	520	536	468	517	652
451	649	88	197	198	416	226	110	134	157	185	172
592	556	337	415	66	524	734	75	241	279	113	698
744	654	315	422	351	27	515	320	687	611	686	700
297	572	653	650	361	543	629	761	14	475	619	175
586	51	579	288	355	540	466	637	244	730	606	564
353	36	691	235	230	563	379	398	44	522	567	576
7	403	718	388	363	685	565	758	599	329	133	273
137	459	229	122	625	584	17	275	322	251	187	367
410	334	645	560	158	123	633	583	305	101	151	741
428	537	457	452	313	354	207	737	277	366	675	233
258	309	148	620	479	647	58	161	352	767	323	22
42	432	467	713	588	225	668	177	321	419	104	6
360	188	644	301	135	74	72	461	672	31	307	286
171	67	260	284	530	545	608	544	115	13	662	514

Le tableau suivant (tableau V.2) représente une récapitulation des résultats de chiffrement des données tests présentées ci-dessus :

			Taille donnée (éléments)	Taille clé (bits)	Efficacité	Temps chiffrement (s)	Temps déchiffrement (s)
Données texte	Texte1	Version-Chiff1	1084	15323	15784.0	28.35	2,06
		Version-Chiff2			15030.0	28.02	2,13
	Texte2	Version-Chiff1	1151	15323	19498.0	29.16	2,67
		Version-Chiff2			18712.0	28.97.	2,65
Données images	Lena	Version-Chiff1	131 X 131	7680	139362.0	14.62	3.7
		Version-Chiff2			134074.0	14.21	4.01
	Logo	Version-Chiff1	420 X 395	7680	246426.0	15.06	3.11
		Version-Chiff2			225688.0	15.28	3.81

Tableau V.2. *Résultats obtenus par TabuCrypt.*

V.4. Discussion et évaluation des résultats

La première chose à remarquer d'après les résultats présentés dans la section précédente, est que le temps de chiffrement est proportionnellement dépendant de la taille de l'image à chiffrer. En effet, le temps de chiffrement de l'image Lena (34.826 secondes en moyenne) est plus petit par rapport au temps de chiffrement de l'image Logo (35.59 secondes en moyenne), du fait que la première est plus petite que la deuxième. La même remarque est valable pour le cas des deux données texte Texte1 et Texte2. Cette différence en temps revient au fait que le temps de chiffrement englobe le temps de lecture et de codage de la donnée à chiffrer en plus du temps de calcul, proprement dit, de la solution (temps de la recherche tabou). Le premier (temps de lecture et de codage) est strictement dépendant de la taille de la donnée à chiffrer, tandis que ce n'est pas le cas pour le deuxième (temps du calcul tabou). Donc, c'est surtout le temps de lecture et codage qui fait la différence en temps de chiffrement global d'une donnée par rapport à une autre.

De même, du coté du temps de déchiffrement c'est la phase de décodage de la donnée chiffrée qui fait la différence en temps de calcul.

Maintenant et en comparaison avec nos algorithmes développés et présentés précédemment à travers les deux précédents chapitres, le temps de calcul de ce dernier algorithme se trouve meilleur que le temps de calcul de certains de ces algorithmes et plus mauvais que le temps de calcul de certains autres algorithmes. La figure V.7 positionne le temps de calcul de l'algorithme *TabuCrypt* parmi nos quatre algorithmes PosESecL1, PosESecL2, OEEA et *AntCrypt*.

D'après le résumé de temps de calcul de nos cinq algorithmes développés par utilisation des trois métaheuristiques d'algorithmes évolutionnaires (PosESecL1, PosESecL2 et OEEA), de colonies de fourmis (*AntCrypt*) et de recherche taboue (*TabuCrypt*), nous constatons que *TabuCrypt* possède un temps de calcul meilleur que celui de PosESecL1 et de PosESecL2 mais plus grand que celui de OEEA et celui de *AntCrypt*.

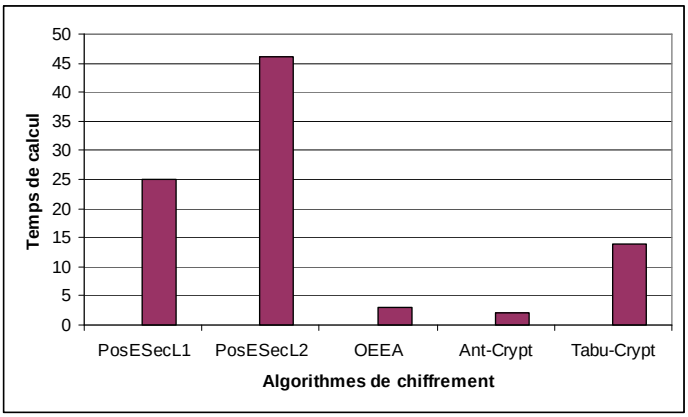


Figure V.7. *Comparaison des temps de calcul.*

Et comme *TabuCrypt* est un algorithme bâti par exploitation d'une métaheuristique, et nous l'avons expliqué précédemment, l'adaptabilité et l'efficacité d'application de telles méthodes dans un tel domaine à cause de leur large utilisation de l'aléatoire a été la source de notre motivation d'application de métaheuristique. L'algorithme développé sera, ainsi, doté d'un grand pouvoir confusionnel pour compliquer le plus possible la tâche de cryptanalyse.

En effet, l'aspect non déterministe innové à travers nos approches proposées, et par conséquent celle, ici, présentée, représente l'un des points forts de l'algorithme *TabuCrypt* tout comme nos autres algorithmes leur assurant, ainsi, une résistibilité contre les attaques différentielles. Ceci est assuré par une élection aléatoire, sur l'ensemble des générations, des voisins des éléments codant une solution intermédiaire pour calculer une autre solution intermédiaire ou une solution finale qui soit satisfaisante.
De même, l'attaque statistique sera presque impossible à appliquer grâce à ce dernier point en plus du fait que les données originales et leurs version chiffrées sur une instance, seront presque toutes différentes comme le montre le tableau suivant (tableau V.3) résumant les valeurs d'application des mesures de similarité NPCR, MAE et MSE.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.8927	0.97	0.68
Lena-version chiffrée 2	0.9104	1.05	0.71

Tableau V.3. *Niveaux de confusion de Tabu-Crypt.*

De son tour, l'attaque exhaustive sera mise à l'écart grâce à la taille de clé qui est largement sécurisée et qui est de taille égale à 15323 bits dans le cas de manipulation de données texte et qui égale à 7680 bits dans le cas de manipulation de données images.

Les attaques fréquentielle sont aussi pénalisées du fait, d'un coté, que ces dernières étudient les fréquences d'apparition des caractères dans un message (précisément un texte) écrit suivant une langue naturelle, alors que notre algorithme *TabuCrypt* peut traiter toutes sortes de message constitué de caractères du code Unicode. D'un autre coté, le message chiffré peut être constitué de caractères complètement ou partiellement différents de ceux constituant le message en clair.

V.5. Conclusion

La recherche Tabou est une méthode d'optimisation qui utilise la notion de voisinage d'une solution. Cette méthode permet, à partir d'une solution initiale, de visiter un ensemble de voisins. Ces voisins seront évalués et permettent l'évolution, à travers des règles spécifiques, vers une solution finale.

Ainsi et dans ce chapitre, nous avons présenté notre nouvel algorithme de chiffrement tabou, *TabuCrypt*, qui opère suivant le mode de chiffrement à base d'occurrence expliqué et validé dans le troisième chapitre.

Nous avons présenté en détails les différentes étapes du processus développé et nous l'avons testé sur différentes données test. Ainsi, notre objectif qui se résume en l'exploitation d'un voisinage en recherche locale de type taboue pour la manipulation de données texte et images, est atteint dans certains cotés vu les qualités de la méthode tabou (simplicité du principe, performance…) en donnant lieu à des résultats de bon niveau de confusion. Toutefois, le temps de convergence du processus de résolution est relativement lent par rapport à certains de nos précédents algorithmes proposés : OEEA et *AntCrypt*. Cela revient à l'aspect parallèle encapsulé à travers le principe soit des AEs ou des algorithmes de colonies de fourmis.

Conclusion et perspectives

Au cours de cette thèse nous avons proposé une nouvelle approche de cryptage de données texte et images par exploitation de métaheuristiques parmi lesquelles, nous nous sommes intéressés à celle inspirée du monde biologique et désignée par « approches biomimétiques ». Il s'agit des algorithmes évolutionnaires inspirés des principes de l'évolution naturelle des espèces. Par la suite, nous avons abordé une approche inspirée de modèles d'organisation naturels observés dans les sociétés animales, en particulier, dans une colonie de fourmis. Ces deux méthodes exploitées appartiennent à la catégorie des métaheuristiques à population. De même et pour l'autre catégorie qui est celle des métaheuristiques à trajectoire, nous sommes aussi arrivés à proposer un autre algorithme de cryptage utilisant le principe d'une recherche tabou.

Ainsi et après avoir étudié un panel assez diversifié des techniques de cryptage et avoir analysé les problèmes de sécurisation du transfert, nous avons élaboré nos propositions de stratégies de résolution d'un tel problème. Autrement dit et afin de détailler nos apports, notre thèse a été organisée autour de deux parties dont l'objectif de la première est de situer le lecteur dans le contexte de notre travail pour lui permettre de suivre les démarches réalisées dans cette étude. Dans la deuxième, nous avons détaillé la conception, la réalisation et l'évaluation de l'approche de résolution proposée.

Notre première contribution a porté sur l'élaboration et la conception de nouveaux algorithmes de chiffrement efficaces et robustes qui traitent le problème de taille de clés leur permettant, ainsi, d'être loin des attaques exhaustives.

Pour ce faire, nous avons commencé par une étude approfondie du problème tout en essayant de le ramener en un problème d'optimisation par une formulation adéquate aux métaheuristiques à appliquer : métaheuristique des algorithmes évolutionnaires, métaheuristique des algorithmes de colonies de fourmis et celle de recherche tabou. Pour la première, les obstacles à franchir étaient surtout la définition d'un codage adéquat de solutions. Dans ce sens, deux modes de codage différents ont été proposés : codage à base de positions et un deuxième à base d'occurrences donnant naissance, ainsi, à trois algorithmes de chiffrement, PosESecL1, PosESecL2 et OEEA de qualités différentes dont celui opérant suivant le mode de chiffrement par occurrences (OEEA) était le plus efficace et le plus robuste.

L'application de la deuxième métaheuristique a démontré l'efficacité du codage à base d'occurrences. À ce stade aussi, il a fallu définir avec précaution certains paramètres tels que la fonction d'évaluation, le mécanisme de manipulation de phéromone (incrémentation et évaporation) ainsi que celui de génération de la clé de chiffrement qui doit doter l'algorithme développé, *AntCrypt*, d'une résistance contre les attaques exhaustives vu que cet algorithme est de type symétrique.

De même, l'exploitation de la troisième métaheuristique a, aussi, abouti au développement d'un cinquième algorithme de chiffrement, *TabuCrypt*. Cela a nécessité la projection des composants d'une telle métaheuristique sur le problème de cryptage, à savoir la

liste tabou, stratégie de choix de voisins, table de hachage et les mécanismes avancés en recherche tabou (Intensification, diversification et aspiration). Les résultats étaient satisfaisants que ce soit sur le plan cryptographique que cryptanalytique, toutefois, il nécessite plus de temps de calcul que certains de nos autres algorithmes (*OEEA* et *AntCrypt*).

Notre deuxième contribution était l'innovation d'algorithmes de chiffrement non déterministes en se bénéficiant de l'aspect aléatoire de la sélection naturelle ou du déplacement des fourmis. En effet, l'aléatoire représente l'ennemi des cryptanalystes. En appliquant toutes les attaques redoutables, il est donc quasiment impossible de parvenir à l'information initiale ou au moins d'établir une relation entre les données chiffrées puisqu'on obtient toujours des versions chiffrées différentes à chaque application de l'algorithme même lors d'un chiffrement sur plusieurs instances d'une même donnée originale.

En fait, à travers notre étude nous sommes arrivés à présenter les métaheuristiques comme des méthodes de résolution approchées se voulant simples et adaptables à tout type de problèmes. Leur capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la solution trouvée. Du point de vue de la recherche opérationnelle, cet inconvénient n'est pas toujours un problème, tout spécialement quand une seule approximation de la solution optimale est recherchée. En nous situant dans ce cas, nous aurions pu mettre en évidence l'avantage de ces nouvelles approches pour résoudre le problème de cryptage de données texte ou images.

Toutefois et malgré que nos algorithmes développés soient de bonne qualité cryptographique, plusieurs autres idées et applications en relation avec ce présent travail pour le compléter, restent à concrétiser. En premier lieu, il est urgent d'aborder le problème de métaréglage [Dréo, 2004], non seulement pour régler automatiquement les paramètres des métaheuristiques, mais aussi pour maîtriser leurs fonctionnements.

De même et malgré que les algorithmes évolutionnaires ou même les algorithmes de colonies de fourmis permettent d'obtenir des solutions pas toujours optimales, mais possiblement satisfaisantes à de nombreux problèmes complexes, leur puissance d'optimisation est liée à la puissance sous-jacente des machines les exécutant.

Comme ce sont des algorithmes inspirés de la nature, ils sont fortement parallélisables et il se trouve que le récent développement des cartes graphiques, dédiées au calcul de rendu 3D dans un premier temps, puis généralisées au milieu des années 2000 avec les *General Purpose Graphic Processing Units (GPGPU)*, permet d'utiliser le grand nombre de cœurs de calcul qu'elles comportent pour autre chose que du rendu graphique.

Dans [Mait, 2011], l'auteur détaille et étudie comment les algorithmes évolutionnaires peuvent bénéficier de ces nouvelles architectures, avec leurs particularités. Plusieurs algorithmes ont été présentés, qui permettent l'utilisation efficace de ces processeurs pour l'évolution artificielle, que ce soit pour les algorithmes génétiques et stratégies d'évolution, mais aussi pour la programmation génétique. Différentes stratégies de parallélisation ont été utilisées, dont l'une nécessitant le développement d'un opérateur parallèle de réduction de la population. L'adaptation des paramètres de ces algorithmes pour permettre le portage de ces derniers sur GPGPU est détaillée, ainsi que certains points impactant directement les performances de ces portages.

Ainsi, de tels schémas peuvent être envisagés pour une parallélisation de nos algorithmes développés surtout PosESecL2 possédant un bon niveau confusionnel mais souffrant d'une lenteur de calcul vu la grande taille de chromosomes manipulés codant des données à chiffrer de grandes tailles.

D’un autre coté et vu que nos algorithmes de cryptage développés sont de type symétrique, une méthode de communication sécurisée de clé de chiffrement secrète peut être envisagée lors de la manipulation de données images. Il s’agit de marquer l'image cryptée dans des régions spécifiques, ou des régions déterminées pour l'utilisateur. Nous pouvons également utiliser des méthodes de marquage (watermarking) sans perte pour insérer la clef cryptée dans l'image cryptée. Il serait souhaitable de développer des évaluations sur la robustesse concernant les attaques. Ainsi, nous souhaitons faire des études concernant la cryptanalyse de nos méthodes pour classifier son niveau de sécurité face aux attaquants.

Par ailleurs, nos perspectives de recherche à long terme s’orientent vers l’étude et le développement d’autres méthodologies de cryptage de données. Nous nous intéresserons, en particulier à :

- La métaheuristique des automates cellulaires inventée par Ulam [Ulam, 1950] et Von Neumann [VonN, 1966] et basée sur le concept de la vie artificielle. Ce concept s’appuie sur des règles extrêmement simples permettant de construire des structures très complexes et esthétiques, d’où sa ressemblance avec le principe d’auto-organisation en biologie animale, notamment avec les mécanismes de la colonie de fourmis. Ces règles peuvent stipuler, par exemple, qu'une naissance nécessite un certain rassemblement de population, que les cellules ne peuvent survivre à un trop grand isolement et qu'une trop forte concentration les étouffe. Donc, nous comptons adapter ces règles au problème de cryptage où de génération en génération, l’application de ces règles permettra l’émergence d’une solution au problème.

- L’approche des « hyper-heuristiques » introduite dans [Cowl, 2000], [Cowl, 2002], [Burk, 2003], [Burk, 2005] et qui consiste à travailler dans un espace de recherche composé de métaheuristiques afin d’optimiser le choix de la métaheuristique à utiliser pour un problème donné. Il s’agit, donc, de mettre en confrontation un ensemble de métaheuristiques différentes pour en sélectionner la plus optimale.

Ministère de l'Enseignement Supérieur et de la recherche Scientifique

BADJI MOKHTAR-ANNABA UNIVERSITY
NIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار – عنابة

Faculté des sciences de l'ingénieur
Département d'informatique

THESE

Présentée en vue d'obtention du diplôme de ***DOCTORAT en Informatique***

Sécurisation évolutionnaire du transfert d'images

Option
Informatique

Par
SOUICI Ismahane

Directeur de Thèse
SERIDI Hamid Pr. Univ. 08 mai 1945 Guelma

Devant le jury

Président :	**KHADIR Tarek**	Pr. Univ. Badji Mokhtar Annaba
Examinateurs :		
	FARAH Nadir	Pr. Univ. Badji Mokhtar Annaba
	MEROUANI Hayet	MC. Univ. Badji Mokhtar Annaba
	KHOLADI Khiredine	Pr. Univ. Mentouri Constantine
	CHIKHI Salim	Pr. Univ. Mentouri Constantine

Année universitaire : 2012/2013

Remerciements

En premier lieu, je tiens à exprimer ma profonde gratitude à Monsieur SERIDI Hamid, Professeur à l'université 08 mai 1945 de Guelma, pour avoir dirigé ma recherche depuis le Magister et pour la confiance et l'intérêt qu'il m'a témoignés tout au long de l'élaboration de cette thèse. Je lui suis très reconnaissante pour le suivi régulier et formateur reçu durant ces quatre années de Doctorat, et bien avant, durant les trois années de Magister. Je le remercie pour la démarche scientifique rigoureuse et l'esprit d'auto-critique qu'il a su m'inculquer. J'espère avoir été à la hauteur de ces espérances.

J'adresse mes remerciements les plus sincères à Monsieur KHADIR Tarek, Professeur à l'université Badji Mokhtar d'Annaba, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Mes vifs remerciements vont aux membres du jury qui ont accepté de prendre de leur temps pour examiner mon travail : Monsieur FARAH Nadir, Professeur à l'université Badji Mokhtar d'Annaba, Madame MEROUANI Hayet, Maître de Conférences à l'université Badji Mokhtar d'Annaba, Monsieur KHOLLADI Khiredine, Professeur à l'université Mentouri de Constantine et Monsieur CHIKHI Salim, Professeur à l'Université Mentouri de Constantine.

Au cours de ces quatre années de recherche, j'ai eu à effectuer un stage de courte de durée en 2010, au laboratoire CReSTIC de l'université de Reims Champagne Ardenne, France, sous la direction de Monsieur AKDAG Herman, Professeur à l'université de Reims Champagne Ardenne, France. Qu'il trouve ici, l'expression de mes remerciements les plus sincères, pour son chaleureux accueil. Je tiens à remercier, aussi, Monsieur Cyril DE RUNZ, Maitre de Conférences à l'université de Reims Champagne Ardenne, France pour sa disponibilité, ainsi que pour ses discussions fructueuses que j'ai eues avec lui.

Merci à celles et ceux qui auront à lire tout ou une partie de ce manuscrit et qui y trouveront un intérêt quelconque.

Enfin, un grand MERCI à ma famille et, en particulier, mes parents pour le soutien et les encouragements qu'ils ont su m'apporter pour arriver au terme de cette thèse.
J'espère les honorer avec ce travail.

الملخص

في يومنا هذا أصبحت شبكات المعلوماتية معقدة و التنصتات غير القانونية عليها قائمة. لهذا أصبح من الضروري حماية المعلومات الحساسة المنقولة عبر هذه الشبكات, أين يعد علم التشفير واحد من الحلول الجد فعالة, الذي و على الرغم من كل التطورات التي سجلت بهذا المجال ماتزال بعض النقائص مسجلة خاصة ما يتعلق بحجم المفتاح المستعمل و مقاومة الهجمات المتطورة.

في هذا السياق ينصب موضوع بحثنا للدكتوراه حيث نسعى إلى تطوير خوارزمات تشفير باستعمال métaheuristiques التي تعد من الطرق التي توفر حلولا تقريبية و تفتح مجالات تصميمية لطرق حل مهمة لمشاكل التحسين.
هذه الأخيرة تنقسم إلى قسمين : metaheuristiques التي تستعمل مجموعة حلول métaheuristiques التي تستعمل حلا واحدا. وبما أن métaheuristiques توظف طابع العشوائية في معظم مراحل عملها و الذي يعد عنصرا مهم إستعماله في مجال التشفير لتعقيد مهمة المهاجمين, لهذا وقع إختيارنا على هذه الطرق لحل مشكل تشفير المعلومات النصية و الصور.

إذن وضمن الصنف الأول الذي يشمل métaheuristiques التي تستعمل مجموعة حلول قمنا بتطوير ثلاث خوارزمات تعتمد الطريقة التطورية المستوحاة من نظرية "دارون" و هذا حسب نوعين مختلفين من التشفير (التشفير المعتمد على الوضعيات و التشفير المعتمد على الظهور), بالإضافة إلى خوارزم رابع و الذي يستعمل métaheuristique مستوحاة من comportement الحقيقي للنمل.
بالنسبة للصنف الثاني والذي يضم métaheuristique التي تستعمل حلا واحدا, توصلنا إلى تكييف طريقة البحث الطابوه إستعمالها في حل المشكل لمدروس و هكذا قمنا بتطويرخوارزم تشفير طابوه خامس.

أخيرا قمنا بتجريب ومقارنة كل الخوارزمات المقترحة فيما بينها وبالنسبة لعدد من مراجع التشفير حيث أظهروا خصائص جيدة سواءا فيما يتعلق بدرجة الغموض أو مقاومة الهجمات الأكثر تطورا. قطة أخرى جد مهمة توصلنا إليها عبر عملنا هذا هي استحداث خوارزمات non déterministes , نقطة تساهم بدرجة كبيرة في زيادة قوة مثل هذه الخوارزمات.

Résumé

Aujourd'hui, les réseaux informatiques sont complexes et les écoutes illégales possibles. Il se pose donc un réel problème quant à la sécurité lors de la transmission de données. Pour des raisons éthiques, le transfert des données délicates ne peut se faire avec un tel risque et doit donc se protéger. La protection la plus adaptée pour ce type de communication réside dans la cryptographie. Cependant et malgré toutes ses évolutions et ses mises en œuvre, elle est toujours entravée par quelques défauts citant en particulier la taille des clés et la résistance contre les attaques avancées.

Ainsi, notre travail de thèse porte sur le développement de nouveaux algorithmes cryptographiques par exploitation des métaheuristiques constituant une partie importante des méthodes approchées et ouvrant des voies très intéressante en matière de conception de méthodes heuristiques pour l'optimisation. Ces dernières se scindent en deux catégories : les métaheuristiques à population de solutions et les métaheuristiques à une seule solution ou encore dites à trajectoire. Et vu que les métaheuristiques exploitent un aspect pseudo-parallèle durant différentes étapes de leurs schémas opératoires, chose qui est très intéressante à exploiter dans le domaine cryptographique pour compliquer de façon considérable la tâche des cryptanalystes, donc notre choix s'est porté sur l'utilisation de telles méthodes pour résoudre le problème de cryptage de données texte et images.

En effet, et pour la première catégorie de métaheuristiques à population, nous avons développé trois algorithmes de chiffrement exploitant les principes évolutionnaires inspirés de la théorie darwinienne, mais opérants suivant deux modes de chiffrement différents (chiffrement à base de positions et un chiffrement à base d'occurrences) ; et un quatrième algorithme utilisant une métaheuristique inspirée du comportement réel des fourmis (ACO).

Pour la deuxième catégorie qui est celle de métaheuristiques à trajectoire, nous avant arrivé à adapter la méthode de recherche taboue et l'exploiter pour résoudre notre problème étudié et ainsi développer un cinquième algorithme de cryptage tabou.

Les algorithmes proposés ont été évalués et comparés entre eux et avec les principaux algorithmes standards de cryptage où ils ont montré de bonnes caractéristiques relatives soit à leur degré confusionnel ou à leur résistibilité aux attaques les plus avancées. L'autre point crucial résultant de nos travaux de thèse était l'innovation d'algorithmes de chiffrement non déterministes, chose qui augmente considérablement leur robustesse.

Mots clés : Cryptage, attaques avancées, métaheuristiques, algorithmes évolutionnaires, recherche tabou, algorithme de colonies de fourmis.

Abstract

Today, computer networks are complex and the illegal wiretapping is possible. This raises a real problem of security when transmitting data. For ethical reasons, the transfer of delicate data can not be done with such a risk and must protect themselves. Protection best suited for this type of communication lies in cryptography. However, despite all its developments and implementations, it is still hampered by some shortcomings citing in particular the key size and resistance against advanced attacks.

Thus, the object of our thesis is the development of new cryptographic algorithms by exploitation of metaheuristics constituting a significant portion of the approximate methods and opening very interesting channels in the design of heuristic methods for optimization. The latter fall into two categories: metaheuristics of solutions population and metaheuristics of one solution or also tell trajectory metaheuristics. And since metaheuristics exploit a pseudo-parallel aspect during different stages of their operating patterns, something that is very interesting to use in the cryptographic domain to significantly complicate the task of the cryptanalysts, so our choice fell on the use of such methods to solve the problem of the encryption of text and images data.

Indeed, for the first class of metaheuristics of population, we developed three encryption algorithms exploiting the evolutionary principles inspired by the Darwinian theory but operating in two different encryption modes (positions based encryption and occurrences based encryption) and a fourth algorithm using a metaheuristic inspired by the ants real behavior (ACO).

For the second category which is trajectory metaheuristics, we arrived to adapt the method of tabu search and use it to solve the problem at hand. Thus, a fifth taboo encryption algorithm was developed.

The proposed algorithms were evaluated and compared among themselves and with principal encryption standard algorithms where they showed good characteristics for either their confusional degree or their resistibility against the most advanced attacks. The other crucial point arising from our thesis work was the innovation of non-deterministic encryption algorithms, something that greatly increases their robustness.

Table des matières

Chapitre III : CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

Chapitre IV : CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

Chapitre V : CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Dès que les hommes ont appris à communiquer, ils ont trouvé des moyens d'assurer la confidentialité d'une partie de leurs communications : l'origine de la cryptographie remonte sans doute aux origines de l'homme. En effet, le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire de les rendre inintelligibles sans une action spécifique.

Du bâton nommé « scytale » au Vie siècle avant JC, en passant par le carré de Polybe ou encore le code de César, on assista au développement plus ou moins ingénieux de techniques de chiffrement expérimentales dont la sécurité reposait essentiellement dans la confiance que leur accordaient leurs utilisateurs. Après la Première Guerre mondiale a lieu une première révolution technologique.

Mais ce n'est qu'à l'avènement de l'informatique et d'Internet que la cryptographie prend tout son sens. Les efforts conjoints d'IBM et de la NSA conduisent à l'élaboration du DES (Data Encryption Standard), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XXe siècle. À l'ère d'Internet, le nombre d'applications civiles de chiffrement (banques, télécommunications, cartes bleues…) explose. Le besoin d'apporter une sécurité accrue dans les transactions électroniques fait naître les notions de signature et authentification électroniques. La première technique de chiffrement à clef publique sûre (intimement liée à ces notions) apparaît : le RSA.

Donc, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information. C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à ramener et à modéliser le problème de cryptage comme un problème d'optimisation.

En effet, l'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solutions algorithmiques efficaces valables pour toutes les données.

Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles que les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens.

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques*. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, les chromosomes et les mécanismes d'intensification et de diversification pour la métaheuristique des algorithmes évolutionnaires), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Les métaheuristiques sont représentées essentiellement par les *méthodes de voisinage* comme le recuit simulé et la recherche tabou, et les *algorithmes évolutifs* comme les algorithmes génétiques et les stratégies d'évolution. Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi et hormis cette introduction et la conclusion générale qui reprennent les travaux de l'ensemble des chapitres et quelques perspectives majeures pour la poursuite de ce travail, le manuscrit est divisé en deux grandes parties.

Un état de l'art aussi complet que possible relatif soit au problème étudié qui est celui de cryptage, soit aux approches de résolutions exploitées qui sont les métaheuristiques, fera l'objet des deux premiers chapitres formant la première partie. En effet, le premier chapitre traite la cryptographie depuis sa première apparition jusqu'à nos jours en présentant un bref aperçu historique permettant de comprendre la distinction entre les trois disciplines (cryptologie, cryptographie et cryptanalyse), les fonctionnalités de base de la cryptographie (confidentialité, authentification, intégrité et la non-répudiation) et leurs différentes catégories (la cryptographie symétrique, asymétrique, hybride et quantique) tout en citant les fameux algorithmes appartenant aux principales catégories.
De son tour, le deuxième chapitre présente une introduction aux métaheuristiques tout en citant les concepts de base ainsi qu'une classification des métaheuristiques illustrée d'exemples d'algorithmes de chacune des classes.

Dans la deuxième partie de notre recherche, nous proposons un nouvel axe de recherche que représente l'application des métaheuristiques pour résoudre le problème de cryptage de données textes et images. Ainsi, les deux catégories de métaheuristiques ont été exploitées : métaheuristiques à population et les métaheuristiques à trajectoire.
Dans le cadre de la première, nous avons choisi d'exploiter deux métaheuristiques qui sont les algorithmes évolutionnaires (AEs) et les algorithmes de colonies de fourmis. Le principal avantage de ces méthodes heuristiques vient de leur capacité à traiter le problème de cryptage en ne possédant qu'un minimum d'informations sur celui-ci. Chose qui est primordiale dans ce problème pour augmenter la confusion des algorithmes développés.

Ainsi, le troisième chapitre résume l'application d'AEs où les différentes étapes partant du codage sont explicitées. En effet, pour un problème à résoudre, il est nécessaire d'adapter la représentation des individus aux objectifs recherchés. Cette adaptation permet de faire converger l'AE plus ou moins rapidement et de tenir compte naturellement des contraintes de la modélisation du problème. Dans le cas présent, l'espace de recherche que nous cherchons à optimiser est l'ensemble représenté par les solutions possibles du problème de cryptage et par une fonction d'évaluation de chaque solution.

Le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, les algorithmes ont été groupés en deux catégories distinctes suivant les modes de chiffrement utilisé implémentant différents codages : chiffrement à base de position ou chiffrement à base d'occurrences.

D'une manière globale, nous allons définir un individu de la population comme une donnée chiffrée possible. Cet individu doit pouvoir être évalué numériquement par la fonction d'évaluation proposée. Cette fonction de fitness calcule la qualité d'une donnée chiffrée en fonction du besoin de confusion. Nous montrerons aussi l'application de notre méthode sur quelques exemples d'images de différentes tailles. Une interprétation et une discussion des résultats obtenus seront abordées pour pouvoir, par la suite, comparer les nouveaux algorithmes proposés de cryptage évolutionnaire où l'algorithme opérant suivant le mode de chiffrement par occurrences a montré une efficacité en termes de temps de calcul, de pouvoir de confusion, de résistibilité aux attaques et de longueur de clé meilleure que celles des algorithmes opérants suivant le mode de chiffrement par positions.

Cette conclusion a été démontrée par l'application d'une deuxième méthode de résolution exploitant les algorithmes de colonies de fourmis illustrée sur un quatrième chapitre où les résultats obtenus été très satisfaisants.

Au niveau du cinquième chapitre, un autre algorithme de cryptage a été proposé s'inscrivant cette fois-ci, sous la deuxième catégorie de métaheuristiques dans le but d'explorer l'espace de recherche par exploitation de la notion de voisinage. Il s'agit d'un algorithme utilisant le principe d'une recherche tabou pour pouvoir choisir la configuration représentant la meilleure solution au problème de cryptage de données texte ou images.

Dans chacun des chapitres décrivant nos algorithmes de cryptage proposés, nous présentons des exemples et résultats expérimentaux sur des données tests. Nous avons aussi effectué des bilans en faisant ressortir les avantages et faiblesses des méthodes développées.

CHAPITRE I

CRYPTOGRAPHIE

I.1. Introduction

Depuis les temps historiques les plus reculés, l'homme a perçu le besoin de cacher, de dissimuler ou faire mystère des informations personnelles ou confidentielles, et cela bien avant l'ère informatique afin de les rendre inintelligibles à des lecteurs indésirables. C'est pourquoi, les codes ont existé.

Les origines de la cryptographie semblent remonter à plus de 4000 ans en Egypte. Plusieurs indications archéologiques tendent à montrer que les « écritures secrètes » sont en fait anciennes que l'invention de l'écriture elle-même. Polybius développa un système de codage des lettres de l'alphabet consistant à remplacer chaque lettre de l'alphabet par deux nombres, donnant la ligne et la colonne où se trouve cette lettre dans une matrice. Jules César utilisait une simple méthode de substitution de lettres pour communiquer secrètement avec ses généraux : c'est un chiffre par décalage,…etc.

C'est cependant au cours de la seconde guerre mondiale que la cryptographie s'inscrit véritablement comme élément central des stratégies militaires. Le cas le plus connu est certainement l'histoire entourant le décodage du code Enigma par les Polonais et les Britanniques. Une conjonction d'espionnage classique et d'efforts de mathématiciens polonais permet de déduire la clé utilisée et ainsi de décoder les messages encodés avec Enigma. On trouve apparemment bien moins de détails sur les efforts cryptographiques durant la guerre froide, probablement parce que ces informations sont encore « Top Secret ».

On en est maintenant à l'époque moderne où le champ d'application de la cryptologie s'est élargi et a trouvé un regain d'actualité avec toutes les applications nouvelles suscitées par l'utilisation de l'Internet. La révolution d'Internet et l'utilisation de plus en plus d'informations massives sous forme numérique facilitent les communications et rendent de ce fait plus fragiles les informations que l'on détient, c'est pourquoi il devient nécessaire de protéger le contenu de certains messages des inévitables curieux. En effet, les réseaux ouverts créent des brèches de sécurité et il est plus aisé à un adversaire d'accéder aux informations. Dans une communication à distance, des questions cruciales se posent et leurs réponses s'imposent ; comment être sur :

- que l'on parle à la bonne personne (authenticité),
- que nos propos ne sont pas altérés (intégrité),
- que la conversation n'est pas espionnée (confidentialité),
- de l'identité de l'émetteur (non répudiation) ?

Indéniablement, avec l'essor fulgurant des nouvelles technologies, le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements.

Alors que la cryptographie consiste à sécuriser les données, tandis que, la cryptanalyse est l'étude des informations cryptées afin d'en découvrir le secret. Grâce à la cryptanalyse, les militaires ont pu mener leurs guerres en découvrant les correspondances de leurs ennemis et en contrôlant les réseaux de communications mais d'autre part si elle est utilisée par une personne mal intentionnée il peut générer des dégâts onéreux aux entreprises ou aux sociétés.

I.2. Les Fondements de la Cryptographie

I.2.1. Terminologie

Comme toute science, la cryptographie possède son propre langage. Dans ce qui suit les mots clés de domaine cryptographique.

I.2.1.1. Cryptographie

Le terme cryptographie vient en effet des deux mots grecs : *Kruptus* qu'on peut traduire comme secret et *Graphein* pour écriture. Ainsi la cryptographie est l'art de dissimuler une information écrite en clair (plain text) en *cryptogramme* (cipher text) pour qu'elle soit incompréhensible que par son destinataire légitime par le biais d'une clé appelé « clé de chiffrement » (processus de *chiffrement*). Pour rendre l'information à nouveau intelligible par le biais d'une clé appelé « clé de déchiffrement » le processus inverse est appliqué (processus de *déchiffrement*).

On distingue généralement deux types de clefs :

- **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

Le schéma suivant illustre le processus cryptographique :

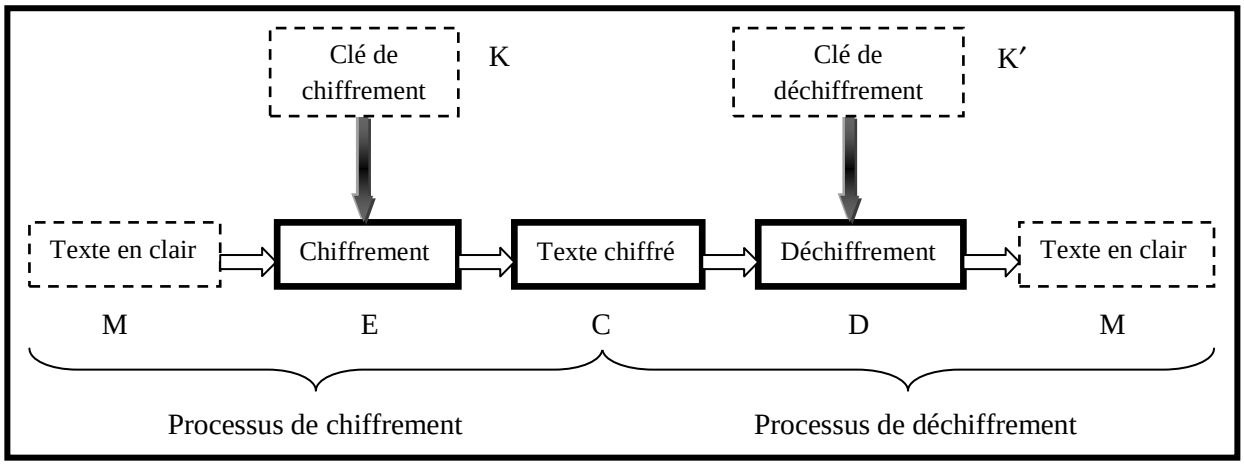


Figure I.1. *Processus cryptographique.*

I.2.1.2. Cryptanalyse

a. Définition

C'est l'art d'étude des crypto systèmes en cherchant leurs failles et leurs vulnérabilités afin de retrouver des messages clairs correspondant à des messages chiffrés sans avoir à connaître les clés utilisées dans le chiffrement. Lorsque tous les éléments de la méthode utilisée pour coder des messages sont repérés, on dit qu'on a **cassé** ou **brisé** le système cryptographique utilisé. Plus un système est difficile à briser, plus il est sûr.
La personne qui pratique la cryptanalyse est appelée : cryptanalyste. Il tente à décrypter le message chiffré pour découvrir son secret. On a distingué entre le verbe « décrypter » et « déchiffrer » puisque ce dernier est réservé pour le déchiffrement par le destinataire légitime.

b. Types de cryptanalyse

On distingue plusieurs types d'attaques :

- ***Attaque sur texte chiffré seul (ciphertext-only)*** : l'attaquant a seulement la possibilité d'intercepter un ou plusieurs messages chiffrés. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- ***Attaque à texte clair connu (known-plaintext attack*)** : se base sur la connaissance d'une partie du texte en clair pour déduire le reste du message. La tâche est de retrouver la clef utilisée pour chiffrer ce message.
- ***Attaque à texte clair choisi (chosen-plaintext attack)*** : se base sur la possibilité de choisir un texte clair et d'obtenir son chiffrement et en ayant la possibilité de générer les versions chiffrées de messages clair avec un algorithme considérer comme une **boîte noire** tel que les algorithmes à clé publique puisque l'algorithme est public.
- ***Attaque à texte chiffré choisi (chosen-ciphertext attack)*** : le cryptanalyste possède des messages chiffrés et essaye de les déchiffrer de son choix. Sa tâche est de retrouver la clef.

c. Familles d'attaques cryptanalytiques

Il existe plusieurs familles d'attaques cryptanalytiques, les plus connues sont les suivantes :

- **L'analyse fréquentielle** : examine les répétitions des lettres du message chiffré afin de trouver la clé. Cette technique est découverte par Al-Kindi au IXe siècle [www1] contre les chiffrements mono-alphabétiques. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est basée sur le fait que, dans chaque langue, certaines lettres ou combinaisons de lettres apparaissent avec une certaine fréquence.
- **L'attaque par dictionnaire** : le mot testé est pris dans une liste prédéfinis contenant les mots de passe les plus courants et aussi des variantes de ceux-ci. Ces listes sont généralement dans toutes les langues les plus utilisées, elles contiennent des mots existants ou des mots diminutifs (par exemple « powa » pour « power » ou « K7 » pour « cassette »). Elle est souvent couplée à l'attaque par force brute.
- **L'attaque par force brute** : s'appuie sur le cassage d'un mot de passe en testant tous les mots de passe possibles. C'est le seul moyen de récupérer la clé dans les algorithmes les plus modernes et encore inviolés comme AES.
- **La cryptanalyse linéaire** : c'est une attaque à texte clair inventée par le japonais Mitsuru Matsui [www1]. L'idée est de trouver des approximations linéaires entre les bits de sortie, les bits d'entrée et les bits de la clé. Si certaines de ces approximations apparaissent avec une probabilité suffisante, on a alors démontré que la correspondance

entre entrée et sortie n'est pas purement aléatoire. Le nombre de clés à envisager pour déchiffrer le message est restreint.

- **La cryptanalyse différentielle** : découverte par deux cryptologies israéliennes : Bihan et Shamir [www1]. Elle consiste à comparer les sorties de l'algorithme quand on lui met en entrée deux messages ayant une différence fixe. On étudie comme variant les sorties si les deux messages ne diffèrent que par un seul bit. Si en déplaçant ce bit à l'intérieur des messages, certains bits des sorties restent inchangés, on a alors trouvé une faille dans l'algorithme et celui-ci est attaquable.

I.2.1.3. Cryptologie

La cryptologie du grec kruptos « secret, caché » et logos « discours » qui embrasse à la fois la cryptographie et la cryptanalyse. Elle se partage entre la cryptographie, qui inclut la conception des mécanismes destinés à assurer les fonctions suivantes : confidentialité, intégrité, authentification et traçabilité (non répudiation), et la cryptanalyse dont le but est de déjouer les protections ainsi mises en place.

I.2.1.4. Fonction de hachage

Une fonction de hachage est une fonction mathématique qui à partir d'un message (d'une donnée) génère une autre chaîne, l'empreinte (généralement plus courte). Cette empreinte est très sensible au texte initial (une petite modification du texte provoque une grande modification d'empreinte).

I.2.1.5. Signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message. Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur.

I.2.1.6. Les certificats

Pour assurer l'intégrité des clés publiques, ces dernières sont publiées avec un certificat. Un certificat (ou certificat de clés publiques) est une structure de données qui est uniquement signée par une autorité certifiée.

CA, Certification Autorité, est une autorité en qui les utilisateurs peuvent faire confiance. Il contient une série de valeurs, comme le nom du certificat et son utilisation, des informations identifiants le propriétaire et la clé publique, la clé publique elle-même, la date d'expiration et le nom de l'organisme de certificats. Le CA utilise sa clé privée pour signer le certificat et assure ainsi une sécurité supplémentaire. Si le récepteur connaît la clé publique du CA, il peut vérifier que le certificat provient vraiment de l'autorité concernée et, ainsi, de s'assurer que le certificat contient des informations viables et une clé publique valide.

I.2.2. Fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur fiabilité et confidentialité au travers d'un canal peu sûr (téléphone, réseau informatique ou autre). Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions en plus de la fiabilité et la confidentialité. Il s'agit de garantir l'intégrité et l'authenticité des données échangées. Les postulats de sécurité associés à la cryptographie sont :

a. Intégrité

Vérifier l'intégrité des données consiste à déterminer si les données, ressources, traitements ou services n'ont pas été altérées durant la communication de manière fortuite ou intentionnelle.

b. Confidentialité

La confidentialité est le maintien du secret des informations. Elle consiste à rendre l'information discrète ou inintelligible à d'autres personnes que les seuls acteurs de la transaction. La confidentialité peut être vue comme la «protection des données contre une divulgation non autorisée» [Gher, 2004].

c. Authentification

L'authentification consiste à assurer l'identité d'un utilisateur c.à.d. de garantir à chacun de correspondant que son partenaire est bien celui qu'il croit être.
On distingue deux types d'authentification :

- ***Le contrôle d'accès :*** C'est l'opération permettant d'être certain de l'identité d'une personne utilisateur pour permettre l'accès à des ressources uniquement pour la personne autorisée (par exemple l'utilisation d'un mot de passe pour un disque dur).
- ***Authentification de l'origine des données :*** Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services, l'authentification et l'intégrité.

d. La non répudiation (traçabilité)

La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction ; l'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information.

I.2.3. Problèmes de la cryptographie

Quelque soit le cryptosystème utilisé pour le chiffrement, ceci reste toujours cassable un jour ou l'autre. La sécurité d'un cryptosystème repose en fait sur la complexité des algorithmes définis et sur les puissances de calcul disponibles pour une attaque. Ainsi la solution adoptée actuellement est de faire « *retarder le travail des cryptanalystes* », c.à.d. faire en sorte que la durée nécessaire pour déchiffrer un code soit supérieure à la durée de validité des données.

I.3. Algorithmes cryptographiques

Au cours d'un échange visant à communiquer de façon secrète deux protagonistes, appelé ici l'émetteur et le récepteur à travers un canal peu sûr, l'information à transmettre sera donc chiffrée par un procédé de chiffrement en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de en utilisant une clé de déchiffrement sans laquelle son procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique : *algorithme cryptographique* (ou *chiffre***)**. La sécurité des données chiffrées est entièrement dépendante de deux facteurs : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un *cryptosystème***.**

I.3.1. Le principe de Kerckhoffs

Pour briser un cryptosystème, un opposant cherche à obtenir deux éléments d'information :

1. Quel est le type de système de codage utilisé ? et,
2. Quelle est la clé d'encodage utilisée ?

Bien entendu, son travail est simplifié (mais certainement pas terminé) s'il connait le type de système utilisé. Avec le temps cette information finit par circuler. Cette hypothèse de travail est appelée le principe de Kerckhoffs. Ce principe consiste à affirmer que la sécurité d'un système de chiffrement ne devrait pas être fondée sur le secret de la procédure utilisée, mais essentiellement sur le secret de la clé.

Auguste Kerckhoffs écrit en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire », où il disait [Kerc, 1883] :

« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants:

1) le système doit être matériellement, sinon mathématiquement, indéchiffrable.
2) il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.
3) la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
4) il faut qu'il soit applicable à la correspondance télégraphique.
5) il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
6) enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer ».

Les points 2 et 3 sont les axiomes fondamentaux de la cryptographie suivant lesquels l'attaquant possède tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé spécifique pour le chiffrement. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ce secret sera découvert ou volé.

I.3.2. Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplait (P, C, K, E, D) satisfaisant les points suivants :

1) P est un ensemble fini de blocs de textes clairs possibles.
2) C est un ensemble fini de blocs de textes chiffrés possibles.
3) K est un ensemble fini de clefs possibles.
4) Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k (e_k (x)) = x$ pour tout texte clair $x \in P$.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$ qui doit être transmise préalablement loin des yeux d'Oscar (à travers un canal de communication sûr). Supposant qu'Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne : $x = x_1 x_2 \ldots x_n$ avec : $n \in Z$, $n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera : $y = y_1 y_2 \ldots y_n$.

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1 x_2 \ldots x_n$. Le procédé de communication est illustré sur la Figure I.2.

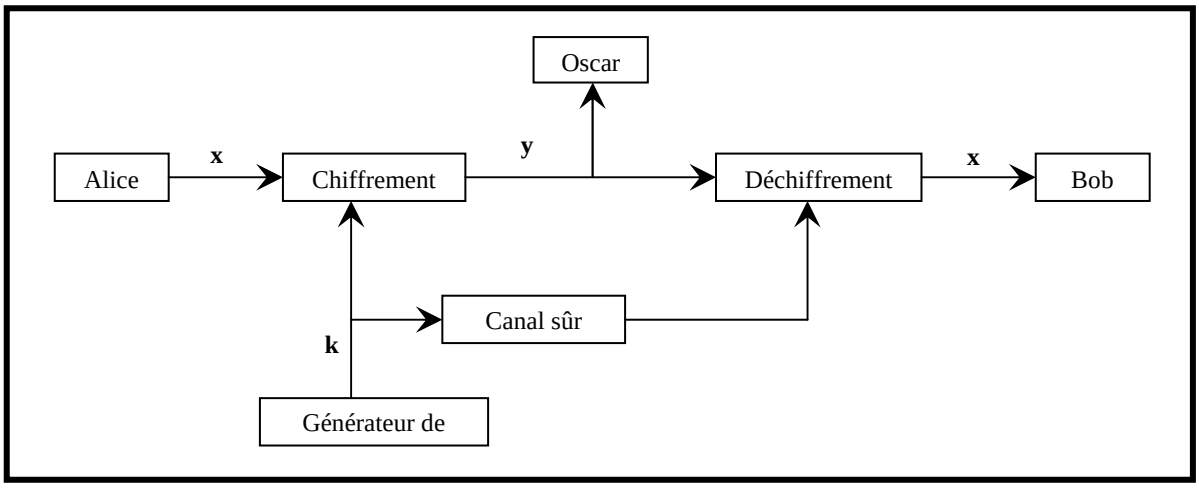


Figure I.2. *Le procédé de communication.*

I.3.3. Classes de cryptographie

Le schéma suivant illustre les différentes classes de la cryptographie :

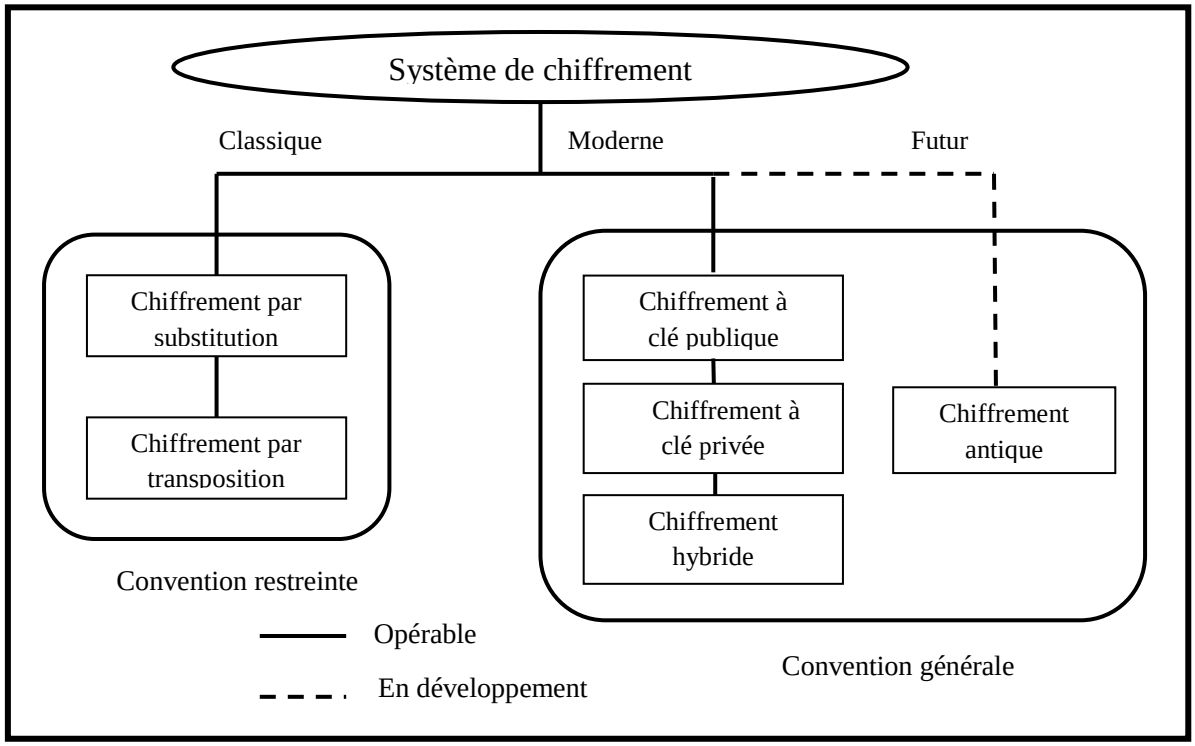


Figure I.3. *Les classes de la cryptographie*

I.3.3.1. La cryptographie classique

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble et ils sont trop simples pour offrir la moindre sécurité. Pour cacher la substance d'un texte, ils utilisent la substitution de caractères par d'autres ou les transposer dans des ordres différents. De ce fait, la confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide. On appelle généralement cette classe de méthodes : le chiffrement **à usage restreint**.

a. Cryptographie par substitution

La substitution signifie que chaque lettre (ou groupe de lettres) est substituée par une (ou groupe) lettre(s), chiffre(s) ou symbole(s). Le déchiffrement consiste à effectuer la substitution inverse. Selon la façon de substituer, on quatre catégories :

- **Substitution simple (mono-alphabétique)**

Le codage par substitution mono-alphabétique (ou encore les alphabets désordonnés) est le plus simple à imaginer. Chaque lettre dans le message clair est remplacée dans le message chiffré par une autre lettre différente unique pour toutes les occurrences de celle-ci.

Dans la littérature, plusieurs algorithmes ont été proposés, entre autres, nous citons : le chiffre de César, le chiffre Atbash, le carré de Polybe, etc.

- **Substitution poly-alphabétique**

Au lieu de remplacer une lettre par une même autre lettre dans tout le message comme dans la substitution simple, elle est remplacée périodiquement par différentes lettres. L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles.

- **Substitutions homophoniques**

Au lieu d'associer un seul caractère crypté à un caractère en clair, on dispose d'un ensemble de possibilités de substitution de caractères dans lequel on choisit aléatoirement. Par exemple : C=>S, K ; G=>G, J ; Q=>K ; S=>S, Z ; PH=>F ; …etc.

- **Substitution par polygrammes**

Au lieu de substituer des caractères, on substitue par exemple des digrammes : groupe de deux caractères. Pour se faire, deux moyen sont utilisés : soit par table (Chiffre de Playfair) ou par transformation mathématique (Chiffre de Hill).

b. Cryptographie par transposition

Elle consiste à permuter les lettres du message à chiffrer entre elles, afin de le rendre inintelligible. Plusieurs variations de transposition sont utilisées, parmi eux on trouve :

- **Transposition simple (à base matricielle)**

Elle consiste à écrire le texte en clair dans une matrice de ***n*** colonnes (une lettre dans chaque case), et ensuite de construire le texte chiffré en prenant les lettres à partir de cette matrice colonne par colonne. La clé dans ce cas est le nombre ***n***.

- **Transposition avec substitution simple**

L'idée dans ce cas est de combiner la transposition avec une substitution simple. Il s'agit ainsi de chiffrer le message clair par une méthode de substitution simple, et en suite d'en appliquer une transposition. Une autre astuce est souvent utilisée qui consiste à appliquer une fonction de permutation sur l'ordre d'arrangement des colonnes. On cite à titre d'exemple : le chiffre de DELASTELLE.

c. Conclusion

Les nouvelles techniques de communications (moyens de transports rapides, journaux, télégraphe, télégraphie sans fil) donne une nouvelle impulsion à la cryptologie. L'interception devient simple et le décryptement des informations devient vital. La cryptologie entre dans son ère moderne et les algorithmes classiques ne sont plus efficaces.

I.3.3.2. La cryptographie moderne

La cryptographie entre dans son ère moderne avec l'utilisation intensive des ordinateurs à partir des années septante. Vue la nécessité croissante de sécuriser les données dans tous les domaines (économique, industriel, informatique,...etc.) …. La cryptographie est appelée à devenir une technique de plus en plus fondamentale pour la protection des informations possédées et/ou échangées par des individus ou des organisations. Dans la cryptographie moderne, on utilise aussi des problèmes mathématiques que l'on ne sait pas (encore) résoudre, par exemple factoriser des grands nombres (chiffre RSA).

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ la *cryptographie à clef secrète,* ou encore appelée *symétrique,*
- ✓ la *cryptographie à clef publique,* dite également *asymétrique*.

a. La cryptographie symétrique

- **Principe**

Le cryptage à clé privée ou symétrique (ou encore dit conventionnel) utilise la même clé pour chiffrer et déchiffrer un message. Autrement dit, les clefs de chiffrement et de déchiffrement sont identiques ou on peut facilement calculer l'une à partir de l'autre. La connaissance de cette clef est cruciale pour la confidentialité des informations échangées. L'emploi d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Ces systèmes sont vulnérables parce qu'il est généralement possible de découvrir la clé à partir des messages codés. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1}. En excepte le système à masque jetable qui est hors porté de cette attaque.

Les algorithmes symétriques sont de deux types :

- ✓ Les algorithmes de *chiffrement en continu*, qui agissent sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé *chiffrement en flux* (Stream cipher).
- ✓ Les algorithmes de *chiffrement par blocs* (Bloc cipher), qui opèrent sur le texte en clair par groupes de bits appelés blocs.

Plusieurs algorithmes de chiffrement symétriques ont été définis. Nous présentons, ci-dessous, les plus connues de ces méthodes.

- **Quelques algorithmes de chiffrement symétrique**

A ce niveau, on va présenter, plus ou moins en détails, les plus fameux des algorithmes de chiffrement s'inscrivant sous ce mode.

1) Masque jetable (one-time pad)

Ce chiffre appelé aussi chiffre de Vernam ou encore le chiffrement parfait; est un algorithme de cryptographie inventé par Gilbert Vernam en 1917. Ce chiffrement est le seul qui soit théoriquement impossible à casser puisque la sécurité de ce système repose sur la génération complètement aléatoire de la clé [www2]. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront équiprobables. Malgré cela, il présente d'importantes difficultés de mise en œuvre pratique, il ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

Il consiste à combiner le message en clair avec une clé présentant les caractéristiques suivantes :

- choisir une clé K_M aussi longue que le texte à chiffrer.
- utiliser une clé constituée de caractères choisis aléatoirement.
- ne jamais utiliser 2 fois la même clé (d'où le nom de masque jetable).
- pour chiffrer un message faire le ou exclusif du message et de la clé : $C=M \oplus K_M$.
- pour déchiffrer un message l'opération est la même : $M= C \oplus K_M = M \oplus K_M \oplus K_M$.

2) DES (Data Encryption Standard)

Jusqu'aux années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence de besoins civils, le NBS (National Bureau of Standards) lança le 15 mai 1973 un appel d'offres dans le Federal Register (l'équivalent du Journal Officiel américain) [Stin, 1996] pour la création d'un système cryptographique qui doit :

- Reposer sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement.
- Etre facile à implémenter, logiciellment et matériellement et être très rapide aussi.
- Avoir un haut niveau de sécurité, lié uniquement à la clé et non à sa confidentialité.

Ce cryptosystème permet de chiffrer des messages de 64 bits avec une clef de 56 bits. De ce fait, c'est un système de chiffrement par blocs. Pour chiffrer un texte, il faut d'abord le découper en blocs de 64 bits puis appliquer le chiffrement sur chacun des blocs. Ainsi, les données en entrée de cet algorithme (texte clair et les données en sortie (texte chiffré) seront des bocs de 64 bits.
Sa clé est une chaîne binaire de 64 bits, mais en fait seuls 56 bits servent réellement à définir la clé. Les bits 8, 16, 24, 32, 40, 48, 56, 64 sont des bits de parité. Le $8^{ème}$ bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Ceci permet d'éviter les erreurs de transmission. Il y a donc pour DES 2^{56} clés possibles, soit environ 72 milliards possibilités. Malgré ça, c'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, alors qu'elle a été largement suffisante au moment de sa conception.

Nous donnons ci-dessous une idée simple et intuitive du fonctionnement du DES qui est un algorithme relativement simple puisqu'il combine des permutations et des substitutions. Il se déroule en quatre étapes [www3] :

1. **Préparation-Diversification de la clé :** le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K 16 sous-clésK_1,…, K_{16} à 48 bits.
   La figure ci-dessous montre comment obtenir à partir d'une clé de 64 bits 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES.

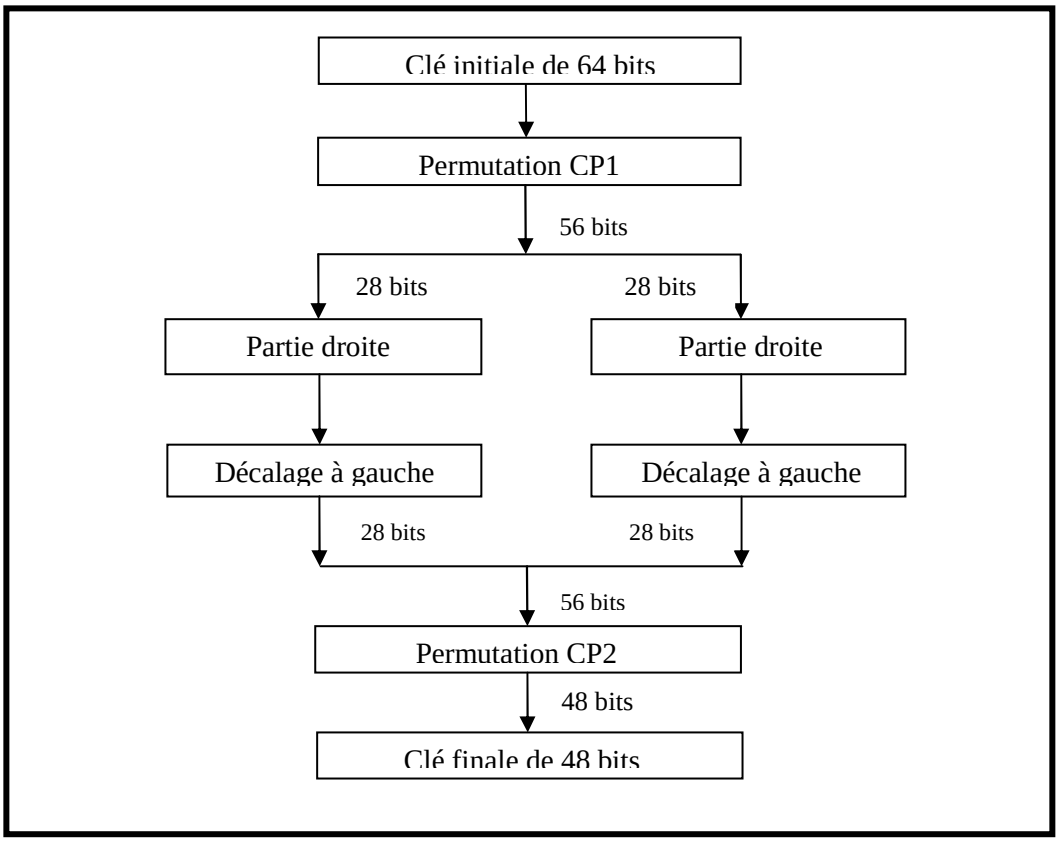


Figure I.4. *Génération des clés.*

La première étape consiste à faire une permutation noté CP1 dont la matrice est présentée ci-dessous, puis éliminer les bits de parité afin d'obtenir une clé d'une longueur de 56 bits. Elle est composée de deux matrice G_i et D_i chacune de 28 bits. Ces deux blocs subissent ensuite une rotation à gauche. Et enfin, ces derniers sont regroupé en un bloc de 56 bits ensuite une permutation CP2 a eu place pour fournir en sortie une clé de 48 bits. Des itérations de l'algorithme permettent de donner les 16 clés K_1, …, K_{16}.

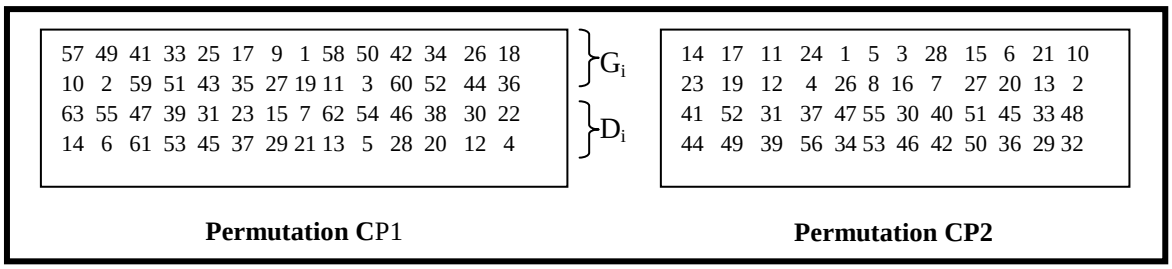


Figure I.5. *Permutation CP1 et CP2.*

2. **Permutation initiale :** pour chaque bloc de 64 bits x du texte, on calcule une permutation finie y=PI(x). y est représenté sous la forme : y = G_0 D_0 , G_{16} étant les 32 bits à gauche de y, D_0 les 32 bits à droite (voir Figure I.6). Quant à leurs significations, par exemple, la permutation initiale (IP) signifie que le $58^{ème}$ bit de la chaine à chiffrer, *x*, est le premier bit de IP(*x*),…

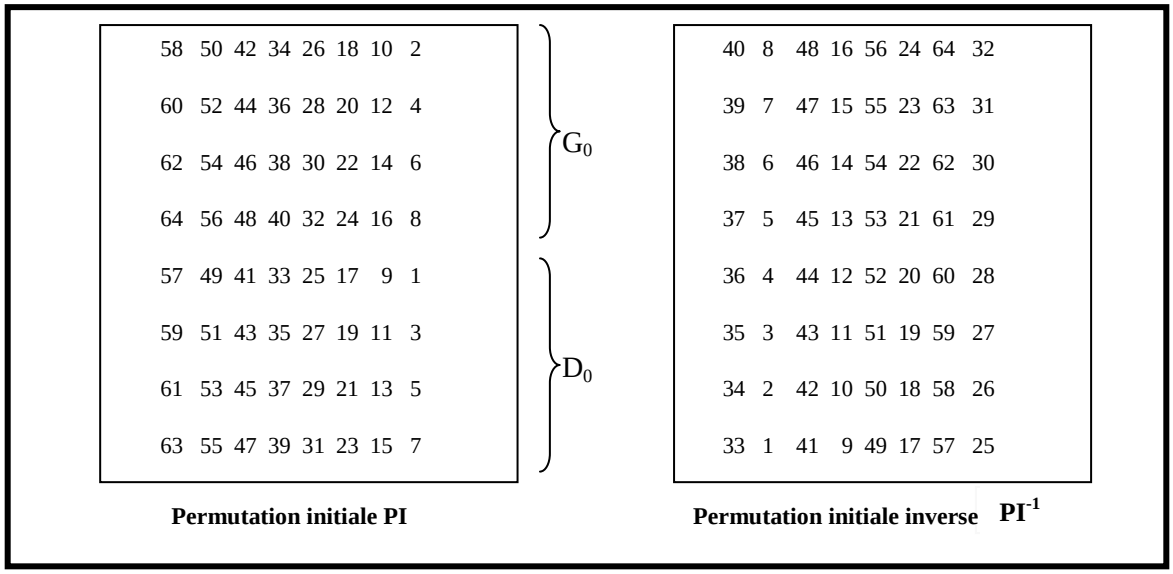


Figure I.6. *La permutation initiale et son inverse.*

3. **Itérations (Rondes) :** On applique 16 rondes d'une même fonction. A partir de $G_{i-1}D_{i-1}$ (pour i de 1 à 16), on calcule G_iD_i en posant :
   - $G_i = D_{i-1}$
   - $D_i = G_{i-1} \oplus f(D_{i-1},K_i)$

D'après la formule qui correspond au calcul de D_i, on constate que la fonction *f* utilise deux arguments ayant des tailles différentes : D_{i-1} de 32 bits et k_i de 48 bits. Ainsi, D_{i-1} sera étendu en 48 bits grâce à une matrice appelée table d'expansion (notée E), dont les 48 bits sont mélangés et 16 d'entre eux sont dupliqués (voire Figure I.7).

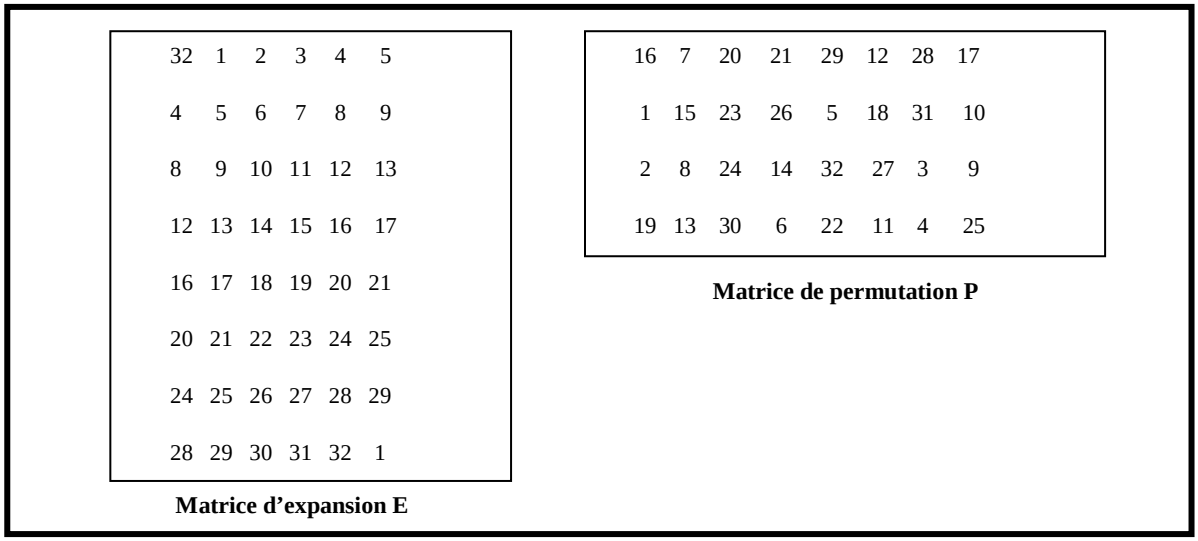


Figure I.7. *Matrice d'expansion E et de permutation P.*

L'opération qui suit consiste à calculer un « ou exclusif » entre le résultat obtenu jusqu'à maintenant, codé sur 48 bits, et G_{i-1} codé sur 32 bits. Donc, le résultat final de *f* doit être codé sur 32 bits au lieu de 48 bits. Pour cela, la chaine de $48 = 8 \times 6$ bits sera transformée en une chaîne de $32 = 8 \times 4$ bits en utilisant des dispositifs appelés *boites-S*. Elles sont au

nombre de huit, où chacune calcule un bloc de 4 bits à partir d'un bloc de 6 bits. Enfin, on applique une permutation à ce 32 bits pour obtenir la valeur finale de *f* (voir la Figure I.7).
L'ensemble de ces résultats en sortie de P est soumis à un « ou exclusif » avec le G_0 de départ (comme il est indiqué dans la Figure I.9) pour donner D_1, tandis que le D_0 initial donne G_1, et ainsi de suite pour les 16 itérations comme il est mentionné auparavant.
La succession d'opérations constituant la fonction ***f*** est représentée à travers la Figure I.8**.**

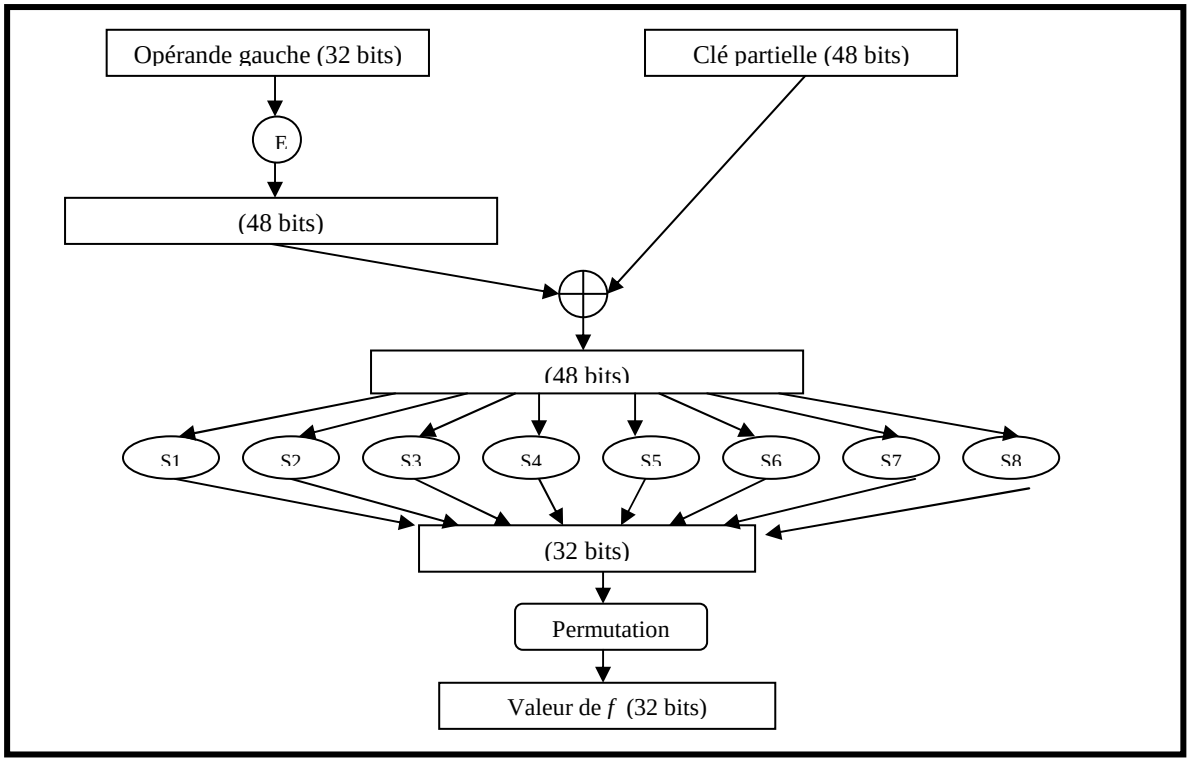


Figure I.8. *Schéma de la fonction f.*

4. **Permutation initiale inverse :** A la fin des itérations, les deux blocs G_{16} et D_{16} sont récoltés, puis soumis à la permutation initiale inverse : $Z = PI^{-1}(G_{16}\ D_{16})$. Le résultat en sortie est un texte codé de 64 bits.
   Cette description peut être résumée par le schéma général illustré par Figure I.9, où on a seulement représenté quelques-unes des 16 étapes.

Remarque : le déchiffrement suit le même algorithme avec la même clef *K*. Seules les sous-clés sont appliquées dans le sens inverse en commençant par la clé k_{16} jusqu'à la clé k_1.

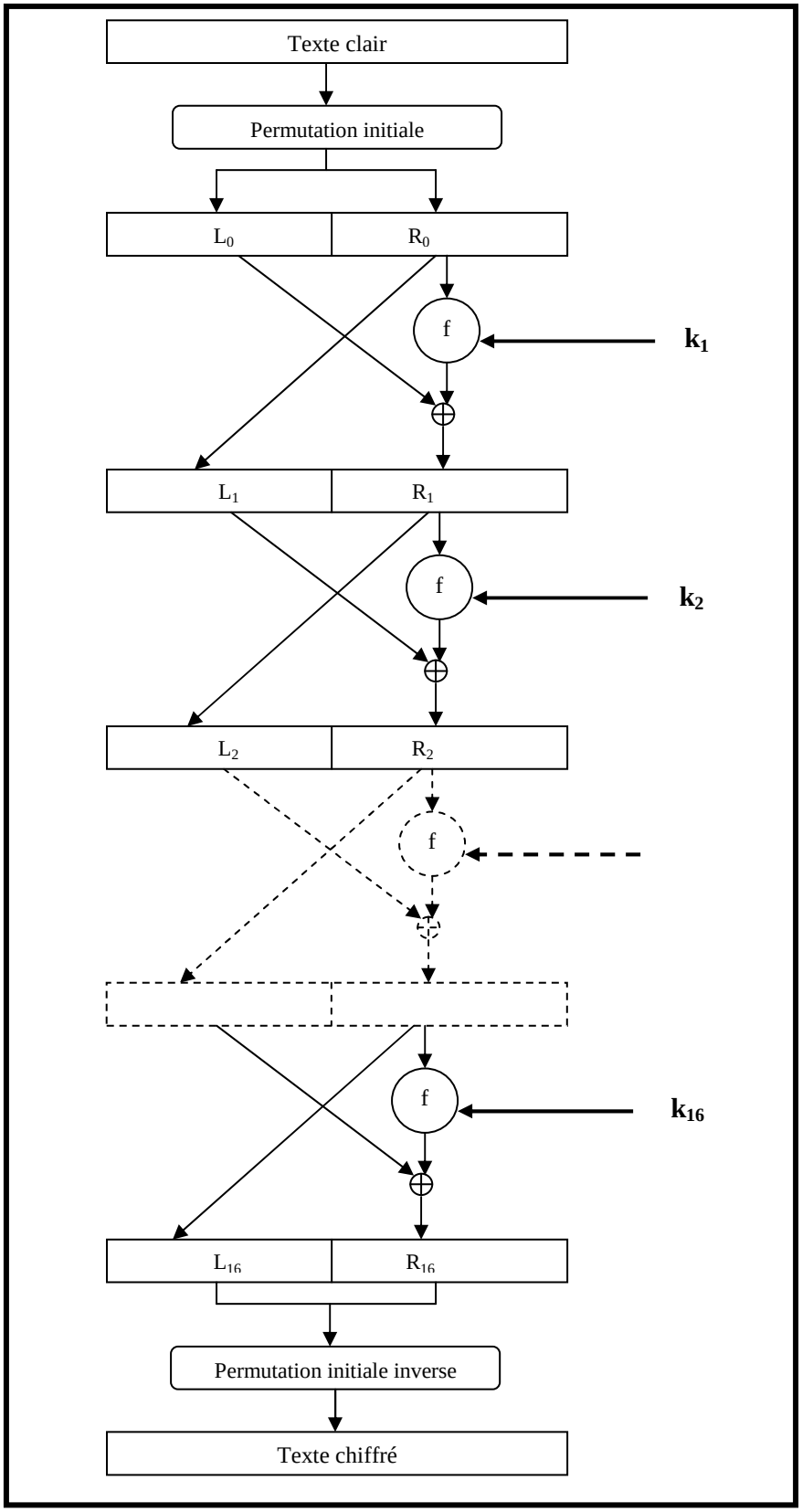


Figure I.9. *Schéma général de DES.*

Le DES a fait l'objet de très nombreuses attaques. On peut citer quelques une :

- **Cryptanalyse différentielle :** c'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires. Ainsi, un DES à 8 ou à 10 tours peut facilement être cassé, mais le DES complet à 16 tours est resté hors de portée de cette attaque.

- **Cryptanalyse linéaire :** c'est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Elle n'est utilisable que pour un DES restreint à quelques tours, mais le DES réel n'est pas menacé par cette attaque.

- **Recherche exhaustive :** les laboratoires RSA ont lancé en Janvier 1997 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES pour démontrer que la taille des clés DES a devenu insuffisante. Ils ont réussi le 17 Juin 1997. Bien qu'une telle recherche demande des moyens considérables, elle a montré que le DES n'offre plus aujourd'hui une grande sécurité.

3) Triple DES (3DES)

Pour palier l'insuffisance cryptographique observée du cryptosystème DES, dû à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. Triple DES est apparait comme une solution pour remédier les faiblesses de DES. Comme son nom l'indique, le principe du triple DES est d'effectuer 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Ce principe peut être formulé comme suit :

$$\textbf{Triple-DES}_{k1,\,k2} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}^{-1}_{k2} \;\textbf{O}\; \textbf{DES}_{k1}$$

On peut constater que le DES sera retrouvé comme cas particulier de la formule ci-dessus, lorsque $k_1 = k_2$. Le déchiffrement de son tour est formulé par :

$$\textbf{Triple-DES}^{-1}_{k1,\,k2} = \textbf{DES}^{-1}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}^{-1}_{k1}$$

Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc composée de 112 bits. Une autre variante à trois clés DES différentes peut être conçue. D'une façon plus formelle, son principe peut être donné par la formule suivante :

$$\textbf{Triple-DES}_{k1,\,k2,\,k3} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}_{k3}$$

Malgré cela, cette variante reste aussi fragile à une attaque de coût en 2^{112} s'appuyant sur l'un des deux messages intermédiaires.

4) AES (**Advanced Encryption Standard**)

Avec le temps et les progrès de l'informatique, les 2^{56} clés possibles du DES se voient incapables de fournir une barrière infranchissable dû à la faiblesse des clés de 56 bits. Désormais avec des moyens modestes, percer les messages chiffrés par DES en un temps raisonnable, ne pose aucun problème. De ce fait, l'AES a vu le jour. Il est issu d'un appel à candidatures international lancé en janvier 1997 et ayant reçu 15 propositions. Au bout de cette évaluation, ce fut le candidat Rijndael du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen [www4] qui a été choisi par le NIST en Octobre 2000 pour être l'algorithme AES et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. II a été déclaré vainqueur de la deuxième ronde dans laquelle

s'opposaient les 5 candidats finalistes (MARS, RC6, Rijndael, Serpent, TwoFish). L'AES a une taille longue de sa clé allant jusqu'à 256 bits qu'il le permet de résister toujours a la cryptanalyse. L'AES est devenu le nouveau standard du chiffrement symétrique comme le signifie cet acronyme. L'AES est libre d'utilisation comme l'est l'algorithme DES.

Techniquement, le chiffrement AES travaille avec des blocs de 128 bits seulement et la longueur des clés utilisées peut être de 128, 192 ou de 256 bits. Chaque bloc subit une séquence de transformations que nous résumons à travers les points suivants :

1. Addition de la clé secrète et du bloc en question avec un « ou exclusif ».
2. ByteSub : les 128 bits sont répartis en 16 blocs de 8 bits (16 octets), qui sont ensuite placés dans une matrice de 4×4. Chaque octet est transformé par une fonction non linéaire S (S-box) conçu pour résister à la cryptanalyse linéaire et différentielle.
3. ShiftRow: les lignes de cette matrice sont soumises à une rotation vers la droite où l'incrément pour la rotation varie selon le numéro de la ligne. La $2^{ème}$ ligne est décalée d'une colonne, la $3^{ème}$ ligne de 2 colonnes, et la $4^{ème}$ ligne de 3 colonnes.
4. MixColumn: chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne. Cela revient à multiplier la matrice 4×4 par une autre matrice 4×4.
5. AddRoundKey: une clé dite *de tour* est générée à partir de la clé secrète par un sous-algorithme dit *de cadencement*. Cette clé de tour est ajoutée par un « ou exclusif » au dernier bloc obtenu.

Ces différentes opérations, définissant un *tour*, sont répétées plusieurs fois. Cependant dans le dernier tour il n'y a pas d'opération MixColumn. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours. La figure suivante résume le principe de fonctionnement de cet algorithme de chiffrement.

Le déchiffrement consiste en appliquer les opérations inverses dans chacune des étapes (*InvSubBytes*, *InvShiftRows*, *InvMixColumns*). *AddRoundKey* (à cause du XOR) est son propre inverse. On réitère ce processus le nombre de tour-1. Pour le dernier tour on exclu l'opération *InvMixColumns* ; et comme dans le chiffrement pas d'opération *InvMixColumns* dans le dernier tour.

De nombreux travaux de cryptanalyse ont été publiés mais pour l'instant il n'a pas été cassé et la recherche exhaustive demeure la seule solution. En particulier, on cite :

- **Attaques sur des versions simplifiées :** *Niels Ferguson* et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours. En effet, cette dernière attaque repose sur le principe des clés apparentées. Dans une telle attaque, la clé demeure secrète mais l'attaquant peut spécifier des transformations sur la clé et chiffrer des textes à sa guise. Il peut donc légèrement modifier la clé et regarder comment la sortie de l'AES se comporte.

- **Attaques sur la version complète :** plusieurs chercheurs ont mis en évidence des possibilités d'attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL. Le XSL fait appel à une analyse heuristique dont la réussite n'est pas systématique. Elles sont impraticables car le XSL demande au moins 2^{87} opérations voire 2^{100} dans certains cas. Le principe est d'établir les équations (quadratiques / booléennes) qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer et l'AES est toujours considéré comme sûr.

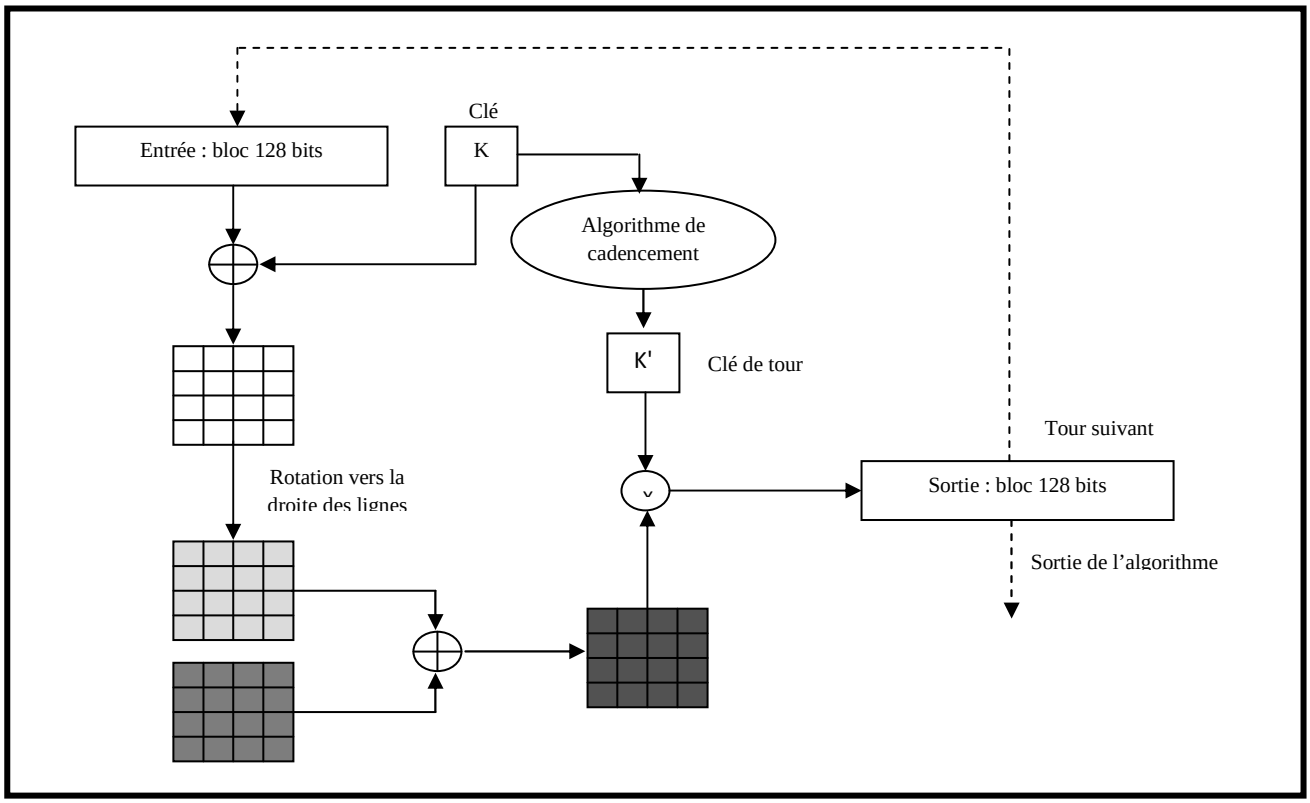


Figure I.10. *Schéma général de l'AES.*

5) IDEA (International Data Encryption Algorithm)

L'**IDEA** qui est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 rondes inventé en 1990 à Zürich par James L. Massey et Xuejia Lai. C'est la taille des clés qui apporte une grande sécurité au système (que l'on n'a pas, à l'heure actuelle, officiellement réussi à "casser").

Chaque bloc est divisé en 4 sous-blocs de 16 bits : X_1, X_2, X_3 et X_4. Ces quatre sous-blocs deviennent les entrées de la première ronde de l'algorithme.
À chaque ronde, la séquence d'évènements est la suivante (voir Figure I.11) :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé ;
5. combinez par OU exclusif les résultats des étapes (1) et (3) ;
6. combinez par OU exclusif les résultats des étapes (2) et (4) ;
7. multipliez le résultat de l'étape (5) avec la cinquième sous-clé ;
8. additionnez les résultats des étapes (6) et (7) ;
9. multipliez le résultat de l'étape (8) par la sixième sous-clé ;
10. additionnez les résultats des étapes (7) et (9) ;
11. combinez par OU exclusif les résultats des étapes (1) et (9) ;
12. combinez par OU exclusif les résultats des étapes (3) et (9) ;
13. combinez par OU exclusif les résultats des étapes (2) et (10) ;
14. combinez par OU exclusif les résultats des étapes (4) et (10).

La sortie de la ronde est constituée des 4 sous-blocs produits par les étapes (11), (13), (12) et (14). Changez les deux blocs intérieurs (sauf lors de la dernière ronde) et cela donne l'entrée de la ronde suivante.
Après la huitième ronde, il y a une transformation finale :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé.

Enfin les 4 sous-blocs sont réassemblés pour former le texte chiffré.

Chaque ronde utilise 6 sous-clés K_i^r de 16 bits : K_1^1, ..., K_6^1, …, K_1^8, …, K_6^8. La transformation finale utilise 4 sous-clés : $K_1^9, K_2^9, K_3^9, K_4^9$; donc un total de 52 sous-clés est utilisé, les premiers 8 sous-clés sont extraites directement à partir de clé, des groupes de 8 clés sont extraits par rotation à gauche de 25 bits de la clé K, ce qui donne 6 rotations en total.
Concernant le processus de déchiffrement on utilise celui de chiffrement avec un seul changement dans la génération des clés. En fait, on utilise K pour générer les sous clés K_i^r ; à partir de ces derniers, d'autres clés $K_i^{'r}$ sont obtenues dans le Tableau I.1 [Omar, 2006] ; ensuite on utilise $K_i^{'r}$ à la place des K_i^r dans l'algorithme de chiffrement IDEA. Dans le Tableau I.1, $-K_i$ dénote l'opposé modulo 2^{16} de K_i. K_i^{-1} dénote l'inverse multiplicative de $K_i \bmod(2^{16}+1)$, se trouvant aussi dans {0, 1, …,216-1}.

Ronde r	$K_1^{'(r)}$	$K_2^{'(r)}$	$K_3^{'(r)}$	$K_4^{'(r)}$	$K_5^{'(r)}$	$K_6^{'(r)}$
r =1	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
2 ≤ r ≤ 8	$(K_1^{10-r})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
r= 9	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	--	--

Tableau I.1. *Les sous clés de déchiffrement $K_i^{'r}$ générées à partir des sous clés K_i^r.*

La méthode de génération des sous-clés de l'IDEA est toujours régulière et donc pourrait être une faiblesse à l'algorithme. Cependant, il est considéré comme étant hautement sécuritaire. En effet, pour résoudre IDEA avec l'attaque en force brute, il faudrait effectuer 2^{128}, donc 10^{38} opérations [www5].

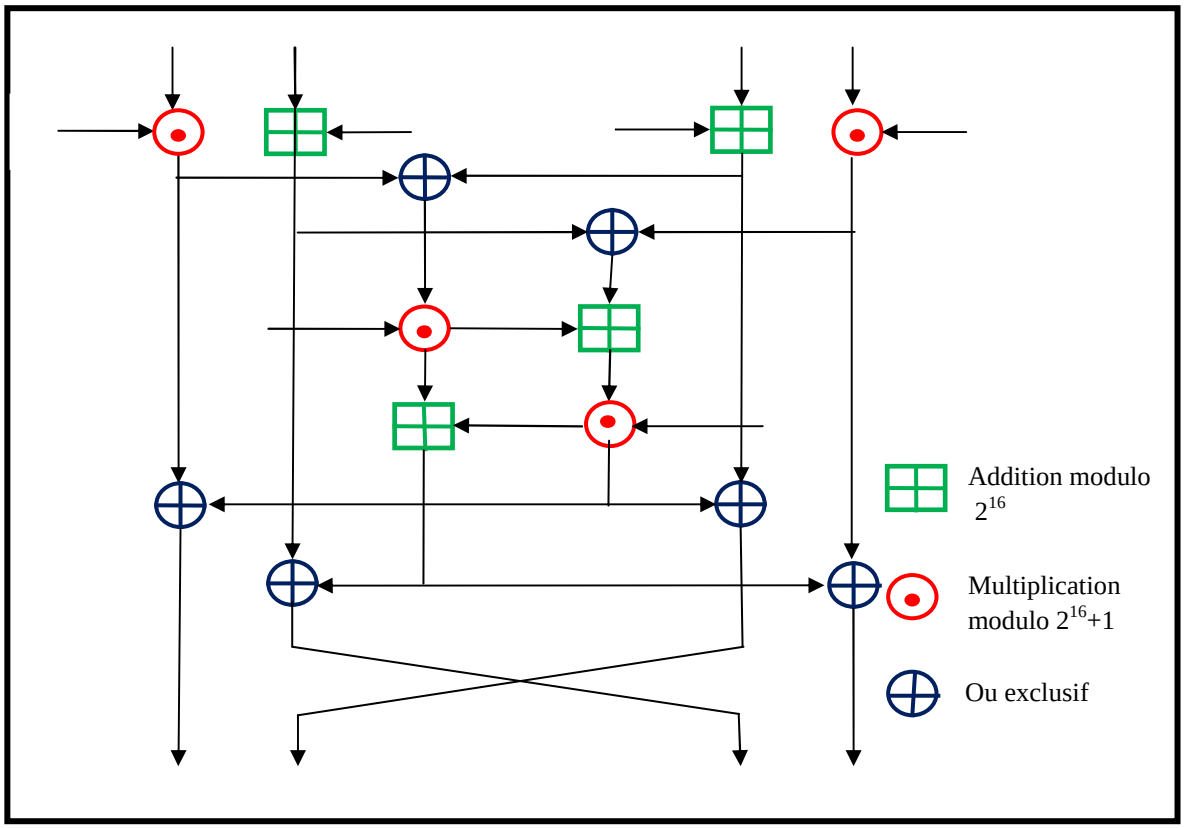


Figure I.11. *Schéma général de l' IDEA.*

- **Conclusion**

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. AES est immunisé contre les attaques et il reste néanmoins le meilleur choix dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause. D'autre part, IDEA est considéré par les spécialistes comme l'un des meilleurs cryptosystèmes à clé privée. Il est utilisé par le PGP pour le chiffrement de données.

b. La cryptographie asymétrique

- **Principe**

La *cryptographie asymétrique* ou encore dite *à clé publique* utilise deux clés différentes pour chaque utilisateur : une est privée et n'est connue que de l'utilisateur et qui est sensé d'être en mesure de faire signature ou déchiffrement. L'autre est publique diffusée en général dans un annuaire et donc accessible par quiconque afin de permettre aux interlocuteurs de mettre en œuvre les opérations réciproques (vérification de signature ou chiffrement de message). Les clés, publique et privée sont mathématiquement liées par l'algorithme de cryptage de telle manière qu'un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de *fonction à sens unique avec trappe*. Sachons qu'une fonction est appelée à *sens unique* si elle est

aisément calculée, mais extrêmement difficile de déduire la fonction inverse. Et elle sera dite *à trappe*, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la *trappe*. L'utilité d'utilisation d'une telle fonction réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas assez facile puisqu'elles s'appuient généralement sur des problèmes mathématiques réputés difficiles.

Ce cryptage présente l'avantage de permettre le placement de signature numérique dans le message et ainsi permettre l'authentification de l'émetteur grâce à la fonction de hachage. Le principal avantage consiste à résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé puisque la clé privée n'est connue que par l'utilisateur. Bien que plus lent que la plupart des cryptages à clé privée il reste toujours préférable pour 3 raisons :

- ✓ plus évolutif pour les systèmes possédant des millions d'utilisateurs.
- ✓ authentification plus flexible.
- ✓ supporte les signatures numériques.

Les systèmes asymétriques les plus connus sont [Mene, 1996] : RSA basé sur la difficulté de la factorisation des grands entiers, El Gamal basé sur la difficulté de résoudre le problème du logarithme discret dans un corps fini et les systèmes sur les courbes elliptiques basés sur la difficulté de certains calculs sur les courbes elliptiques.

- **Quelques algorithmes asymétriques**

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivants :

1) RSA

La méthode de cryptographie RSA a été inventée en 1977. Elle titre son nom des noms de ses trois inventeurs: *R. Rivest, A. Shamir* et *L. Adleman* [www6]. Le RSA est le premier et est encore le système cryptographique à clé publique le plus utilisé de nos jours et toujours considéré comme sûr. Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers ; à l'heure actuelle, il est pratiquement impossible de les reconstituer en un temps raisonnable. Ainsi, la sécurité de RSA semble satisfaisante malgré qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser.

On peut résumer le fonctionnement de ce cryptosystème dans les étapes suivantes :

- **Fonction d'encodage E (publique) :** la clé publique k utilisée pour l'encodage comporte deux entiers: k = (e,n). L'opération de chiffrement se fait au moyen de l'élévation à la puissance **e** modulo n : $E_k(M) = M^e \bmod n$.
- **Fonction de décodage D (privée) :** la clé secrète k′ utilisée pour le déchiffrement est aussi un couple d'entiers : k′(d,n). Pour reconstituer le message initial, la fonction inverse d'encodage est appelée, elle est ainsi : $D_{k'}(M) = M^d \bmod n$.
- **Détermination des clés :**
  - **Détermination de n :** pour calculer n on doit initialement choisir deux entiers premiers *p* et *q* très grands et leurs valeurs sont secrètes connus que part l'utilisateur. Le choix de *p* et *q* affecte grandement le niveau de sécurité de RSA. Pour cela, il faut évidemment se prémunir contre les algorithmes de factorisation dont la complexité dépend essentiellement de la taille du plus petit facteur premier de *n* [Zimm, 2005], donc si possible choisir *p* et *q* de même taille.

- **Détermination de e :** pour calculer e, on calcule premièrement un entier z = (p-1)*(q-1) puis tout simplement choisir un entier e premier avec z.
- **Détermination de d :** pour calculer d, on procède ainsi : e*d≡1[z].

La sécurité de RSA est basée sur l'hypothèse que la fonction E est à sens unique, ce qui rend impossible à décrypter un texte chiffré. Mais à l'aide de la trappe qui est la factorisation n = p*q, il est possible d'en trouver d et donc la clé privée.

Depuis son apparition, plusieurs attaques ont été découvertes contre le RSA. Et même si aucune de ces attaques n'est réellement destructive, elles démontrent toutefois qu'il faut implémenter RSA avec beaucoup de précautions. La fameuse attaque est :

- **Recherche exhaustive :** comme la fonction de chiffrement RSA est déterministe, si l'ensemble des messages possibles est connu et de petite taille, il sera facile de décrypter par une recherche exhaustive [Ster, 2004]. Pour éviter cette attaque, il est indispensable de randomiser les messages avant chiffrement.

2) Chiffrement d'ElGamal

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste c.à.d. un même message M peut avoir plusieurs chiffres différents [Lafo, 2006]. Il est basé sur la difficulté du problème des logarithmes discrets c.à.d. la difficulté de trouver l'unique *a* noté $\log_\alpha ß$ / $0 \leq a \leq p-2$ tel que : $\alpha^a \equiv ß \pmod p$ où *p* est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$. Cependant, et pour éviter les attaques connues, *p* doit être convenablement choisi, et $p-1$ doit avoir un grand facteur premier.

D'une manière formelle, l'algorithme ElGamal peut être résumé comme suit :

Soit *p* un nombre premier tel que le problème du logarithme discret dans Z_p soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et $K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod p\}$
Les valeurs *p*, α et ß sont publiques, et *a* est secret.
Pour k = (*p*, α, *a*, ß), et pour un $k \in Z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$\mathbf{e_k(x, k) = (y_1, y_2)}$$

Où

$$\mathbf{y_1 = \alpha^k \bmod p}$$

Et

$$\mathbf{y_2 = x\, ß^k \bmod p}$$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit comme suit :

$$\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$$

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par $ß^k$, en produisant y_2 ;
- ✓ la valeur α^k est également transmise en tant que partie du texte chiffré ;
- ✓ Bob, qui connaît l'exposant secret *a*, peut calculer $ß^k$ à partir de α^k. Il peut alors « enlever le masque » en divisant y_2 par $ß^k$ et obtenir le texte clair x.

Une attaque possible à ce cryptosystème est celle dite ***man in the middle***. Son principe dans le cas d'ElGamal fonctionnant avec le mode *Diffie-hellman* est illustré sur la figure I.12.

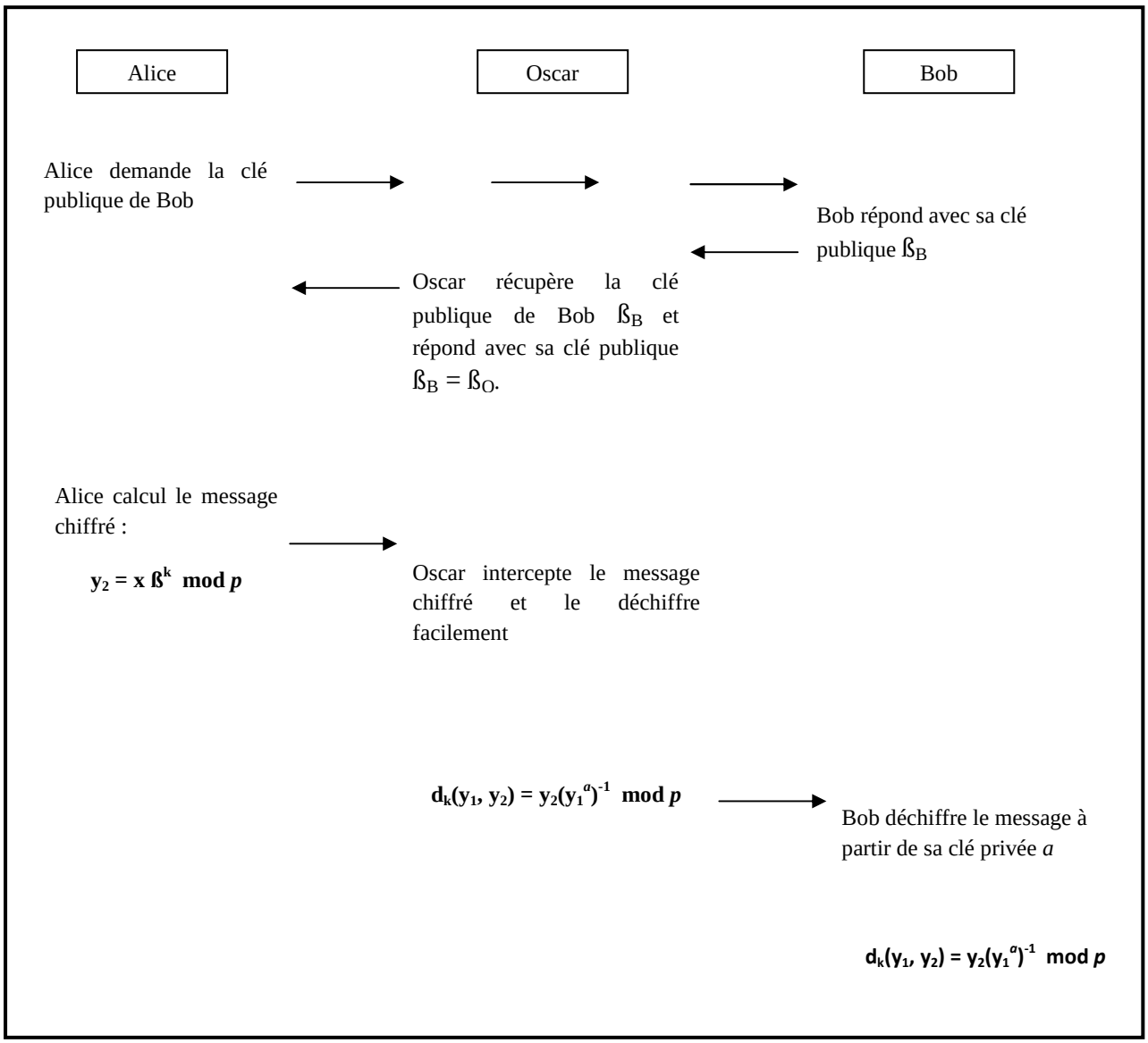


Figure I.12. *Principe de l'attaque « man in the middle ».*

- **Conclusion**

Les calculs faits en 1995 ont ouvert un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent des chiffres plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique serait capable de remettre en cause ce cryptosystème.

De même, casser l'algorithme ElGamal est aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Et pour assurer une bonne sécurisation de cet algorithme, Zimmermann en 2005 [Zimm, 2005] a recommandé l'utilisation des clés d'au moins 1024 bits.

c. Comparaison entre les cryptosystèmes symétriques et asymétriques

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage.

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ?
À partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que l'importance de la taille de clé n'est légitime que dans le cas de la clé secrète, puisque les seules attaques possibles sont les attaques exhaustives [www7]. Mais, dans le cas de la clé publique, la taille de la clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

Méthode	Avantages	Inconvénients
À clefs Secrètes	▪ Rapidité de calcul en général (dépend de la taille de la clé). ▪ Adaptée au cryptage de flux de données.	▪ Moins sécurisé (DES). ▪ Problème de communication de clefs entre émetteur et récepteur. ▪ Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clefs Publiques	▪ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre. ▪ Permet la signature électronique. ▪ Un couple de clés publique/privée suffisant pour 'n' correspondants.	▪ Lente. ▪ Problèmes de gestion de clefs publiques.

Tableau I.2. *Comparaison entre les méthodes de chiffrement symétriques et asymétriques.*

d. La cryptographie hybride

- **Principe**

La cryptographie asymétrique est beaucoup plus lente que la cryptographie symétrique qui brille par sa rapidité. En revanche, cette dernière souffre d'une grave lacune ; assurer une transmission secrète de la clé. Pour palier ce défaut et cumuler les avantages des deux méthodes on a fait recourt à la cryptographie hybride. On code tout d'abord les données avec une clé privée dite *clé de session*, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Comme la clé est courte, on utilise l'algorithme asymétrique puisqu'il prend peu de temps. En revanche, chiffrer l'ensemble du message avec un algorithme asymétrique serait plus lourd. Il suffit ensuite d'envoyer le message chiffré avec une clé privée et accompagné de cette dernière chiffrée avec une clé publique. Le destinataire procède inversement, il commence à déchiffrer la clé symétrique avec sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, via un déchiffrement symétrique pour retrouver le message original. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

- **Quelques algorithme hybride**

1) PGP (**Pretty Good Privacy**)

PGP est inventé par *Philip Zimmermann*, qui dise que tout individu a droit à la confidentialité, notamment les organisations des droits de l'homme dans des pays soumis à la dictature. Il l'a mis à disposition gratuitement sur Internet. Cela lui a valu de sérieux ennuis avec la justice américaine, car les logiciels de cryptage sont considérés comme du matériel de guerre et sont interdits à l'exportation.

Il est souvent utilisé pour garantir l'authentification et le contrôle d'intégrité des fichiers et du courrier électronique. Mais avant de crypter un texte avec PGP, les données doivent tout d'abord compressées dont le but de réduire le temps de transmission, ainsi d'économiser l'espace disque et, surtout, de renforcer la sécurité cryptographique puisque les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement alors que la compression réduit ces modèles dans le texte en clair.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes qui sont [www8] :

✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
✓ PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de déchiffrement se fait également en deux étapes, qui sont [www8] :

✓ PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
✓ PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

Depuis 1978, la recherche universitaire civile a intensément attaqué la cryptographie à clé publique, sans pour autant réussir à la remettre en cause. Mais cela ne fournisse aucune garantie totale sur la sécurité de cette manière de chiffrer, car une attaque menée par le gouvernement, par exemple, qui ne se manque pas de ressources très développées ou même en utilisant quelques nouvelles percées mathématiques classées top-secret, peut tenir à bout ces cryptosystèmes conventionnels utilisés dans PGP.

Tout de même, l'optimisme semble justifié. Les algorithmes de clé publique, les algorithmes de contraction de message, et les chiffres par blocs utilisés dans PGP ont été conçus par les meilleurs cryptographes du monde [Loid, 2005]. Les chiffres de PGP ont subi des analyses de sécurité approfondies et des examens méticuleux de la part des meilleurs cryptographes dans le monde non classé top secret. De plus, et même si les chiffres par blocs utilisés dans PGP possèdent quelques faiblesses, la compression du texte clair utilisée avant le chiffrement réduit de façon considérable ces faiblesses.

2) GPG (**GNU Privacy Guard**)

GPG est la version GNU de PGP. En raison qu'il est sous licence GPL (General Public License) il permet la fourniture du code-source, la libre modification et la libre redistribution de ce code-source. Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

- **Conclusion**

Aujourd'hui, de nombreux gouvernements ont restreint l'usage du PGP, qui a été largement diffusé par son développeur, en pensant qu'un cryptage trop fiable ferait le jeu des terroristes et des trafiquants. Toutefois, il y'a pas mal d'applications qui utilisent encore ce système de chiffrement, telles que les paiements en ligne qui se font grâce au procédé SSL

fonctionnant selon le principe du PGP. De sa part, GPG, qui est un cryptosystème utilisant des algorithmes de chiffrement à clé publiques (DSA, RSA et ElGamal), est largement utilisé dans les communications par messagerie, c.-à-d. pour chiffrer des mails, ainsi que pour signer des données.

I.3.3.3. Cryptographie quantique

a. Principe

La cryptographie à base d'algorithmes aura toujours des faiblesses. Même si la cryptanalyse bloque devant un algorithme, la force brute pourra toujours décrypter n'importe quel code si on lui donne assez de temps. Même avec le plus solide des chiffrages, le contenu du message peut être subtilisé et dupliqué. Les clés, quant à elles, peuvent être volées en chemin ou présenter des faiblesses qui les rendent prévisibles.

Si maintenant on base notre cryptographie, non pas sur un algorithme mathématique, mais sur des lois de la physique quantique, il n'y plus de force brute qui peut briser un code ici. Le cryptage ne se trouve pas dans des formules, mais dans des photons, ainsi tout le monde peut accéder à des formules, mais pas tout le monde peut accéder aux photons sans que le message devienne inutile. De ce fait, l'information n'est plus sécurisée par des subterfuges mathématiques, mais plutôt par des lois fondamentales de physique. Au de-là, la cryptographie quantique a vu le jour.

La figure I.13 [Nave, 2002] présente un exemple relatif à la possibilité soit qu'un photon traverse le filtre ou pas, selon l'orientation de sa polarisation. Sachons qu'un filtre permet de distinguer entre les photons polarisés horizontalement (0°) et verticalement (90°) ; un autre entre les photons polarisés en diagonale (45°, 135°).

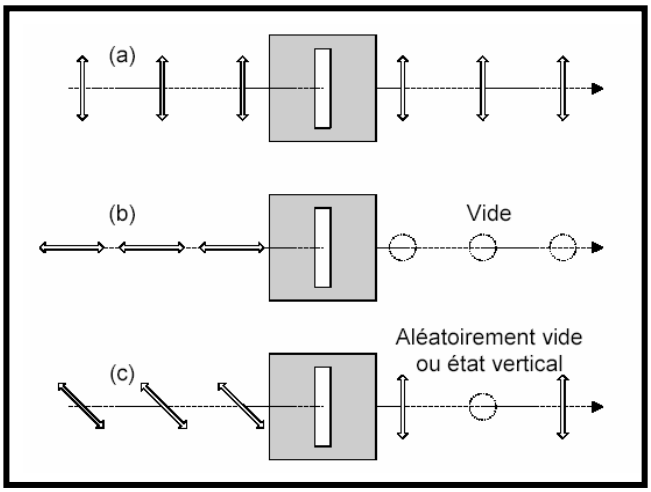


Figure I.13. *Photon unique traversant un filtre ne laissant passer que la lumière polarisée verticalement :*
(a) *Les états polarisés verticalement traversent le filtre sans être absorbés*
(b) *Les états polarisés horizontalement sont tous absorbés*
(c)*Les états polarisés diagonalement sont aléatoirement absorbés ou transmis*

Pour mieux comprendre ce type de cryptographie, on va l'aborder à partir d'un exemple dont le protocole est bien : le protocole BB84. Avant de commencer, on doit mentionner qu'il existe deux modes de polarisations possibles [www9] :

- **Mode 1 :** "0" est codé par un photon d'axe de polarisation 0° et "1" par un photon de polarisation 90°.
- **Mode 2 :** "0" est codé par un photon d'axe de polarisation 45° et "1" par un photon de polarisation 135°.

Pour plus de clarté, ces deux modes peuvent être schématisés comme suit :

Figure I.14. *Les deux modes de polarisation.*

Cet exemple se déroule en plusieurs étapes que l'on va présenter ci-dessous. On a Alice l'expéditrice, Bob le destinataire et Oscar l'éventuel espion [www9]:

1. Alice génère aléatoirement un bit selon un mode (mode 1 ou mode 2) choisi lui aussi aléatoirement, ce que la physique quantique permet, et transmet le photon obtenu à Bob. Elle répète cette opération autant de fois que nécessaire.
2. Pour chaque photon reçu, Bob choisit aléatoirement un mode, c'est-à-dire un polariseur, et note si le photon traverse le polariseur.
3. Ensuite une phase de réconciliation : Alice et Bob communiquent entre eux et pour chaque photon ils comparent si leurs modes coïncident, si c'est le cas, ils ont le même bit, qu'ils conservent, sinon ils ne conservent pas le bit en question. Vue le nombre de photons inutiles, il est donc nécessaire de prévoir un excédent suffisant de photons pour obtenir une clé de longueur donnée. Comme Bob a une chance sur deux de choisir le bon mode, en moyenne on observe ***N*** erreurs pour ***2N*** photons envoyés.
4. Enfin, Alice et Bob contrôlent la sûreté de la clé : parmi les bits conservés ils en choisissent un certain nombre qu'ils comparent publiquement, s'ils ont été espionnés ils obtiendront en moyenne 25% de bits différents (Comme Oscar ne peut pas cloner les photons, il est dans l'obligation de les intercepter, de les mesurer et de renvoyer à Bob un photon similaire à celui qu'il a mesuré. Cependant il a une chance sur deux de choisir le mauvais mode et donc de renvoyer un photon différent de celui qu'Alice avait envoyé à Bob. Puis comme Bob a lui aussi une chance sur deux d'avoir le bon mode, en cas d'espionnage pour chaque photon on a une chance sur 4 (25%) de détecter l'intrusion). Dans ce cas, ils n'utiliseront pas la clé. S'ils n'ont pas de différences, alors ils peuvent conserver la clé pour l'utiliser ultérieurement.

Alice et Bob décident alors de sacrifier une partie de leur clé commune et les comparent publiquement par le canal radio (comme exemple de canal). Dans le Tableau I.3 ils ont quatre bits différents c.à.d. N bits parmi 2N. Ainsi la clé obtenue : 0011.

Photon envoyé par Alice								
Mode choisi par Bob	Mode 1	Mode 2	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								
Après réconciliation								

Tableau I.3. *Exemple sans Oscar.*

Photon envoyé par Alice								
Mode choisi par Oscar	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 2	Mode 2
Résultat mesuré et renvoyée à Bob par Oscar								
Mode choisi par Bob	Mode 1	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								

Tableau I.4. *Exemple avec Oscar.*

Plusieurs cas de figure se présentent à nous dans le Tableau I.4 [Camp, 2010] :

1. Oscar et Bob ont tous les deux le bon mode, alors Oscar mesure bien la bonne polarisation qu'il renvoie à Bob et puis de même pour Bob. Le bit est valable pour la clé et Oscar est passé inaperçue. Exemple : premier photon envoyé.
2. Oscar a le bon mode, mais pas Bob, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon. Oscar passe encore inaperçue, mais sa bonne mesure lui est inutile. Exemple : le deuxième photon envoyé.
3. Oscar a le mauvais mode, mais Bob a le bon mode, alors d'après ce que l'on a vu, Bob a une chance sur deux d'obtenir une mesure compatible avec ce qu'à envoyer Alice. Donc Oscar a une chance sur deux d'être détecté, de plus sa mesure est inutile. Exemple : sixième photon envoyé (avec détection) et dernier photon envoyé (sans détection).
4. Oscar et Bob ont le mauvais mode, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon, Oscar passe donc inaperçue. Exemple : cinquième photon envoyé.

Une fois la réconciliation terminée il ne reste que les bits de la possibilité 1 et 3 qui sont équiprobables. Pour la première possibilité Oscar passe inaperçue, et pour la troisième il a une chance sur deux de se faire détecter. On retrouve bien le 25% ($\frac{1}{2}\frac{1}{2} = \frac{1}{4}$) vu ci-dessus.

b. Conclusion

Le MIT (Massachussets Institute of Technology) a récemment classé la cryptographie quantique parmi les «10 inventions qui changeront le monde». Si on voit peu les avantages à l'heure actuelle, les possibilités pour le futur sont impressionnantes. Le centre canadien IRCA est un leader mondial en recherche quantique, comptant parmi ses membres des chercheurs montréalais au front de cette révolution du monde des télécommunications. Avec le développement du calcul et de la cryptographie quantique, on peut espérer qu'Alice et Bob pourront un jour communiquer en toute quiétude. Entre temps, considérons d'un œil sceptique les soi-disant algorithmes « incassables » qui n'attendent que le prochain superordinateur pour céder comme une coquille d'œuf [Camp, 2010].

I.3.3.4. Algorithme de chiffrement évolutionniste OTL

Omary Fouzia a développé en 2006 un nouvel algorithme de chiffrement en le simulant à un problème d'optimisation dont la résolution est par les algorithmes génétiques. Le but de cet algorithme est de modifier au maximum les fréquences d'apparition des caractères dans le message à chiffrer et d'établir le plus de désordre dans leurs positions. Ces caractères appartiennent à l'ensemble des 256 caractères du code ASCII. L'application de l'algorithme est précédée d'une phase de brouillage du texte initial (M_0) qui peut combiner plusieurs méthodes simples comme les substitutions, les permutations, le chiffrement affiné, etc..., pour obtenir un texte initialement chiffré (M_0').

Pour ce faire, le codage adopté pour représenter les individus, qui sont les différents messages, est résumé à travers la figure ci-dessous :

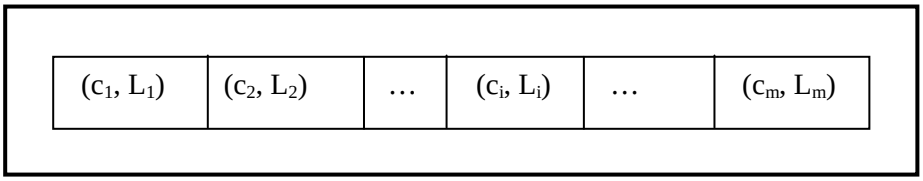


Figure I.15. *Codage des individus.*

Avec :

c_i : un caractère appartenant au message.
L_i : la liste des positions du caractère c_i.
m : le nombre des différents caractères du message.
$L_i \cap L_j = \emptyset, \ \forall i, j \in [1, m]$

L'algorithme OTL cherche à changer itérativement la répartition des listes L_i sur les différents caractères du message (sans modifier le contenu des listes) de telle manière que la différence entre le cardinal de la nouvelle liste affectée à chaque caractère c_i et le cardinal de la liste L_i d'origine soit maximale [Omar, 2006]. Cette variation, en terme de répartition, est assurée grâce aux opérateurs génétiques choisis (MPX pour le croisement avec un taux compris entre 60% et 100%, et une simple permutation pour la mutation avec un taux compris entre 0.1% et 5%). Et pour juger la pertinence des individus construits, la fonction

d'évaluation utilisée est celle donnée ci-dessous. Les meilleurs individus sont ensuite sélectionnés par le biais de la méthode classique de la roulette, en vue de se reproduire.

$$F(X_j) = \sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right|$$

Le processus évolutionnaire est répété jusqu'à la satisfaction d'un critère d'arrêt exprimé à l'aide de la fonction F. Cette dernière est bornée car : $0 \le F(X) \le 2 * l$ (l est la taille du message) pour tout individu X. En fait [Omar, 2006] :

$$\sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right| \le \sum_{i=1}^{m} (card(L_{j_i}) + card(L_i)) \le 2 * l$$

L'opération de déchiffrement, quand à elle, se fait en deux étapes :

1) Tout d'abord, le texte chiffré est représenté suivant le codage proposé en une suite de listes de positions, et c'est grâce à la clé génétique que les caractères vont retrouver leurs listes de position correspondantes dans le message en clair. En effet, la clé, qui peut être d'un usage symétrique ou asymétrique, est une permutation de {1,2,…, m}. Comme résultat, nous obtenons le message M_0'.
2) La deuxième étape, correspond au déchiffrement du message M_0' pour obtenir M_0.

I.4. Conclusion

Dans ce chapitre, nous avons présenté les différentes catégories de cryptographie depuis sa première apparition jusqu'à nos jours. Un état de l'art aussi riche que possible sur les plus célèbres algorithmes de chiffrement, ainsi les fameuses ruses des cryptanalystes ont été exposées. D'après cette étude, un système cryptographique est considéré comme sûr si personne n'a encore mis en défaut sa sécurité. Nous avons vu que ni la cryptographie classique ni symétrique n'a pu s'assurer ce besoin dû aux leurs inconvénients. Pour résoudre cela, les cryptographes ont cherché à déplacer la difficulté ; plutôt que d'utiliser de simples substitutions et de faire reposer la sécurité sur le nombre de clés possibles, ils ont essayé de faire reposer la sécurité sur des difficultés calculatoires. Cela a donné naissance aux algorithmes de cryptographie asymétrique. Cependant il s'avère que l'augmentation constante de la puissance de calcul de nos machines nécessite d'augmenter constamment la difficulté de nos algorithmes. Il se peut en effet qu'une nouvelle technologie anéantisse toute la difficulté d'un algorithme. La cryptographie hybride est aussi un sujet d'attaque puisqu'elle n'est qu'une combinaison de la cryptographie symétrique et asymétrique. Une nouvelle tendance basée sur la cryptographie quantique a aussi été développée. Cette cryptographie est sûre, néanmoins elle est basée sur des principes beaucoup plus théorique et lourds.

Enfin, à partir de la richesse et de la variété des modèles mathématiques exploités dans le domaine de la cryptographie, de nouvelles approches cryptographiques peuvent être proposées par exploitation de méthodes approchées. Le précédent chapitre présentera de telles méthodes de résolution de problèmes : les métaheuristiques.

CHAPITRE II

METAHEURISTIQUES

II.1. Introduction

Un très grand nombre de méthodes existent en RO et en IA pour résoudre différentes sortes de problèmes tels que les problèmes d'optimisation combinatoire. D'une manière très générale, les méthodes de résolution suivent quatre approches différentes pour la recherche d'une solution [JinK, 1999] : l'approche de construction, l'approche de relaxation, l'approche de voisinage et l'approche d'évolution. Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative : par exemple, la méthode de Lin et Kernighan qui resta longtemps le champion des algorithmes pour le problème du voyageur de commerce [LinS, 1973].

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques* [Reev, 1993], [Aart, 1997]. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [Lapo, 1996], [Osma, 1996]. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi, les métaheuristiques sont conçues pour résoudre des problèmes d'optimisation complexes où d'autres méthodes d'optimisation ne sont pas efficaces. Ces méthodes ont fini par être reconnues comme l'une des approches les plus pratiques pour résoudre des problèmes nombreux et complexes, et ceci est particulièrement vrai pour les divers problèmes du monde réel qui sont de nature combinatoire. L'avantage pratique de métaheuristiques réside dans leur efficacité et leur application générale. Dans la littérature et au début des recherches, des heuristiques spécialisées ont été généralement développées pour résoudre des problèmes complexes d'optimisation combinatoire.
D'autre part, avec l'émergence de stratégies des solutions plus générales, y compris les métaheuristiques telles que la recherche taboue, algorithmes génétiques, recuit simulé, le principal défi est devenu l'adaptation des méta-heuristiques à un problème particulier ou une catégorie de problème. Cela nécessite généralement beaucoup moins de travail que de développer une heuristique spécialisée pour une application spécifique, ce qui rend les métaheuristiques un choix attrayant pour la mise en œuvre dans les logiciels à usage général. En outre, une bonne mise en œuvre de métaheuristique est susceptible de fournir des solutions quasi optimales en temps de calcul raisonnables.

Ainsi, une métaheuristique est définie de manière similaire à une heuristique (qui est une méthode conçue pour un problème d'optimisation donné et qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème), mais à un niveau d'abstraction plus élevé (d'après E. Taillard).

II.2. Principes

Les métaheuristiques sont apparues dans les années 80. Ce sont des méthodes d'optimisation, de type stochastique, conçus pour les problèmes difficiles. Des progrès importants ont été réalisés avec l'apparition de nouvelles générations de méthodes approchées puissantes et générales, souvent appelées métaheuristiques [Reev, 1993] [Aart, 1997].

Le terme « métaheuristique » a été initialement utilisé par F. Glover pour distinguer la méthode tabou des heuristiques spécifiques [Glov, 1986]. Notons que ce terme est également utilisé par J-L. Laurière dans son système de résolution Alice [Laur, 1978].

L'origine du mot méta heuristique nous aide à comprendre sa signification. En grec: « Méta » signifie « au-delà », ou « au niveau supérieur », et « Heuristique» veut dire « Trouver ». Ainsi, l'interprétation de ces mots peut signifier que l'on effectue des recherches à un très haut niveau et que la procédure algorithmique regroupe plusieurs heuristiques.

D'après Glover, metaheuristique "se réfère à une stratégie maître qui guide et modifie d'autres heuristiques pour produire des solutions au-delà de ceux qui sont normalement générés dans une quête d'optimalité locale" [Glov, 1997].

Selon [Glov, 1997] « métaheuristiques dans leurs formes modernes sont basées sur une variété d'interprétations de ce qui constitue une recherche intelligente », où le terme de recherche intelligente a été mis en évidence par Pearl [Poire, 1984] (en ce qui concerne heuristiques dans un contexte de l'intelligence artificielle) et [vobs, 1993] (en ce qui concerne le contexte de la recherche opérationnelle). Dans ce sens, on peut aussi considérer la définition suivante: « Une métaheuristique est un processus à générations itératives qui guide

une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter les espaces de recherche en utilisant des stratégies lesarning pour structurer l'information afin de trouver des solutions quasi-optimales » [Osma, 1996]

Pour résumer, la définition suivante semble être la plus appropriée : « Une métaheuristique est un processus maître itératif qui guide et modifie les opérations d'heuristiques subordonnées pour produire efficacement des solutions de haute qualité. Il peut manipuler une solution unique complète (ou incomplète) ou un ensemble de solutions à chaque itération. Les heuristiques subordonnées peuvent être des procédures de niveau élevé (ou faible), ou une recherche locale simple, ou tout simplement une méthode de construction. La famille de métaheuristiques comprend, mais n'est pas limitée à, des procédures de mémoire d'adaptation, de recherche tabou, des systèmes de fourmis, avides de recherche randomisée d'adaptation, de recherche à voisinage variable, des méthodes évolutionnaires, des algorithmes génétiques, de recherche de points, des réseaux neuronaux, de recuit simulé, et leurs hybrides. " [VobS, 1999]

Le but des métaheuristiques est de minimiser ou de maximiser une, ou plusieurs fonctions objectives. Comme exemple, on peut demander de trouver le plus court chemin entre un point A et un point B, sachant que des obstacles sont présents donc il s'agit d'un problème de minimisation ; ou si on veut trouver comment effectuer le plus grand nombre de tâches en un temps limité alors c'est la maximisation du travail à faire.

L'évolution des métaheuristiques se fait de manière itérative. Ceci a l'avantage de permettre l'arrêt de l'algorithme quand on le souhaite, et de récupérer la meilleure solution trouvée jusqu'à présent sans être obligé d'attendre la fin de l'exécution. Un autre point important des métaheuristiques est qu'elles font évoluer des solutions en les améliorant à chaque itération.

Donc, les métaheuristiques sont généralement des algorithmes itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction. Les itérations successives permettent d'évoluer d'une solution peu satisfaisante à la solution la plus adaptée. Les métaheuristiques se comportent, donc, comme des algorithmes de recherche tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution.

Les métaheuristiques sont souvent inspirées de systèmes naturels, qu'ils soient pris en physique -cas du recuit simulé-, en biologie de l'évolution -cas des algorithmes génétiques- ou encore en éthologie -cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires.
Les métaheuristiques désignées par M sont, souvent, des algorithmes utilisant un échantillonnage probabiliste. Elles tentent de trouver l'optimum global (G) d'un problème d'optimisation difficile (avec des discontinuités (D), par exemple), sans être piégées par les optima locaux (L). Ceci est indiqué à la figure suivante :

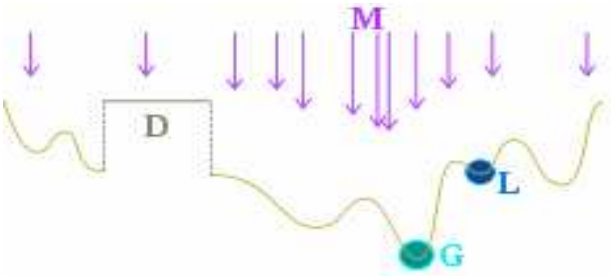


Figure II.1. *Principe général des métaheuristiques.*

II.3. Organisation d'une métaheuristique

D'une manière générale, les métaheuristiques s'articulent autour des principales notions suivantes :

II.3.1. Le Voisinage

Le voisinage d'une solution est un sous-ensemble de solutions qu'il est possible d'atteindre par une série de transformations données.
Formellement parlant [JinK, 1999], soit X l'ensemble des configurations admissibles d'un problème, on appelle *voisinage* toute application $N : X \to 2^X$. On appelle *mécanisme d'exploration* du voisinage toute procédure qui précise comment la recherche passe d'une configuration $s \in X$ à une configuration $s' \in N(s)$. Une configuration *s* est un *optimum local* par rapport au voisinage *N* si $f(s) \leq f(s')$ pour toute configuration $s' \in N(s)$.

Une méthode typique de voisinage débute avec une configuration initiale, et réalise ensuite un processus itératif qui consiste à remplacer la configuration courante par l'un de ses voisins en tenant compte de la fonction de coût. Ce processus s'arrête et retourne la meilleure configuration trouvée quand la condition d'arrêt est réalisée.

Un des avantages de cette stratégie de recherche réside précisément dans la possibilité de contrôler le temps de calcul : la qualité de la solution trouvée tend à s'améliorer progressivement au cours du temps et l'utilisateur est libre d'arrêter l'exécution au moment qu'il aura choisi. Dans la littérature, plusieurs voisinages ainsi que plusieurs stratégies de parcours de ces voisinages ont été définies [Gold, 1989], [Hert, 2003].

II.3.2. Diversification, intensification et apprentissage

La *diversification* (ou *exploration*, synonyme utilisé presque indifféremment dans la littérature des algorithmes évolutionnaires) désigne les processus visant à récolter de l'information sur le problème optimisé. L'*intensification* (ou *exploitation*) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Autrement dit, l'intensification insiste sur la capacité d'examiner en profondeur par une méthode des zones de recherche particulières alors que la diversification met en avant la capacité de découvrir des zones de recherche prometteuses.

La *mémoire* est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi les optima locaux.

Les métaheuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage, ou en mêlant ces notions de façon plus

étroites. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

Les notions d'intensification et de diversifications sont prépondérantes dans la conception des métaheuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherches. En effet, l'application systématique du seul principe d'exploitation ne permet pas une recherche efficace. En effet, l'exploitation conduit à confiner la recherche dans une zone limitée qui finit par s'épuiser. Le cas de l'amélioration itérative rapidement piégée dans un optimum local illustre cruellement ce phénomène. Une autre illustration souvent évoquée est fournie par le problème de convergence prématurée des algorithmes génétiques : du fait de la sélection, la population finit par n'être constituée que d'individus tous similaires. L'une des préoccupations majeures dans les algorithmes génétiques consiste d'ailleurs à préserver le plus longtemps possible une diversité suffisante dans la population. Face à ce type de difficulté, la solution consiste à diriger la poursuite de la recherche vers de nouvelles zones, *i.e.*, à recourir à l'exploration. Les deux notions ne sont, donc, pas contradictoires mais complémentaires, et il existe de nombreuses stratégies mêlant à la fois l'un et l'autre des aspects.

II.4. Méthodes générales et méthodes spécifiques

Contrairement aux algorithmes traditionnels dédiés à un problème spécifique, les métaheuristiques constituent des mécanismes très généraux qui peuvent être adaptés pour traiter de nombreux problèmes différents. Pour expliquer l'efficacité d'un algorithme spécifique, on invoque souvent le fait qu'il utilise des connaissances spécifiques du problème. Pour expliquer l'efficacité d'une métaheuristique, deux types d'arguments opposés sont généralement avancés.

Selon certains, des mécanismes généraux suffisamment puissants ont par eux mêmes la faculté de mener efficacement la recherche sans disposer d'information spécifique du problème considéré (contexte de boîte noire) : c'était notamment le point de vue classique porté sur les algorithmes génétiques [Gold, 1989]. Malheureusement, de même qu'il ne peut pas exister de stratégie avantageuse dans un jeu de hasard comme la roulette, il existe également des limitations théoriques fondamentales qui ruinent les espoirs d'une méthode aveugle dans le cas le plus général. En fait, le théorème « *No Free Lunch (NFL)* » [Wolp, 1997] montre que pour les problèmes de type boîte noire, toutes les méthodes sont équivalentes et font aussi bien, ou plutôt aussi mal, que l'énumération aléatoire.

Selon le point de vue opposé, la puissance d'une métaheuristique est d'abord liée à son aptitude à intégrer des connaissances spécifiques du problème. La connaissance du problème la plus fondamentale réside dans le codage du problème et dans le choix de la fonction de voisinage. Plusieurs auteurs insistent sur l'importance du codage du problème, voir par exemple [Radc, 1995]. Dans le cas du voyageur de commerce, plusieurs types de codages différents ont été proposés et conduisent à des performances très variées [Mich, 1992]. En général, il n'existe pas de codage universellement efficace. Un « bon » codage doit permettre de restreindre l'espace de recherche et d'intégrer des connaissances du problème.

Les métaheuristiques tentent également d'améliorer leur efficacité en incorporant des connaissances supplémentaires dans leurs opérateurs. Plus les opérateurs d'une méthode utilisent des connaissances spécifiques, plus la méthode dispose de moyens potentiels pour conduire efficacement la recherche. En contrepartie, l'intégration de ces connaissances spécifiques (en supposant que ces connaissances soient disponibles) nécessite un effort pour spécialiser ou adapter la méthode. La méthode tabou vise à incorporer le plus possible de connaissances du problème pour atteindre le maximum d'efficacité : l'utilisateur doit notamment définir de façon pertinente le type de caractéristique figurant dans la liste tabou.

De leur coté, les algorithmes génétiques se sont éloignés du modèle standard pour intégrer également des connaissances du problème : codage non binaire, opérateurs spécifiques. Au contraire, le recuit simulé est parfois présenté comme une méthode facile à adapter à un problème et qui ne tente pas d'exploiter de connaissances spécifiques.

En général, une méthode offrant des possibilités d'intégrer des connaissances du problème a plus de chance de produire de bons résultats, mais demande un effort d'adaptation et de spécialisation. Au contraire, une méthode très générale qui prétend n'intégrer aucune connaissance propre ne peut pas être compétitive.

II.5. Classification des métaheuristiques

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents.
Dans la littérature, plusieurs classifications de métaheuristiques ont été proposées. Nous présentons à ce niveau un aperçu des principales classifications.

II.5.1. Les métaheuristiques inspirées et non inspirées d'un phénomène naturel

Une manière intuitive de classifier les métaheuristiques consiste à séparer celles qui sont inspirées d'un phénomène naturel, de celles qui ne le sont pas.

Les algorithmes génétiques ou les algorithmes par colonies de fourmi entrent clairement dans la première catégorie, tandis que la méthode de descente, ou la recherche Tabou, vont dans la seconde.

II.5.2 Les métaheuristiques avec fonction objective statique ou dynamique

Les métaheuristiques peuvent être aussi classées selon la façon dont ils utilisent la fonction objective. Certains algorithmes conservent la fonction objective donnée dans la représentation de problème telle qu'elle est (comme la recherche locale guidée (GLS)) et la modifie lors de la recherche. L'idée de cette approche est d'éviter les minimums locaux en modifiant l'espace de recherche. En conséquence, lors de la recherche la fonction objective est altérée en essayant d'intégrer des informations intégrées au cours du processus de recherche.

II.5.3. Les métaheuristiques avec une ou plusieurs structures de voisinage

La plupart des métaheuristiques travaillent sur une seule structure de voisinage. En d'autres termes, la topologie du paysage de fitness ne change pas en cours de l'algorithme. D'autres métaheuristiques telles que la recherche par voisinage variable utilisent un ensemble de structures de voisinage qui donne la possibilité de diversifier la recherche en échangeant entre les structures de voisinage qui ont des formes différentes.

II.5.4. Les métaheuristiques avec et sans mémoire

Les métaheuristiques utilisent l'historique de leur recherche pour guider l'optimisation aux itérations suivantes. Dans le cas le plus simple, elles se limitent à considérer l'état de la recherche à une itération donnée pour déterminer la prochaine itération : il s'agit alors d'un processus de décision markovien, et on parlera de méthode *sans mémoire*. C'est le cas de la plupart des méthodes de recherche locale (recherche à voisinage variable, recherche locale

itérée, recherche locale stochastique, recherche locale guidée), algorithme d'estimation de distribution, recuit simulé, GRASP.

Beaucoup de métaheuristiques utilisent une mémoire plus évoluée, que ce soit sur le court terme (solutions visitées récemment, par exemple) ou sur le long terme (mémorisation d'un ensemble de paramètres synthétiques décrivant la recherche).

II.5.5. Les métaheuristiques implicite, explicite, directe

En considérant les métaheuristiques comme des méthodes itératives utilisant un échantillonnage de la fonction objective comme base d'apprentissage (définition plus particulièrement adaptée aux métaheuristiques à populations) apparaît le problème du choix de l'échantillonnage.

Dans la très grande majorité des cas, cet échantillonnage se fait sur une base aléatoire, et peut donc être décrit via une distribution de probabilités. Il existe alors trois classes de métaheuristiques, selon l'approche utilisée pour manipuler cette distribution.

La première classe est celle des méthodes *implicites*, où la distribution de probabilité n'est pas connue *a priori*. C'est le cas par exemple des algorithmes génétiques, où le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. L'évolution différentielle, Scatter search, algorithmes à essaim de particules, stratégies d'évolution et algorithmes de colonie de fourmis sont des méthodes implicites.

Par opposition, on peut donc classer les méthodes *explicites*, qui utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution.

Dans cette classification, le recuit simulé occupe une place particulière, puisqu'on peut considérer qu'il échantillonne la fonction objective en utilisant directement celle-ci comme distribution de probabilité (les meilleures solutions ayant une probabilité plus grande d'être tirées). Il n'est donc ni explicite ni implicite, mais plutôt « direct » (exemple : le recuit simulé) [JinK, 1999].

II.5.6. Les métaheuristiques évolutionnaires et non évolutionnaires

On trouve parfois une classification présentant les algorithmes d'optimisations stochastiques comme étant « évolutionnaires » (ou « évolutionnistes ») ou non. L'algorithme sera considéré comme faisant partie de la classe des algorithmes évolutionnaires s'il manipule une population *via* des *opérateurs*, selon un algorithme général donné.

Cette façon de présenter les métaheuristiques dispose d'une nomenclature adaptée : on parlera d'opérateurs pour toute action modifiant l'état d'une ou plusieurs solutions. Un opérateur construisant une nouvelle solution sera dénommé *générateur*, alors qu'un opérateur modifiant une solution existante sera appelé *mutateur*.

Dans cette optique, la structure générale des algorithmes évolutionnaires enchaîne des étapes de *sélection*, de *reproduction* (ou *croisement*), de *mutation* et enfin de *remplacement*. Chaque étape utilise des opérateurs plus ou moins spécifiques.

Quelques algorithmes évolutionnaires : l'algorithme génétique, la programmation génétique, l'algorithme d'évolution différentielle, les stratégies évolutionnaires et l'algorithme d'estimation de distribution [JinK, 1999].

II.5.7. Les métaheuristiques à base de population et les métaheuristiques à trajectoire

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

Les plus connues dans cette classe sont le recuit simulé, la recherche tabou, la recherche à voisinage variable, la méthode GRASP. L'autre approche utilise la notion de population. La métaheuristique manipule un ensemble de solutions en parallèle, à chaque itération. On peut citer les algorithmes génétiques, l'optimisation par essaims particulaires et les algorithmes de colonies de fourmis.

La frontière est parfois floue entre ces deux classes. On peut ainsi considérer qu'un recuit simulé où la température baisse par paliers, a une structure à population. En effet, dans ce cas on manipule un ensemble de points à chaque palier, il s'agit simplement d'une méthode d'échantillonnage particulière.

II.5.7.1. Les métaheuristiques à trajectoire (à solution unique)

Ici, nous résumons les plus connues des métaheuristiques à trajectoire.

a. La méthode de descente

Le principe de la méthode de descente (dite aussi *basic local search*) consiste, à partir d'une solution *S*, à choisir une solution *S'* dans un voisinage de *S* telle que *S' améliore* la recherche (généralement telle que $f(S') < f(S)$). On peut décider soit d'examiner toutes les solutions du voisinage *N* et prendre la meilleure de toutes ces solutions (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.
La méthode de descente est la méthode la plus élémentaire de recherche locale. On peut la formaliser comme suit :

Algorithme II.1 *descente_simple* (solution initiale *S*)

Début
  Répéter
    Choisir *S'* dans N(*S*)
    Si $f(S') < f(S)$ alors $S \leftarrow S'$
  Jusqu'à ce que $f(S') \geq f(S)$, $\forall S' \in$ **S**
Fin

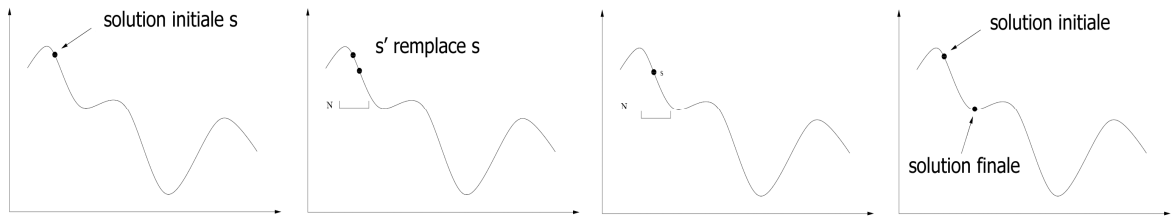


Figure II.2. *Evolution d'une solution dans la méthode de descente.*

On peut varier cette méthode en choisissant à chaque fois la solution S' dans N(S) qui améliore le plus la valeur de f. C'est la méthode de plus grande descente [Cost, 2006].

b. Recherche aléatoire (Random Search : RS)

C'est la plus simple des méthodes stochastiques. Cette méthode consiste à tirer à chaque itération une solution au hasard. La fonction objective *f* est évaluée en ce point. La nouvelle valeur est comparée à la précédente. Si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continu. Sinon on repart du point précédent et on recommence le procédé, jusqu'à ce que les conditions d'arrêt soient atteintes. L'algorithme II.2 [Luke, 2010] présente un pseudo code de la recherche aléatoire dans le cas d'un problème de minimisation.

Algorithme II.2 *Recherche aléatoire*

Début
$S_0 \leftarrow$ solution aléatoire;
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$;
Répéter
S ← solution aléatoire ;
Si ($f(S) < f_{min}$) *Alors*
$f_{min} \leftarrow f(S)$;
$x_{min} \leftarrow S$;
Fin Si
Jusqu'à conditions d'arrêt satisfaites.
Fin

c. Hill Climbing (HC)

On part d'une solution si possible "bonne" (par exemple donnée par une heuristique) et on balaie l'ensemble des voisins de cette solution ; s'il n'existe pas de voisin meilleur que notre solution, cela veut dire qu'un optimum local a été trouvé et on arrête la recherche. Sinon, on choisit le meilleur des voisins et on recommence. Une autre implémentation consiste non pas à passer au meilleur des voisins à chaque étape, mais au premier meilleur voisin trouvé. La convergence vers un optimum local pouvant être très lente, on peut éventuellement fixer un nombre de boucles maximum, si on veut limiter le temps d'exécution [Sea, 2010].

Cette méthode a l'inconvénient de rester bloquée dans un optimum local : une fois un optimum local trouvé, on s'arrête, même si ce n'est pas l'optimum global. Selon le paysage des solutions, l'optimum local peut être très bon ou très mauvais par rapport à l'optimum global. Si la solution de départ est donnée par une heuristique déterministe, l'algorithme sera déterministe. Si elle est tirée au hasard, l'algorithme devient non déterministe et donc plusieurs exécutions différentes sur la même instance pourront donner des solutions différentes et de qualités différentes [Sea, 2010].

La notion de voisinage est primordiale. Si les voisins sont très nombreux, on a de fortes chances de trouver l'optimum global, mais visiter un voisinage peut être très long: on visitera une grande partie de l'espace des solutions. Si le voisinage est très restreint, on risque fort de rester bloqué dans un optimum local de "mauvaise qualité". Le choix de la notion de voisinage est un compromis entre efficacité et qualité.

Le pseudo-code de l'algorithme Hill Climbing est celui présenté par l'algorithme II.3 [Sea, 2010]:

Algorithme II.3 *HILL CLIMBING*

Début
$S_0 \leftarrow$ Solution aléatoire
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$
Répéter
Engendrer un N-échantillon $S_i(x)$ voisinage de S(x) *ET* calculer
$f(S(x)) = \min[f(S_i(x))]$ avec $1 \leq i \leq N$

Si ($f(S) < f_{min}$) *then*
$x_{min} \leftarrow S$
Sinon
Sortir de Répéter
Fin Si
Jusqu'à condition d'arrêt satisfaite.
Fin.

d. Recuit Simulé

Le recuit simulé a été introduit et mis au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985. Il a été repris sous différentes formes par Lawrence Davis en 1987 [Lawr, 1987].
Cet algorithme a été dérivé de l'algorithme de Metropolis, développé par les scientifiques du projet ex-Manhattan : Nicholas Metropolis, Arianna et Marshall Rosenbluth, Augusta et Edward Teller en 1953. Ce dernier permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. Partant d'une solution donnée, en la modifiant, on en obtient une seconde. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

La description des phénomènes physiques et quantiques liés au processus de recuit s'appuie sur la statistique de Boltzmann. Pour qu'un métal ait une qualité optimale, il faut que son état d'énergie soit minimal. Pour améliorer la qualité d'un métal, on le chauffe, puis on le refroidit par paliers en attendant à chaque fois que l'état d'énergie soit stabilisé. Au niveau de l'atome, cela signifie qu'à une température chaude, les atomes bougent beaucoup, et en se

refroidissant, ils trouvent leur place optimale. Plus la température descend, moins les atomes bougent et le métal devient de plus en plus résistant.

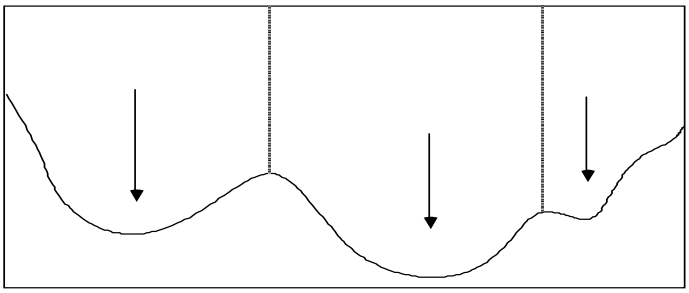


Figure II.3. *Un paysage d'énergie. Suivant la configuration initiale, indiquée par les flèches, la dynamique aboutit à température nulle dans l'un quelconque des minima relatifs, séparés par les barrières indiquées en pointillés. A température élevée, les processus probabilistes permettent au système de sauter les barrières séparant les vallées.*

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. À cette solution correspond une énergie initiale $E = E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T = T_0$ élevée est également choisie. Ce choix est alors totalement arbitraire et va dépendre de la loi de décroissance utilisée.

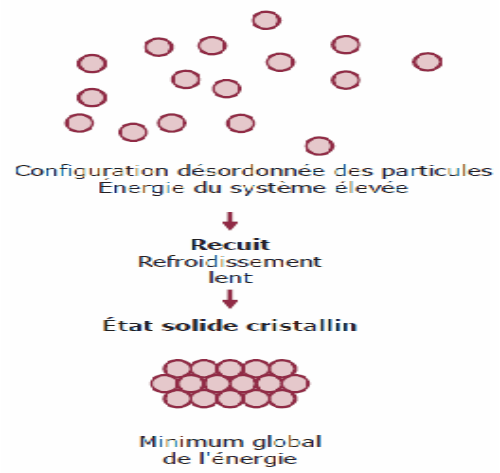


Figure II.4. *Transposition de procédé recuit à la résolution d'un problème d'optimisation.*

À chaque itération de l'algorithme une modification élémentaire de la solution est effectuée. Cette modification entraîne une variation Δ_E de l'énergie du système (toujours calculée à partir du critère que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta_E}{T}}$. Ce choix de l'exponentielle pour la probabilité s'appelle règle de Metropolis.

On itère ensuite selon ce procédé en gardant la température constante.

Algorithme II.4 *Recuit simulé*

```
Début
    Initialisation du temps
    Tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint] Faire
        Initialisation de la température à Tmax
        Tant que [la température est supérieure à 0] et [le temps limite n'est pas atteint] Faire
            Incrémentation du temps
            Modification de la solution courante
            Si [la nouvelle solution est meilleure que l'ancienne] Alors
                la sauvegarder
            Sinon
                Si [le système est stable depuis un nombre défini d'itérations] Alors
                    Diminuer la température
                Fin si
            Fin si
        Fin tant que
    Fin tant que
Fin
```

L'algorithme varie de Hill-Climbing (algorithme 3) dans sa décision du moment de remplacer S, la solution candidate d'origine, avec R, son enfant nouveau peaufiné. Plus précisément, si R est meilleur que S, nous allons toujours remplacer S par R comme d'habitude. Mais si R est pire que S, on peut toujours remplacer S par R avec une certaine probabilité P(t, R, S):

$$P(t, R, S) = e^{\frac{\text{Quality}(R) - \text{Quality}(S)}{t}}$$

Où t> 0. Cette équation est intéressante à deux égards. Notez que la fraction est négative car R est pire que S. Premièrement, si R est bien pire que S, la fraction est plus grande, et donc la probabilité est proche de 0. Si R est très proche de S, la probabilité est proche de 1. Ainsi, si R n'est pas bien pire que S, nous allons toujours choisir R avec une probabilité raisonnable.

Deuxièmement, nous avons un paramètre ajustable t. Si t est proche de 0, la fraction est de nouveau un grand nombre, et donc la probabilité est proche de 0. Si t est élevé, la probabilité est proche de 1. L'idée est d'abord de mettre t comme un grand nombre, ce qui provoque l'algorithme de se déplacer à chaque solution nouvellement créée indépendamment de sa qualité. Nous faisons une marche aléatoire dans l'espace. Puis t diminue lentement, éventuellement à 0, à quel point l'algorithme ne fait rien de plus que de simples Hill-Climbing.

e. Recherche Tabou (Tabu Search : TS)

La recherche avec tabou, ou simplement dite recherche tabou, est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans un article paru en 1986 [Glov, 1986], mais reprenant de nombreuses idées proposées antérieurement dès les années 60 dans les deux articles simplement intitulés « Tabu chearch » [Glov, 1989] proposant la plus part des principes de cette recherche telle qu'elle est décrite actuellement. Maintenant, elle est d'usage très classique en domaine d'optimisation combinatoire pour résoudre les problèmes NP-durs.

L'idée principale de la recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui optimise la fonction objective. Le voisinage choisi est exploré de manière déterministe, il est interdit de reprendre des solutions récemment visitées.

L'un des principes fondamentaux de cette recherche qui la caractérise des autres méta-heuristiques est la construction d'un historique de la recherche itérative ou, ce qui est équivalent, au fait de doter la recherche de mémoire [Glov, 1997]. La recherche tabou examine un échantillonnage de solutions *N(S)* et retient la meilleure solution *S* des solutions obtenues.

A chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche tabou qu'elle est une méthode agressive [www10].

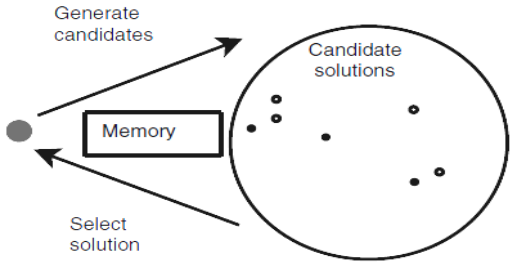
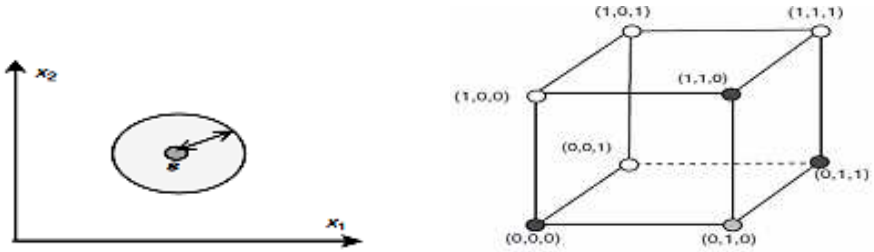


Figure II.5. *Principe de base d'une méta-heuristique à mémoire.*

La recherche tabou est essentiellement axée sur une exploration non triviale de l'ensemble des solutions en utilisant la notion de voisinage. Formellement pour toute solution *s* de *S*, un ensemble de $N(s) \in S$ qu'on appelle ensemble des solutions voisines de *s*.



(a) *(b)*

Figure II.6. *Les types de solutions du voisinage.*
(a) *Le voisinage pour un problème à variables continus : Le cercle symbolise les voisins de la solution s,*
(b) *Le voisinage pour un problème à variables discrètes : Les nœuds du cube sont les solutions et leurs voisins adjacents.*

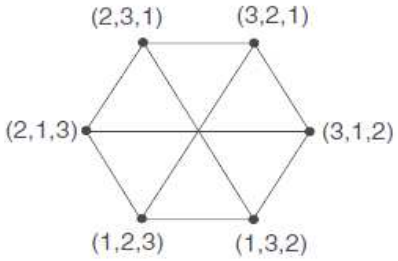


Figure II.7. *Voisinage d'une permutation de taille égale à 3.*
Les voisins de la solution (2.3.1) sont : (3.2.1), (2.1.3), et (1.3.2).

Vue d'un côté pratique l'ensemble N(s) des solutions voisines de s n'est autre que l'ensemble des modifications que l'on peut apporter à cette dernière. On appelle *mouvement* une modification apportée à une solution.

Dans la littérature et dans le cas de la recherche tabou, l'application d'un mouvement *m* à une solution *s* est notée ; d'où l'expression suivante du voisinage :

$$N(s) = \{s' / s' = s \oplus m, m \in M\}$$

La figure II.8 [Kamm, 2006] illustre l'ensemble des mouvements possibles dans le cas d'une solution composée de quatre éléments.

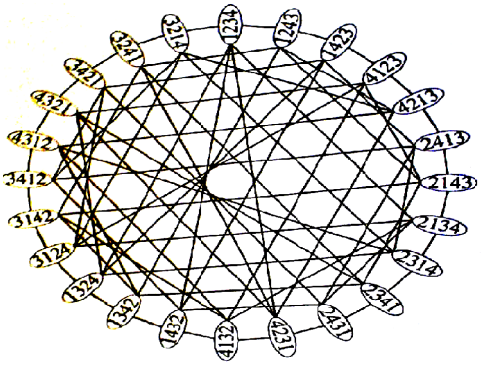


Figure II.8. *Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments.*

La recherche tabou est une méthode itérative qui explore un espace de solutions. Comme toute méthode itérative, il est intéressant d'éviter de revisiter des solutions pour essayer de converger vers la solution optimale le plus rapidement possible et ceci n'est possible qu'à l'aide d'une mémoire caractérisée par une taille définie par l'utilisateur et le problème à résoudre. Cependant, l'utilisation d'une mémoire peut s'avérer dans bien des cas peu efficace, voire mauvaise. Outre le fait de mettre cette idée en pratique est difficile, cela suppose que l'on doit mémoriser chaque solution visitée et de tester à chaque itération et pour chaque solution admissible si cette dernière a été déjà examinée. Les *tables de hachage* permettent de faire cela de manière efficace, mais ceci n'empêche pas la croissance linéaire de la taille de la mémoire avec le nombre des itérations effectuées.

Une table de hachage est définie par un tableau *T* de *J* entiers, avec *J* choisi relativement grand et assez voisin de la capacité de la machine utilisée tout en prenant en considération la taille du problème.

Le but de la table de hachage est :

- ✓ d'interdire le retour aux solutions obtenues pendant les *t* dernières opérations ;
- ✓ ainsi éliminer les cycles de longueurs égales ou inférieures à *t*.

Toutefois, le fait d'interdire un grand nombre de mouvements aura pour conséquence de rendre l'obtention de bons résultats très délicate faute de mouvements disponibles. Si ce nombre diminue, cela augmentera les chances d'une meilleure exploration aux alentours des optimums locaux et d'obtenir ainsi de meilleures solutions. Il ne faut cependant pas trop diminuer ce nombre, car, dans ce cas, il devient très probable de se trouver prisonnier d'un ensemble très restreint de solutions et de les visiter tout le temps de la recherche.

Pour palier à ce compromis du choix de nombre des mouvements à interdire et de bénéficier simultanément des avantages d'un petit nombre, qui permet une visite détaillée du voisinage d'une même solution, et d'un grand nombre qui permet de franchir le voisinage de différentes solutions, ce nombre doit varier au cours du processus itératif. Pour ce faire, plusieurs méthodes existent. Ce nombre peut être tiré au hasard à partir d'un intervalle donné à chaque itération ou après un nombre d'itérations, comme il peut aussi croître ou décroître en fonction des résultats obtenus au cours de la recherche [Tail, 1991], [Tail, 1995] et [Tail, 1999].

Autrement dit, une liste tabou avec trop d'éléments peut devenir très restrictive. En effet, il a été observé que trop de contraintes tabou forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération. Par contre une liste tabou contenant trop peu d'éléments peu s'avérer inutile et mener à des mouvements cycliques [Ayas, 2004].

En plus, dans beaucoup de problèmes, l'interdiction de revisiter des solutions mènerait à des incohérences comme la déconnection de la solution courante de la solution optimale ou bien le blocage de la recherche itérative par cause d'absence de solutions voisines non visitées[Kamm, 2006].

La figure suivante illustre ces derniers cas incohérents :

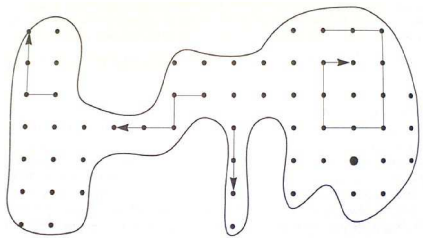


Figure II.9. *Déconnections et blocages dans la recherche tabou.*

Il est à signaler qu'une liste tabou peut être soit statique ou dynamique. Pour la première, la taille de la liste est fixée, en générale au début. Elle dépend de la taille du problème à résoudre et du voisinage choisi. Pour la deuxième, la taille doit changer durant le processus de recherche sans utiliser aucune information sur la mémoire de recherche.

Algorithme II.5 *Recherche Tabou*

Début

Φ Fonction de coût
Variables locales : S solution courante, Liste Taboue L, Meilleure solution M, itération courante K, nombre d'itérations N
Paramétrages : taille de liste taboue, critère d'aspiration
Choix d'une solution initiale S_0
S← S_0
M← S
K ←0
Tant que K< N *faire*
K← K+1
Mise à jour de L
Génération des candidats E par opération de voisinage
C ← Best (E)
Si Φ (S) < Φ(M) *ou* C n'est pas taboue *ou* C vérifie l'aspiration *alors*
S ← C
Sinon
E ← E \ C
Fin tant que
Retourner S

Fin.

f. GRASP

À proprement parler, GRASP (pour *Greedy Randomized Adaptive Search Procedure*) est une méthode hybride, car elle cherche à combiner les avantages des heuristiques gloutonnes, de la recherche aléatoire et des méthodes de voisinage. Un algorithme GRASP répète un processus composé de deux étapes : la construction d'une solution suivie par une descente pour améliorer la solution construite. Durant l'étape de construction, une solution est itérativement construite : chaque itération ajoute un élément dans la solution partielle courante. Pour déterminer l'élément qui sera ajouté, on utilise une liste des meilleurs candidats obtenus avec une fonction gloutonne et on prend *au hasard* un élément dans cette liste. La liste des meilleurs candidats est dynamiquement mise à jour après chaque itération de construction. Cette étape de construction continue jusqu'à ce qu'une solution complète soit obtenue [JinK, 1999].

À partir de cette solution, une descente est appliquée pour améliorer la solution. Une procédure GRASP répète ces deux étapes et retourne à la fin la meilleure solution trouvée. Les deux paramètres de cette méthode sont donc la longueur de la liste de candidats et le nombre d'itérations autorisées.

Algorithme II.6 *GRASP*

```
Début
    x, x*, f* : type Données
    Début
    f* ← infini
    Pour i = 1 à nombre d'itérations faire
        x ← Construction Aléatoire Gloutonne ()
        x ← Recherche Locale(x)
        Si f(x) < f* alors
            x* ← x
            f* ←f(x)
        Fin Si
  Fin Pour
Fin.
```

II.5.7.2. Les métaheuristiques à population

Dans cette section, nous présentons un sommaire des plus connues des métaheuristiques à population.

a. Les algorithmes évolutionnaires

Les *algorithmes évolutionnistes* ou *algorithmes évolutionnaires* (*evolutionary algorithms* ou encore dits *evolutionary computation* en anglais), sont une famille d'algorithmes s'inspirant de la théorie de l'évolution pour résoudre des problèmes divers. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation, elles sont en cela des métaheuristiques, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. On les classe également parmi les méthodes d'intelligence calculatoire.

Selon la génétique et la théorie de l'évolution [www11] :

- ✓ Un enfant hérite son patrimoine génétique pour moitié de sa mère et pour moitié de son père (reproduction sexuée).
- ✓ Les enfants ne sont pas identiques aux parents car des altérations des gènes peuvent se produire (mutations).
- ✓ Parmi les mutations, certaines peuvent être favorables et d'autres défavorables.
- ✓ Il naît beaucoup de descendants : mais seuls les individus les mieux adaptés pourront survivre et transmettre leurs gènes à leur tour à leur descendance.

Dans ce modèle, on observe que [www11] :

- ✓ Le hasard joue un rôle moteur pour produire de nouveaux individus différents de leurs parents.
- ✓ La sélection naturelle effectue le tri entre les variations favorables et les autres.

Ainsi, basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin, ces algorithmes présentent des qualités intéressantes pour la résolution des problèmes d'optimisation. Les individus ou chromosomes d'un algorithme évolutionnaires sont des codages des solutions possibles du problème. Comme dans la nature, ces individus forment une population qui va évoluer dans le temps selon des lois de sélection qui vont favoriser les mieux adaptés à se croiser entre eux en produisant des populations meilleures. L'évolution des individus d'une population à une autre se fait à l'aide de la reproduction. Les individus parents vont se reproduire pour donner des individus enfants qui seront plus performants après avoir subis des opérations génétiques de croisement et de mutation.

Et comme ces reproductions se font avec une part de hasard par analogie à la nature, donc, les parents candidats à la reproduction sont choisis d'une manière probabiliste proportionnelle à leurs aptitudes et l'étape de reproduction est choisie d'une façon totalement aléatoire.

Finalement, passant d'une génération à une autre, les individus forment une progéniture plus performante qui s'approche au mieux de la solution optimale [Kamm, 2006].

La figure suivante résume le principe de résolution de problèmes par algorithmes évolutionnaires.

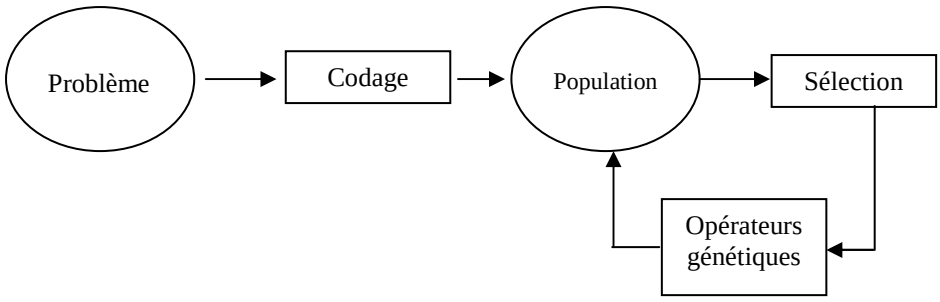


Figure II.10 . *Principe des algorithmes évolutionnaires.*

Ainsi, un algorithme évolutionnaire est construit autour des notions suivantes :

- **Le codage :**

La première étape de la résolution d'un problème à l'aide d'un algorithme génétique est la modélisation appropriée des solutions. Cette modélisation est appelée *codage* et permet de représenter les solutions sous forme de chromosomes.

Le codage se base sur deux notions importantes : le génotype et le phénotype.

- ✓ **Génotype** : représente l'ensemble des valeurs des gènes d'un chromosome.
- ✓ **Phénotype** : c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype.

S'il y a passage immédiat du phénotype au génotype, le codage est dit direct, sinon il est dit indirect et une procédure de passage est indispensable. Le codage d'une solution doit décrire toutes les données du problème et respecter toutes ses contraintes.

- **Population initiale :**

L'espace de recherche étant infini, il est très difficile de répartir plus ou moins équitablement la population initiale sur l'espace de recherche, de sorte que cet espace soit au maximum parcouru. Un certain nombre d'individus doivent être générés aléatoirement s'il n'existe aucune autre méthode d'initialisation.

- **Evaluation de la qualité d'une solution :**

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance (*fitness*) de mesurer cette efficacité pour permettre à l'algorithme de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution. Autrement dit, la fonction de performance *f*, doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre.

- **La sélection :**

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.
Plusieurs modes de sélection ont été définis tels que : sélection par roulette (wheel), sélection par rang, sélection steady-state, élitisme, sélection uniforme, etc.

- **Opérateurs génétiques :**

1) Croisement : C'est un opérateur génétique qui permet à deux chromosomes parents de produire deux chromosomes enfants où de nouvelles séquences de gènes pour les chromosomes enfants sont crées à partir d'une base de configuration des séquences héritées des chromosomes parents. Cet opérateur se produit selon une probabilité *Pc* fixée par l'utilisateur selon le problème à optimiser.

Il existe plusieurs manières d'effectuer un croisement soit par cross-over où l'on a besoin de deux parents qui génèrent à la fin du croisement deux enfants, soit par copie où l'on n'a besoin que d'un seul parent qui nous donnera à la fin du croisement un enfant qui est le parent lui-même (sa copie).

Dans la littérature, il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter [Mesg, 1999]. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que :

- ✓ le croisement à un point.
- ✓ le croisement multipoints.
- ✓ le croisement uniforme.

2) Mutation : permet de changer (permuter) un gène d'un chromosome par un autre d'une manière aléatoire, ce qui est à première vue très ressemblant avec un cross-over. La différence est que le cross-over essaie de converger vers une solution qui lui parait la meilleure (s'intéresse à la qualité), mais que la mutation permet la diversité. En quelque sorte, la mutation sert à éviter une convergence prématurée de l'algorithme. Par exemple, lors de la

recherche d'une solution optimum, la mutation sert à éviter la convergence vers un optimum local.
Cet opérateur est aussi appliqué avec une probabilité P_m. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche aléatoire et conserver le principe de sélection et d'évolution.

Dans la littérature plusieurs opérateurs de mutation ont été définis, tels que : la transposition de deux allèles consécutifs, la transposition de deux allèles quelconques, l'inversion d'allèles, etc.

Les algorithmes évolutionnaires se scindent en grandes familles, dont les principales sont :

1) Programmation évolutionnaire

La programmation évolutionnaire développée par *L.J. Fogel* [Foge, 2003], se base sur l'évolution d'une population d'automates à états finis (Figure II.11) pour résoudre des problèmes de prédiction. Ce modèle évolutionniste accentue l'utilisation de la mutation et n'utilise pas dans sa version originale la recombinaison des individus par croisement [Renn, 2000]. La table de transition des automates est modifiée grâce à des mutations aléatoires uniformes dans l'alphabet discret correspondant. Chaque automate de la population parente génère un enfant par mutation, et les meilleures solutions entre les parents et les enfants sont sélectionnées pour survivre, sachons que l'évaluation de la performance des individus correspond au nombre de symboles prédits correctement.

Par la suite, la programmation évolutionnaire a été développée et son domaine a été élargi par *D.B. Fogel*, afin qu'il puisse travailler dans l'espace réel, où la sélection déterministe est remplacée par un tournoi stochastique.

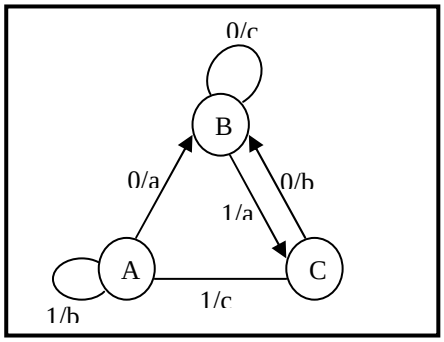


Figure II.11. *Exemple d'un automate à états finis ayant trois états différents S={A,B,C}, un alphabet d'entrée I={0,1}, et un alphabet de sortie O={a,b,c}. Chaque arrête entre deux états indique une transition possible, et la fonction de transition $\delta : S \times I \rightarrow S \times O$ est spécifiée par les labels au niveau des arrêtes ayant la forme i / o, signifiant que $\delta (s_i , i) = (s_j , o)$.*

2) Stratégies d'évolution

Les stratégies d'évolution sont dédiées à l'optimisation des problèmes continus dans l'espace de vecteurs des réels [Magn, 2001]. Les premiers efforts pour la mise en place des stratégies d'évolution ont eu lieu en 1973 à l'université de Berlin par Rechenberg [Rech, 2003] au cours de la résolution d'un problème aérodynamique. C'est avec ces méthodes évolutionnistes que la notion d'auto-adaptativité pour la mutation permettant de contrôler cette fonction de mutation, a été apparue. Une mise en œuvre de ce principe consiste à augmenter l'intensité de la mutation lorsque la proportion de descendants de bonne qualité, c'est à dire le nombre de mutation à succès, dépasse 20% de la population totale [Renn,

2000]. Elle est diminuée dans le cas opposé. Une interprétation possible de cette règle est la suivante : si la proportion de mutation réussie est élevée, l'espace de recherche exploré est restreint autour d'un optimum local, il faut donc diversifier la population en augmentant le taux de mutation, ce qui revient à ajuster la variance de la mutation au cours du temps par le processus d'évolution. C'est ce que les méthodes évolutionnistes auto-adaptatives visent de le faire : automatiser le réglage des paramètres de l'algorithme évolutionniste pour remédier aux méthodes empiriques du type « essais-erreurs » qui sont employées dans la pratique. De plus, ces approches utilisent un opérateur de sélection de type déterministe: les solutions dont le *fitness* est mauvais sont éliminées de la population. En outre, dans le modèle originel, les populations des parents et de leurs descendants sont généralement de taille différente.

3) Algorithmes génétiques

Les algorithmes génétiques (AGs) sont probablement les algorithmes les plus connus et les plus utilisés dans le calcul évolutionnaire. Ils ont été développé dans les années soixante par *Holland* qui les a appliqué à l'optimisation paramétrique pour la première fois en 1975 [Holl, 1975], en posant, ainsi, les fondements de cette technique d'application. Cependant, cette technique n'a pas été appliquée sur des problèmes réels de grande taille, à cause des machines calculatoires, de l'époque, et qui n'ont pas été suffisamment puissantes. Ce n'est que vers la fin des années quatre vingt, précisément, avec l'apparition de l'ouvrage de référence écrit par *Goldberg* [Gold, 1989], que les algorithmes génétiques ont été connus dans la communauté scientifique. Dans ce domaine, d'autres travaux peuvent faire la référence aussi, tel que celui de Michalewicz [Mich, 1996]. Leur particularité est qu'ils sont fondés sur le *Néo-Darwinisme*, c'est-à-dire l'union de la théorie de l'évolution et de la génétique moderne. Ainsi, les variables sont généralement codées en binaire, par analogie avec les quatre lettres de l'alphabet génétique d'ADN, sous forme de gènes dans un chromosome. Ensuite, des opérateurs génétiques, à savoir le croisement et la mutation, sont appliqués à ces chromosomes.

Les AGs sont utilisés pour retrouver une solution résolvant un problème donné, et ce sans avoir de connaissance à priori sur l'espace de recherche. Seul un critère de qualité est nécessaire pour évaluer les différentes solutions en quantifiant, ainsi, leur capacité à résoudre le problème donné. Donc, le but scientifique et technologique visé par ces algorithmes est de pouvoir traiter des problèmes d'optimisation globaux, grâce à la *généralité* avec laquelle on représente l'espace de recherche qui peut contenir des booléens (système actif ou non), des entiers (nombre de composants à optimiser), des réels (intensités associées aux composants réglables), ou des fonctions discrétisées (optimisation de forme), et grâce aussi à la *robustesse* de la convergence [Dupa, 2004].

4) Programmation génétique

L'idée de faire évoluer des programmes date des années cinquante où *Friedberg*, en 1958, a fait plusieurs tentatives pour avoir des ordinateurs auto-programmables en utilisant ce qui est de la mutation actuellement. Donc, et à partir d'une population constituée de programmes aléatoires dont il modifie leurs contenus stochastiquement, il essaye de les améliorer pour aboutir à des résultats satisfaisants. Plutard, *Smith* (1980) travaillant sur les systèmes classifieurs d'apprentissage, a introduit de petits programmes dans les règles qu'il cherche à les faire évoluer [Magn, 2001]. Il convient de noter que, la première utilisation des structures arborescentes dans un algorithme génétique a été suggérée par *Cramer* en 1985 dans le but de faire évoluer des sous-programmes séquentiels d'un langage algorithmique simple.

Toutefois, c'est grâce à *John Koza* (1992) [Koza, 1992] que cette présentation a été adoptée pour définir la programmation génétique comme un nouvel algorithme évolutionnaire, en étendant, ainsi, le modèle d'apprentissage des AGs à l'espace des programmes. Donc, son objectif initial était de faire évoluer des sous-programmes du langage LISP (Figure II.12), et c'est d'ailleurs grâce à son ouvrage que l'utilisation de la programmation génétique s'est étendue à la résolution de nombreux types de problèmes où les solutions peuvent être représentées par des structures arborescentes dont les feuilles sont constituées de symboles terminaux (variables, constantes,…), et les nœuds internes de symboles fonctionnels (opérateurs).

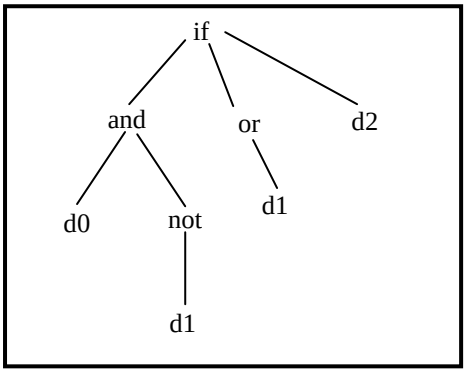


Figure II.12. *Exemple d'une solution Programmation génétique en LISP : {d0, d1, d2} est un ensemble d'instructions constituant les terminaux, et {if, and, or} sont des expressions LISP constituant les nœuds.*

5) Evolution différentielle

L'algorithme à évolution différentielle est inspiré des algorithmes génétiques et de l'évolution des stratégies combinées à une technique de recherche géométrique. Toutefois, l'évolution différentielle s'écarte encore un peu plus des principes de la génétique en abandonnant le codage génotype-phénotype et en faisant varier directement les paramètres réels proches du phénotype. L'évolution différentielle est donc valide pour tout type de fonction objective à valeurs réelles et peut tenir compte des propriétés mathématiques générales de ces fonctions. En revanche, tout comme l'algorithme génétique, l'évolution différentielle utilise une succession de générations définies par des opérations de mutation et de recombinaison de réels et donc possède des notions d'enfants et de parents [Laye, 2010].

L'évolution différentielle a été conçue comme une méthode de recherche parallèle, directe, et stochastique. Chaque solution est encodée avec un vecteur n-dimensionnel basé sur des nombres à virgule flottante. La taille de la population (m) ne change pas au cours du processus d'optimisation (minimisation ou maximisation). A chaque génération, de nouveaux vecteurs sont générés par la combinaison des vecteurs choisis au hasard de la population actuelle. La nouveauté de cet algorithme réside dans l'opérateur de mutation. Contrairement à la mutation des algorithmes génétiques, l'évolution différentielle utilise un concept de mutation auto référentiel se limitant à des combinaisons de mutations déjà présentes dans la population. L'opérateur de mutation consiste à générer de nouveaux vecteurs par le calcul de la différence pondérée de deux (ou quatre) autres vecteurs, selon la formule suivante [Laye, 2010] :

$$\overrightarrow{v_i} = \overrightarrow{a_{r1}} + F \times (\overrightarrow{a_{r2}} - \overrightarrow{a_{r3}})$$

Où i = 1,2,..., m, et les indices aléatoires r1, r2, r3 $\in$ [1,2,..., m] sont mutuellement différents et distincts de l'indice i. Cet opérateur est utilisé en liaison avec l'opérateur de croisement. Ce

dernier est introduit en vue d'augmenter la diversité de la population. Il combine un vecteur muté obtenu par l'opération de mutation avec l'un des vecteurs de la population. Finalement, l'opérateur de sélection compare seulement la valeur de la fonction objective des deux vecteurs en compétition et le meilleur individu est sélectionné pour la population de la prochaine génération.

Un algorithme d'évolution différentielle peut être résumé comme suit [Laye, 2010] :

Algorithme II.7 *Algorithme à évolution différentielle*

Début
Génération aléatoire de la population de vecteurs
Évaluer chaque solution
Répéter
Appliquer la mutation différentielle
Exécuter un croisement différentiel
Évaluer la nouvelle solution
Appliquer la sélection différentielle
Jusqu'à (le nombre de génération est atteint)
Fin.

6) La recherche par dispersion (Scatter Search)

La recherche par dispersion est une méthode d'optimisation relativement ancienne. Cette approche évolutionnaire a pour origine les stratégies de création de règles de décision composées et de contraintes de remplacement. Les études récentes ont démontré les avantages pratiques de cette approche pour résoudre divers problèmes d'optimisation. La recherche par dispersion contraste avec d'autres procédures évolutionnaires, telles que les algorithmes génétiques, en utilisant des conceptions stratégiques là où d'autres approches utilisent l'aléatoire [Jour, 2003].

La recherche par dispersion opère sur un ensemble de solutions appelé l'*ensemble de référence*, en les combinant pour en créer des nouvelles. À la différence d'une "population" dans les algorithmes génétiques, l'ensemble de référence de solutions dans la recherche par dispersion est relativement petit.

La recherche par dispersion comprend les cinq méthodes suivantes [Jour, 2003] :

- ✓ Une méthode de génération de diversification pour produire une collection de solutions diverses, en utilisant une solution d'essai arbitraire (ou solution initiale) comme entrée.
- ✓ Une méthode d'amélioration pour transformer une solution initiale en une ou plusieurs solutions d'essai améliorées.
- ✓ Une méthode de mise à jour de l'ensemble de référence pour construire et maintenir un ensemble de référence comprenant les meilleures solutions trouvées, organisé pour fournir l'accès efficace par d'autres parties de la méthode. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité.
- ✓ Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ces solutions comme une base pour créer des solutions combinées.

- ✓ Une méthode de combinaison de solutions pour transformer un sous-ensemble donné de solutions produit par la méthode de génération d'un sous-ensemble en un ou plusieurs vecteurs combinés de solutions.

7) Algorithmes à Estimation de Distribution (Estimation of Distribution Algorithms)

Récemment, une nouvelle classe d'algorithmes a fait son apparition : les algorithmes à estimation de distribution (EDA). Les stratégies évolutionnaires mettent en œuvre des opérateurs de mutation et de croisement mais il est difficile pour un utilisateur inexpérimenté de choisir l'opérateur approprié à son problème. Les algorithmes à estimation de distribution reprennent les principes des algorithmes à population mais utilisent des modèles probabilistes à la place d'opérateurs de mutation et de croisement pour construire de nouveaux individus.

Le modèle de fonctionnement de l'algorithme est le suivant [Jour, 2003] :

1. Génération de la population initiale.
2. Sélection des individus prometteurs.
3. Estimation de la distribution de ces individus (construction d'un modèle probabiliste).
4. Génération de nouvelles solutions à partir du modèle probabiliste.
5. Retour en 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Les EDA peuvent être appliqués aussi bien sur un domaine discret que sur un domaine continu. Il existe de nombreux algorithmes à estimation de distribution qui peuvent être classés selon le modèle utilisé pour la construction des nouveaux individus [Jour, 2003]: compact GA (produit de distribution de Bernouilli), Population Based Incremental Learning (règle de Hebian), Univariate Marginal Distribution Algorithm, extended compact GA (produit de distribution marginale) et Bayesian Optimization Algorithm (réseau bayesien).

8) Les algorithmes bactériologiques

Les algorithmes bactériologiques basés sur les algorithmes génétiques et qui sont donc assez similaires [Baud, 2005a], sont apparus récemment en tant que métaheuristique. Ils ont été développés par l'équipe Triskell [www12] à Rennes, notamment grâce à la thèse de Benoit Baudry [Baud, 2005b]. Lors du développement d'un générateur de cas de tests utilisant la technique de la mutation-based testing [Baud, 2005c], en utilisant les algorithmes génétiques, l'équipe s'est aperçue que l'utilisation d'algorithmes génétiques n'était pas bien adaptée. En effet, les nouveaux cas de tests n'étaient créés que par l'opérateur de mutation, le croisement ne faisant que récrire des cas déjà existants. L'équipe a donc eu l'idée d'utiliser la reproduction des bactéries, qui se multiplient en se clonant et des mutations s'opèrent. Les bactéries ayant le meilleur patrimoine génétique sont celles qui survivent le mieux dans leur environnement.

Les différences entre algorithmes génétiques et bactériologiques se situent au niveau du croisement des individus et au niveau de la sélection des individus. Dans l'algorithme bactériologiques l'opérateur de croisement a disparu, et pour ne pas perdre les bactéries les mieux adaptées, les meilleures sont sauvegardées lors de la sélection, à chaque itération. Il est possible d'en sauvegarder un certain nombre, mais aussi de sauvegarder les X meilleures. Les premières itérations entraînent alors la sauvegarde de toutes les bactéries.

Un algorithme bactériologique se déroule de manière très similaire à un algorithme génétique. Il opère suivant les étapes suivantes :

- ✓ initialisation du temps,
- ✓ création de la population initiale,
- ✓ tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint], faire:
  - incrémentation du temps,
  - mutations aléatoires à la population (Toutes les bactéries mutent),
  - évaluation de l'adaptation de chaque bactérie,
  - sauvegarde des meilleures bactéries.

Actuellement, la distinction entre ces approches est de plus en plus floue. Le génotype, qui est le codage d'un individu, étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, ...). Donc, la différence entre ces quatre catégories est essentiellement d'ordre historique.

b. Algorithmes de colonies de fourmis

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis, à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [Dene, 1983], [Dene, 1989]. Les fourmis trouvent le plus court chemin entre leur nid et une source de nourriture. La résolution de ce problème, qui est assez complexe en soi, fait appel à une certaine organisation et à un travail collectif. Des fourmis peuvent résoudre ce problème collectivement en se basant sur un moyen de communication particulier : « la phéromone ».

1) Les fourmis réelles

Les fourmis réelles sont aveugles et cherchent de la nourriture en se déplaçant de façon quasi aléatoire. Tout au long de leur déplacement, elles laissent derrière elles une substance chimique appelée phéromone. Cette substance a pour but de guider les fourmis vers leur objectif et possède la propriété de s'évaporer au cours du temps. Une fois cet objectif atteint (dans ce cas, la nourriture trouvée), les fourmis rentrent au nid, en rebroussant chemin, grâce à leur trace de phéromone. Celle-ci s'en trouve renforcée. Plus une trace de phéromone est concentrée, plus elle va attirer les fourmis. Au fil du temps, le plus court chemin, du nid vers la nourriture émergera, grâce au renforcement de la trace de phéromone (figure II.13). D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs congénères.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis. Ainsi, une colonie est capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [Goss, 1989] [Beck, 1992], sans que les individus aient une vision globale du trajet.

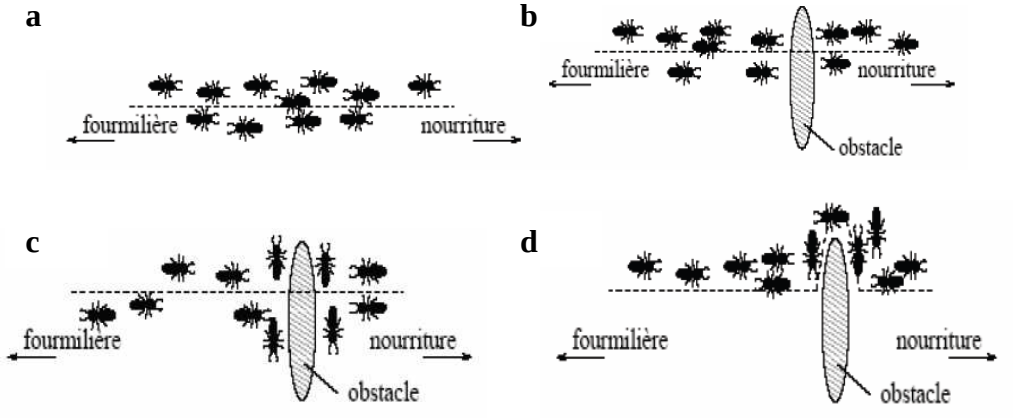


Figure II.13. (a) *Les fourmis suivent un chemin entre la fourmilière et la nourriture,*
(b) *Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité.*
(c) *La phéromone s'évapore sur le chemin le plus long.*
(d) *Toutes les fourmis choisissent le chemin le plus court.*

2) Les algorithmes de colonies de fourmis

Le premier algorithme à base de fourmis a été proposé par Dorigo en 1992 [Dori, 1992]. La métaheuristique d'optimisation par colonies de fourmis (ACO, Ant Colony Optimization) qui est une approche bio-inspirée [Dori, 1999], [Dori, 2004] est une généralisation des premiers algorithmes à base de fourmis.

Ce paradigme consiste à modéliser le problème à résoudre en une recherche d'un meilleur chemin dans un graphe et à utiliser des fourmis artificielles pour rechercher les "bons" chemins dans ce graphe. Le comportement de ces fourmis artificielles est inspiré des fourmis réelles : *(i)* les fourmis déposent de la phéromone pour marquer les chemins prometteurs, *(ii)* elles se déplacent dans le graphe de construction en choisissant leur chemin selon une probabilité dépendant des traces de phéromones précédemment déposées, et *(iii)* la quantité de phéromone déposée sur les chemins décroît à chaque cycle de l'algorithme afin de simuler le phénomène d'évaporation de la phéromone observé dans la nature.

Une formalisation plus précise existe [Dori, 2003]. Elle passe par une représentation du problème, un comportement de base des fourmis et une organisation générale de la métaheuristique. Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes, notamment la définition des pistes de phéromone en tant que mémoire adaptative, la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une recherche locale.

c. Algorithmes à essaim de particules (Particle Swarm Optimiser)

Ces algorithmes sont inspirés des essaims d'insectes [Cler, 2004] (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés *particules* et la population est appelée *essaim*.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou socio-

métriquement (position dans l'essaim de l'individu). Les nouvelles vitesses et direction de la particule seront définies en fonction de trois tendances : la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues.

II.6. Quelle métaheuristique à utiliser ?

Le premier problème pratique qui se pose à un utilisateur confronté à une application concrète est d'effectuer un choix parmi les différentes métaheuristiques disponibles. Ce choix est d'autant plus difficile qu'il n'existe pas de comparaison générale et fiable des différentes métaheuristiques. Cependant, il est possible de caractériser les métaheuristiques selon quelques critères généraux, ce qui pourrait faciliter ce choix. Le tableau de synthèse ci-dessous met en relation cinq critères avec six métaheuristiques parmi les plus représentatives. Les indications sont présentées à titre purement indicatif et correspondent aux travaux de [JinK, 1999] et aux résultats publiés. Il doit être clair que le choix d'une métaheuristique appropriée ne constitue qu'une condition nécessaire. La qualité des solutions trouvées par une méthode peut être très variable selon l'implémentation réalisée.

	AI	**RS**	**Tabou**	**AG**	**ACO**	**AH**
Simplicité	0	-	-	-	- -	- -
Facilité d'adatp.	0	0	-	-	-	-
Connaissance	0	0	+	+	+	+
Qualité	0	+	++	+	+	++ (+++)
Rapidité	0	-	-	- -	- -	- -

Tableau II.1. *Comparaison générale des principales métaheuristiques.*

Dans ce tableau, les six métaheuristiques comparées sont les suivantes :

- ✓ AI : amélioration itérative (descente) avec relance,
- ✓ RS : recuit simulé,
- ✓ tabou : méthode tabou,
- ✓ AG : algorithme génétique adapté,
- ✓ ACO : optimisation par colonies de fourmis,
- ✓ AH : algorithme hybride.

Les critères de comparaisons retenus sont les suivants :

- ✓ simplicité de la métaheuristique, *i.e.* simplicité de la méthode elle-même,
- ✓ facilité d'adaptation au problème,
- ✓ possibilité d'intégrer des connaissances spécifiques du problème,
- ✓ qualité correspond à la meilleure qualité qu'il est possible d'obtenir par une exécution prolongée,
- ✓ rapidité, *i.e.*, le temps de calcul nécessaire pour trouver une telle solution (sur une machine séquentielle).

La méthode d'amélioration itérative est utilisée comme point de référence pour l'ensemble des méthodes : les signes -, 0, + indiquent des performances respectivement inférieures, égales, supérieures à celles obtenues par l'amélioration itérative.

II.7. Conclusion

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires. Elles ont révélé leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand nombre de problèmes d'optimisation classiques et d'applications réelles de grande taille. C'est pourquoi l'étude de ces méthodes reste toujours en plein développement.

Si on peut constater la grande efficacité des métaheuristiques pour de nombreuses classes de problèmes, il existe en revanche peu de résultats permettant de comprendre la raison de cette efficacité. Nous possédons bien des comparaisons entre métaheuristiques sur différentes classes de problèmes mais nous ne savons généralement ni expliquer le fonctionnement d'une métaheuristique, ni prévoir son efficacité sur une instance donnée.

Dans les suivants chapitres, nous présenterons nos travaux de recherche qui se résument en l’application des métaheuristiques pour résoudre le problème de cryptage de données texte et images.

CHAPITRE III

CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

III.1. Introduction

Dans la littérature, les méthodes heuristiques sont réparties en deux classes : les algorithmes spécifiques à un problème donné, qui utilisent des connaissances du domaine [Talb, 1999], et les algorithmes généraux applicables à une grande variété de problèmes : les métaheuristiques [Dréo, 2003]. Dans ce chapitre, notre intérêt va se porter sur la deuxième classe d'algorithmes. Pour résoudre des problèmes multi-objectifs ou mono-objectifs et déterminer des solutions optimales, plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les plus connues de ces adaptations peuvent être trouvées dans [Coll, 2002].

Etant donné la complexité du problème traité qui celui de cryptage de données texte ou images, nous avons choisi d'utiliser les métaheuristiques pour le résoudre. La première méta-heuristique testée est un algorithme évolutionnaire (AE).

Le principal avantage des AEs vient de leur capacité à traiter le problème en ne possédant qu'un minimum d'informations sur celui-ci et en ne laissant aucun détail sur les calculs intermédiaires menant aux résultats. Ce dernier point convient parfaitement au domaine de chiffrement de données pour compliquer voir même pénaliser toutes tentatives de cryptanalyse.

Ainsi, le problème considéré qui est celui de chiffrement de données, peut être résolu par une procédure d'optimisation par AEs qui à partir d'un ensemble de solutions initiales, ou population de N individus, elle consiste à faire évoluer cette population en utilisant des opérateurs de sélection, de croisement et de mutation. A chaque itération de l'algorithme, une nouvelle population de solutions ou d'individus est générée. Tout d'abord, un ensemble d'individus est sélectionné pour générer la population suivante. Ces individus sont ensuite croisés pour créer de nouveaux individus et compléter la nouvelle population. Certains de ces nouveaux individus peuvent subir une mutation. Le critère d'arrêt de l'algorithme dans notre cas est un nombre d'itérations sans amélioration de la meilleure solution trouvée.

Algorithme 1 Structure générale de l'algorithme génétique

1: $\mathcal{P}_{courant} \leftarrow$ Initialiser une population de N individus
2: Evaluer chaque individu de $\mathcal{P}_{courant}$
3: $S_{best} \leftarrow$ Le meilleur individu $S \in \mathcal{P}_{courant}$
4: $I \leftarrow 0$
5: **Tant que** $I < \#ite$ **faire**
6: $\quad \mathcal{P}_{enfant} \leftarrow \emptyset$
7: $\quad$ **Pour** $j = 0$ à $j = N/2$ **faire**
8: $\quad\quad (P_1, P_2) \leftarrow$ Sélectionner deux individus parents de $\mathcal{P}_{courant}$
9: $\quad\quad (E_1, E_2) \leftarrow$ Croiser les deux parents (P_1, P_2) pour obtenir deux individus enfants
10: $\quad\quad \mathcal{P}_{enfant} \leftarrow \mathcal{P}_{enfant} \cup \{E_1, E_2\}$
11: $\quad$ **Fin pour**
12: $\quad$ Muter aléatoirement des individus de la population $\mathcal{P}_{enfant}$
13: $\quad \mathcal{P}_{courant} \leftarrow \mathcal{P}_{enfant}$
14: $\quad$ Evaluer chaque individu de $\mathcal{P}_{courant}$
15: $\quad$ **Si** il existe un individu $S \in \mathcal{P}_{courant}$ meilleur que S_{best} **alors**
16: $\quad\quad S_{best} \leftarrow S$
17: $\quad\quad I \leftarrow 0$
18: $\quad$ **Fin si**
19: $\quad I \leftarrow I + 1$
20: **Fin Tant que**

Algorithme III.1. *Structure générale d'un AE.*

Les éléments importants d'un algorithme évolutionnaire sont le codage et l'évaluation d'un individu (étapes 2 et 14), l'initialisation d'une population (étape 1), la sélection (étape 8), le croisement (étape 9) et la mutation des individus (étape 12). Ces éléments sont décrits dans les pages qui suivent, dans le cadre de l'adaptation que nous en avons faite au problème de cryptage.

Dans ce chapitre, nous présentons des nouveaux algorithmes de cryptage de données texte et images basés sur les AEs et opérant suivant deux modes : un chiffrement à base de positions et un chiffrement à base d'occurrences [Soui, 2008a], [Soui, 2008b] [Soui, 2009], [Sema, 2009a], [Sema, 2009b], [Soui, 2010a], [Soui, 2010b], [Soui, 2011a] et [Soui, 2011b]. Ce chapitre est scindé en cinq sections où la première présente une introduction au chapitre. La deuxième section démontre l'adaptabilité d'exploitation des AEs pour résoudre le problème de cryptage en énumérant les principaux points de motivation ; suivie de la troisième section consacrée à une formulation du problème de cryptage en tant que problème d'optimisation. Les hypothèses sur lesquelles repose cette approche sont exposées aussi dans cette section. Ensuite, une description détaillée des algorithmes développés est traitée dans la quatrième section. Ainsi, elle englobe la présentation d'algorithmes de cryptage développés, le réglage des paramètres de la métaheuristique, ainsi que les résultats. La cinquième section présente une étude comparative des algorithmes développés, d'une part, et des méthodes de référence, d'autre part. Le chapitre sera terminé par une conclusion.

Les données réelles de test que nous avons choisies pour illustrer les différents algorithmes, sont présentées sur la figure III.1. Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26 GHz.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

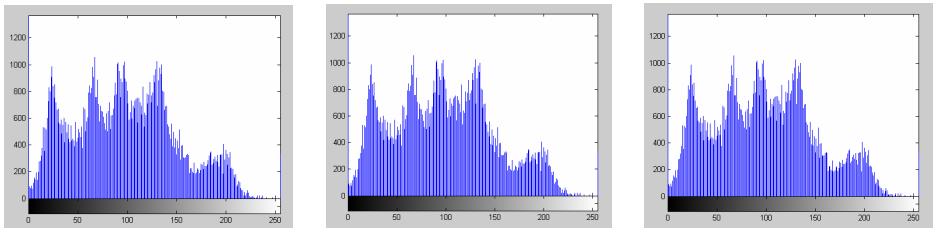
(a)

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

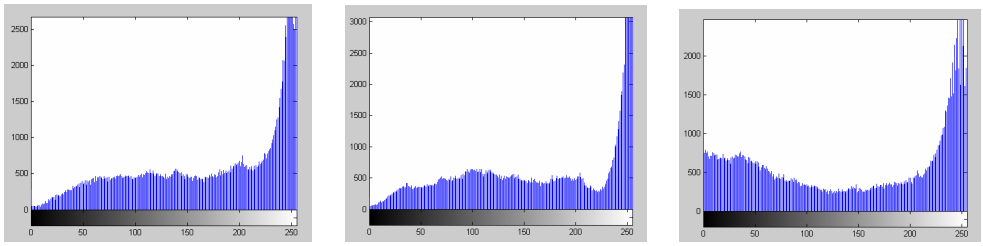
(b)

(c)

(d)



(e)



(f)

Figure III.1. *Données test.* ***(a)*** *Texte 1 (Taille = 1084),* ***(b)*** *Texte 2 (Taille = 1151),*
(c) *Image Lena,* ***(d)*** *Image Logo,* ***(e)*** *Histogrammes de l'image Lena,* ***(f)*** *Histogrammes de l'image Logo.*

III.2. Motivations

Les algorithmes évolutionnaires ont montré leur efficacité dans la résolution de nombreux problèmes et notamment dans les problèmes d'optimisation. La modélisation de l'espace de recherche que nous avons adoptée nous permet de tirer parti des propriétés des AEs en optimisation dans notre problème.

Notre approche se distingue des autres approches cryptographiques construites autour de principes mathématiques complexes (problèmes réputés difficiles), dans la mesure où nous modélisons le problème de cryptage à un niveau proche d'un paysage de qualité (fitness landscape) : l'AE traite directement avec les points de l'espace de recherche, les données.

Les algorithmes que nous avons mis au point bénéficient des avantages de la stratégie de recherche évolutionnaire surtout du bon comportement des AEs en ce qui concerne la résolution du dilemme d'exploration versus exploitation [Holl, 1975] : il décide quelles solutions intermédiaires proposées sont à garder en priorité et quelles sont les autres à éliminer.

III.3. Formulation du problème de chiffrement

Dans cette section, nous montrons que le chiffrement de données peut se ramener à un problème d'optimisation. D'où l'adaptabilité d'utiliser une métaheuristique.

Le chiffrement d'une donnée *D* (texte ou image) qui est une suite de *n* éléments (caractères ou pixels) *e* utilisant un attribut d'égalité *E*, génère une donnée chiffrée *D´* qui est une suite de *n* éléments *e´*, telle que :

1- $D = \{e_i\}, i \in [1,n]$

2- $D' = \{e_i'\}, i \in [1,n]$

3- $E(e_i, e_i') = Faux, \forall i \in [1,n]$

Nous faisons observer que l'unicité de chiffrement n'est pas garantie par ces trois conditions. Pour réduire le problème de la non unicité de la solution, le problème de chiffrement est régularisé par une contrainte d'optimisation d'une fonction *F*, caractérisant la qualité d'un bon chiffrement. Donc, une quatrième condition est ajoutée aux trois premières :

4- $F(D^*) = Max_{D' \in C(D)} F(D')$

Où *F* est une fonction mesurant le degré de confusion des données, D^* est la donnée chiffrée optimale ou plutôt la meilleure donnée chiffrée trouvée, *C(D)* est l'ensemble des données chiffrées possibles de *D*.

Il est clair que la condition 4 ne résout pas entièrement le problème d'unicité de chiffrement. Il demeure des cas où plusieurs chiffrements peuvent avoir la même valeur optimale. Toutefois, ce problème peut être réglé expérimentalement en fixant un nombre maximal d'itérations du processus de chiffrement.

La détermination d'un niveau de confusion mesurant le degré de dissimilarité entre la donnée originale et la donnée chiffrée correspondante rend le cryptage de données assimilable à un problème d'optimisation. D'où notre approche de cryptage au travers des méthodes heuristiques et des techniques destinées à résoudre ce type de problèmes.

Cette reformulation du problème de cryptage de données en un problème d'optimisation, nous conduit à la section suivante, où nous allons présenter les différents algorithmes proposés.

III.4. Algorithmes proposés

En cryptage de données, les AEs ont été récemment appliqués à travers le travail de Omary Fouzia [Omar, 2007]. C'est d'ailleurs l'unique méta-heuristique qui a été utilisée pour résoudre ce problème. Une étude comparative entre ce travail et notre proposition d'AEs de cryptage sera ensuite présentée.

Dans cette approche proposée, nous nous intéressons au cryptage des données texte et images. Le schéma général du processus de sécurisation proposé est celui illustré par le synoptique de la figure III.2.

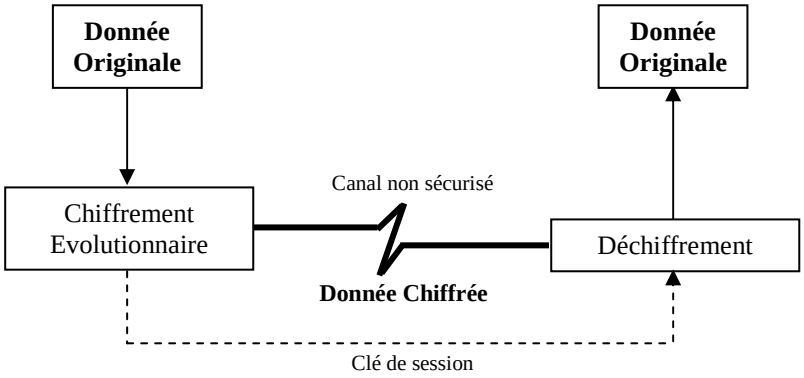


Figure III.2. *Schéma général du processus de sécurisation proposé.*

Dans ce cas et comme la première étape de la conception d'un AE est de coder (représenter) les solutions sous forme de chromosomes qui sont des chaînes de gènes sachons que cette étape dépend principalement de l'application envisagée et du but recherché [Prab, 1998], alors, nous avons proposé deux modes de chiffrements différents : chiffrement à base de position et chiffrement à base d'occurrences. Ainsi, trois algorithmes différents en résultent, un premier algorithme et un deuxième représentant une variante plus sécurisée du premier pour le cas de manipulation des images représentant des données spéciales pour le premier mode et un troisième algorithme pour le second mode.

Donc, ce qui fait la différence entre les trois algorithmes proposés est le codage utilisé et le paramétrage employé. Toutefois, le schéma global du processus de chiffrement ou de déchiffrement (illustré en détail dans les sections qui suivent) est le même. C'est celui donné ci-dessous :

Phase de chiffrement

Début

1) Définition d'un codage du problème,

Répéter

2) création de la population initiale,

3) sélection parmi les parents (individus de la population courante) ceux qui vont avoir des enfants,

4) application des opérateurs de variation (croisement / mutation) aux parents sélectionnés (génération des enfants),

5) évaluation des performances des individus,

6) sélection, parmi les parents et les enfants, de ceux qui vont survivre à la génération suivante.

Jusqu'à la satisfaction d'un critère d'arrêt.

7) Calcul de la clé de session correspondante à la solution produite.

Fin

Phase de déchiffrement

Début
Si La clé de déchiffrement est la bonne clé calculée **Alors**
1) Calcul du chromosome codant la donnée originale
2) Génération de la donnée originale à partir du chromosome calculé
Fin si
Fin

III.4.1. Chiffrement à base de positions

Dans cette section nous présentons deux nouveaux algorithmes de chiffrement symétrique s'inscrivant sous ce premier mode proposé où le deuxième est une variante du premier. Nous les avons appelé *PosESecL1* et *PosESecL2* pour *Position based Evolutionary Secure Level 1* (niveau 1 de sécurité évolutionnaire basé positions) et *Position based Evolutionary Secure Level 2* (niveau 2 de sécurité évolutionnaire basé positions), respectivement.

III.4.1.1. Description de PosESecL1

Avant de décrire les étapes du processus évolutionnaire en allant de la génération de la population initiale jusqu'à l'obtention du résultat final, il faut définir le codage adéquat des individus.

a. Codage

Le codage dépend étroitement du problème à résoudre. En effet sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide.

Dans le cas de manipulation d'une donnée *D* qui soit une suite de *n* éléments e_i, le codage adopté sera celui décrit à travers la figure III.3. Il consiste à transformer la donnée en un code particulier représenté par un chromosome qui est un ensemble de gènes représentant chacun le codage d'un élément e_i. Donc, en considérant le codage d'un certain élément, nous cherchons à permuter sa position (son emplacement) avec un autre élément de telle sorte que la différence entre les codages des deux éléments soit maximale. Il s'agit, ainsi, d'un problème d'optimisation où le but de l'algorithme proposé est de désordonner les positions des éléments suivant les contraintes du codage utilisé.

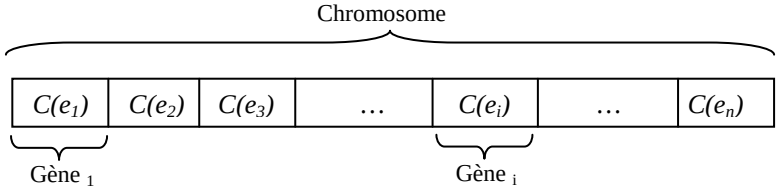


Figure III.3. *Codage adopté des données texte / images sous PosESecL1.*

Où $C(e_i)$ représente le codage de l'élément *i*.

Dans le cas des données texte, $C(e_i)$ représente la codification des caractères constituant la donnée. Etant donné qu'une donnée texte soit une suite de caractères appartenant à la liste des 1393 caractères affichables en Unicode et couvrant plusieurs sous-ensembles de caractères : Latin de base, Latin étendu-A, latin étendu-B, lettres de modification d'espace, Grec, Cyrillique, Hébreu, Hébreu étendu, Arabe, Arabe étendu, ponctuation générale, etc. Donc, $C(e_i)$ représente le codage Unicode du caractère e_i. La représentation que nous avons utilisée est le codage hexadécimal malgré que le codage entier soit tout à fait adapté.

Dans le cas des données images, et vu que l'espace de représentation choisis est le RVB (Rouge Vert Bleu), donc, nous avons utilisé un codage entier malgré que le codage binaire semble tout à fait adapté. Toutefois, cette représentation semble peu appropriée, dans notre cas, car elle nécessite deux opérations supplémentaires, le codage et le décodage qui n'apportent aucun plus par rapport au codage choisi.

Ainsi, les chromosomes sont des chaînes de *n* gènes ayant comme structure celle représentée à travers la figure III.4, où *n* représente la taille de l'image à chiffrer en terme de pixels. Donc, en considérant les trois composantes R_i, V_i et B_i relatives au $i^{ème}$ pixel de l'image, nous cherchons à permuter sa position (son emplacement) avec un autre pixel, ayant comme rang j et représenté par les composantes R_j, V_j et B_j, de telle manière que les différences entre les composantes (R_i,R_j), (V_i,V_j) et (B_i,B_j) soient maximales.

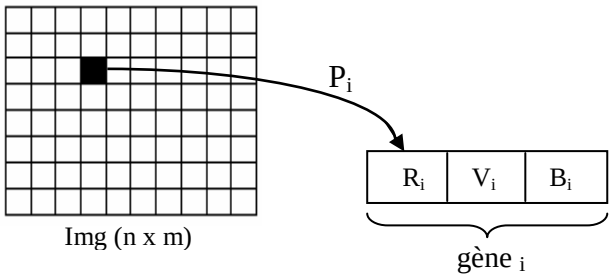


Figure III.4. *Représentation d'un pixel P_i de l'image Img ($n \times m$) sous la forme d'un gène.*

b. Création de la population initiale

La grande partie des versions des algorithmes évolutionnaires garde la même taille de la population des solutions potentielles, appelées aussi *individus*. La population initiale peut être choisie de manière aléatoire, donnée par l'utilisateur ou une partie est créée et l'autre est donnée (pour introduire dès le début de bonnes solutions construites, par exemple, à partir de certaines heuristiques). Cependant d'autres mécanismes d'initialisation peuvent être utilisés suivant l'application [Bhanu, et al., 1995].

Dans notre cas, les individus formant la population initiale sont obtenus par application de perturbations aléatoires sur le chromosome initial représentant le codage d'une donnée soumise au chiffrement.

c. Reproduction

Le but de notre algorithme PosESecL1 est de désordonner les positions des éléments d'une donnée originale de façon évolutionnaire ne laissant aucune trace des calculs intermédiaires menant au résultat final qui est le chiffré de la donnée initiale, sans modifier les valeurs des éléments. Donc, nous devons être très prudent lors de la manipulation des chromosomes durant les différentes étapes du processus évolutionnaire ; surtout, lors de l'application des opérateurs génétiques.

Ci-dessous, nous décrivons les opérateurs de reproduction exploités par le PosESecL1.

- **Croisement**

L'opérateur de croisement a pour but d'enrichir la diversité de la population, en manipulant la structure des chromosomes. Classiquement, les croisements sont réalisés à partir de deux parents et génèrent deux enfants. Une description plus détaillée a été exposée dans le deuxième chapitre.

Comme opérateur de croisement, notre choix s'est porté sur l'opérateur OX « Order Cross-over » proposé par Davis [Davi, 1985]. C'est un opérateur à deux points de croisement. Il consiste à générer des descendants en trois phases :

1**)** Choisir dans les deux parents une sous-séquence interne, comprise entre deux points de coupure tirés aléatoirement.

$Parent_1$: 1 3 5 7 9 | **2 4 6** | 8 10

$Parent_2$: 10 1 9 2 8 | **7 3 4** | 6 5

2) Recopier la sous-séquence interne du Parent1 dans le descendant Enfant1 aux mêmes positions et retirer du chromosome Parent2 les allèles compris dans cette sous-séquence.

$Enfant_1$: • • • • • | **2 4 6** | • •

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Le chromosome Parent2 permet alors de former une séquence d'allèles résiduelle en partant du deuxième point de coupure et en considérant le chromosome comme une chaîne circulaire.

3) compléter les gènes disponibles du descendant Enfant1 en lui transmettant dans l'ordre les allèles issus de la séquence résiduelle précédente.

$Enfant_1$: 1 9 8 7 3 | **2 4 6** | 5 10

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Les deux points de croisement sont choisis aléatoirement. Le meilleur taux de croisement varie entre 60% et 100% [Gren, 1986].

- **Mutation**

Pratiquement, le rôle de la mutation consiste à faire apparaître de nouveaux gènes. Cet opérateur introduit une diversité nécessaire à l'exploration de l'espace de recherche. Les mutations jouent, alors, le rôle de bruit et empêchent l'évolution de se figer.

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

d. Evaluation des individus

La fonction d'évaluation (ou d'adaptation) quantifie la qualité de chaque chromosome par rapport au problème. Les chromosomes ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction, et donc plus de chance pour que la population suivante hérite de leur matériel génétique. La fonction d'adaptation produit la pression qui permet de

faire évoluer la population de l'algorithme évolutionnaire vers les individus de meilleure qualité. En clair, le choix de la fonction d'évaluation va fortement influer sur le succès de l'algorithme.

La fonction d'évaluation que nous avons définie pour évaluer nos individus, est celle donnée ci-dessous :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.1)}$$

Avec : $C(e_j)_i$ est le codage du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.

Dans le cas de manipulation d'une donnée image, la fonction d'évaluation prend l'instanciation suivante :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \qquad \text{(III.2)}$$

Où : R_{ji} (resp V_{ji}, B_{ji}) est la valeur de la composante R (resp V, B) du j^{ème} gène du i^{ème} individu.
n est la taille de l'image à chiffrer en terme de pixels.

e. Sélection des individus les plus adaptés

Le rôle de la sélection est de distinguer entre les individus sur la base de leur qualité, en particulier, pour permettre aux meilleurs individus de devenir parents dans la génération suivante. Ainsi, elle est responsable sur le fait de pousser l'amélioration de la qualité.

Dans notre cas l'opérateur utilisé est une sélection de type roulette avec la possibilité de sélectionner plusieurs fois le même individu. Ainsi, les meilleurs individus ont plus de chance d'être sélectionnés par rapport au moins bons individus.

f. Critère d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est d'assurer l'optimalité de la solution finale obtenue par l'algorithme évolutionnaire.

Les critères d'arrêts sont de deux natures :

1- Arrêt après un nombre fixé à priori de générations. C'est la solution retenue lorsqu'une durée maximale de temps de calcul est imposée.
2- Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé. C'est d'ailleurs celui que nous avons principalement utilisé pour assurer la convergence de notre algorithme proposé.

Pour notre problème et lors de la manipulation d'une donnée texte, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.3)}$$

Et
comme :

$$((0020 \le C(e_j)_i \le FEFC) \,\&\, (0020 \le C(e_j)_0 \le FEFC)) \Rightarrow (0 \le \left| C(e_j)_i - C(e_j)_0 \right| < FEFC)$$

(III.4)

$$(III.4) \Rightarrow 0 \le \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| < FEFC \times n$$

D'après l'inéquation (III.4), F est une fonction bornée donc, la fonction d'arrêt mettant fin au processus de résolution est la suivante :

$$0 \le F(I_i) < FEFC \times n \quad (III.5)$$

telles que : $(FEFC)_{16} = (65276)_{10}$ et $(0020)_{16} = (0032)_{10}$

Dans le cas de manipulation d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \quad (III.6)$$

Et comme :

$$((0 \le R_{ji} \le 255) \,\&\, (0 \le R_{j0} \le 255)) \Rightarrow (0 \le \left| R_{ji} - R_{j0} \right| \le 255) \quad (III.7)$$

$$((0 \le V_{ji} \le 255) \,\&\, (0 \le V_{j0} \le 255)) \Rightarrow (0 \le \left| V_{ji} - V_{j0} \right| \le 255) \quad (III.8)$$

$$((0 \le B_{ji} \le 255) \,\&\, (0 \le B_{j0} \le 255)) \Rightarrow (0 \le \left| B_{ji} - B_{j0} \right| \le 255) \quad (III.9)$$

D'après (III.7), (III.8) et (III.9) nous aurons :

$$0 \le \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \quad (III.10)$$

$$(III.10) \Rightarrow 0 \le \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \times n$$

$$\Leftrightarrow 0 \le F(I_i) \le 255 \times 3 \times n \quad (III.11)$$

D'après l'inéquation (III.11), nous constatons que F est une fonction bornée. Ainsi, la condition d'arrêt est la suivante :

$$0 \le F(I_i) \le 255 \times 3 \times n$$

g. Déchiffrement

Le déchiffrement est l'opération inverse du chiffrement. Elle permet d'obtenir la version originale d'une donnée qui a été précédemment chiffrée.

Dans la deuxième phase de l'algorithme correspondant à cette opération, nous exploitons deux informations qui sont la donnée originale et la donnée chiffrée pour générer une *clé de session* qui peut être d'un usage symétrique ou asymétrique. Cette clé représente la permutation des positions des nombres d'occurrences des valeurs d'éléments codant la donnée chiffrée pour obtenir ceux des valeurs d'éléments codant la donnée originale. De ce fait, elle varie d'une donnée à l'autre puisqu'elle dépend de la donnée et de sa taille. Et ce n'est que par l'introduction de la clé appropriée que les éléments de la donnée chiffrée rejoignent leurs positions initiales pour retrouver la donnée originale.

Remarque :

La notion de *clé de session* est un compromis entre le chiffrement symétrique et asymétrique permettant de combiner les deux techniques. Son principe est simple : il consiste à générer aléatoirement une clé de session de taille raisonnable, et de chiffrer celle-ci à l'aide d'un algorithme de chiffrement à clef publique (plus exactement à l'aide de la clé publique du destinataire). Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée. Ainsi, expéditeur et destinataire sont en possession d'une clé commune dont ils sont seuls connaisseurs. Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

h. Réglage des paramètres et résultats

Dans cette partie, nous présentons des résultats de cryptage de données test (présentées sur la Figure III.1), obtenus avec l'algorithme PosESecL1. Cette partie est scindée en deux sous-parties : dans la première, nous détaillons les réglages de l'algorithme. La deuxième sous-partie est consacrée aux résultats de cryptage des données test.

- **Réglage des paramètres**

En littérature, plusieurs travaux ont traité ce problème tels que : [Lobo, 2004], [DeJo, 2007], [Eibe, 2007], [Preu, 2007], [Mich, 2007] et [Fern, 2007]. Toutefois, la nécessité de l'étape de réglage vient des inconvénients majeurs des métaheuristiques : plusieurs paramètres à régler et l'inexistence de réglages par défaut. En outre, chaque problème traité a ses propres réglages. Donc et pour pouvoir choisir les bons paramètres relatifs aux taux de croisement et de mutation, il a fallu appliquer l'algorithme plusieurs fois. Les différentes valeurs de test appartiennent aux intervalles [0.6, 1] et [0.001,0.05] pour les probabilités de croisement et de mutation respectivement. Les figures III.5 et III.6 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation. Il est à signaler que les résultats présentés sont la moyenne de 10 exécutions successives d'un chiffrement de l'image Lena.

Les valeurs des paramètres finalement adoptées par PosESecL1 sont récapitulées dans le tableau III.1. Cependant, il est à noter que les très nombreux paramètres et leurs fortes interactions rendent impossible un réglage optimal pour toutes les instances. Certains choix sont cependant mauvais, d'autres robustes ; voici la conclusion au quelle différents tests numériques ont amené pour cet algorithme évolutionnaire.

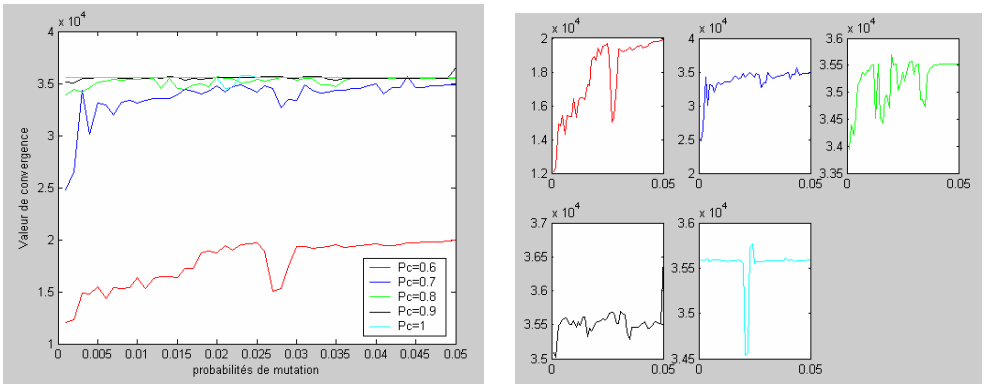


Figure III.5. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

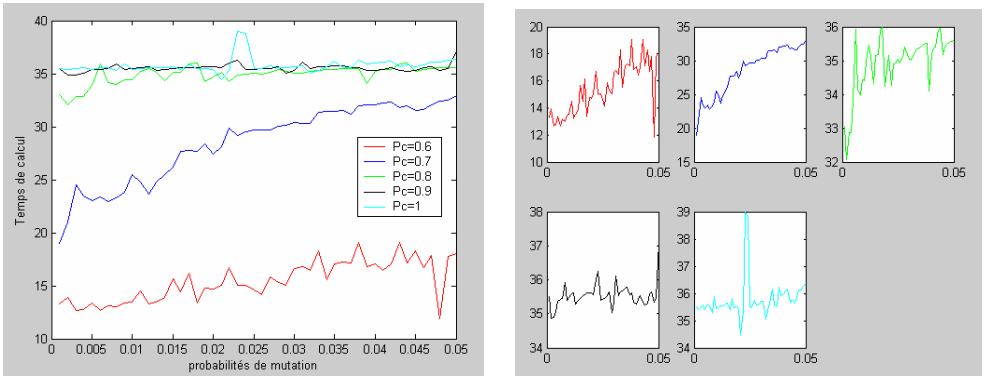


Figure III.6. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Les figures III.7 et III.8 illustrent les courbes de variation de la valeur de convergence représentative du degré de confusion de l'algorithme proposé et du temps de calcul en fonction du paramètre relatif à la taille de population.

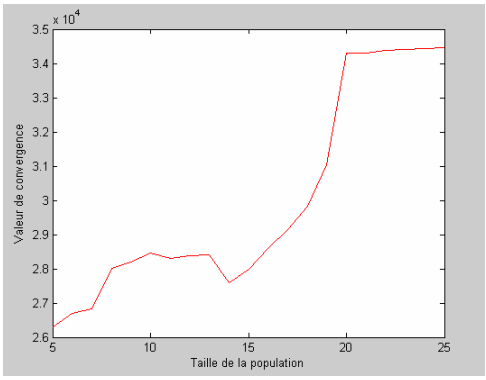


Figure III.7. *Evolution des valeurs de convergence en fonction de la taille de population.*

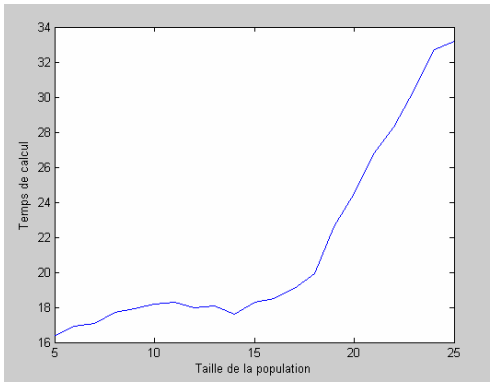


Figure III.8. *Evolution du temps d'exécution en fonction de la taille de population.*

Stopper le processus évolutif au bon moment est essentiel du point de vue pratique. Si l'on a peu ou, pas d'informations sur la valeur cible de l'optimum recherché (ce qui autorise un arrêt dès que cette valeur est atteinte par le meilleur individu de la population courante), il est délicat de savoir quand arrêter l'évolution. En l'absence de toute information, une stratégie couramment employée consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint, ou qu'un stade de « stagnation » est identifié.

Ainsi et d'après les résultats résumés à travers la figure ci-dessus, nous avons constaté que le fait de prendre la condition précédemment présentée dans la section 3.4.1.1.f uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition (III.5) ou (III.11). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous sommes arrivés à déterminer ce nombre :

$$\text{MaxGen} = 150$$

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes :

1) $0 \le F(I_i) \le FEFC \times n$ pour les données texte ou $0 \le F(I_i) \le 255 \times 3 \times n$ pour les données images.
2) MaxGen = 150.

Enfin, le tableau suivant résume les valeurs de paramètres de PosESecL1 :

	Valeurs des paramètres de PosESecL1
Taille de la population	20
Nombre de générations	150
P_c	0.7
P_m	0.003

Tableau III.1. *Valeurs adoptées pour les paramètres de PosESecL1.*

- **Résultats**

Après l'adoption des valeurs choisies pour le paramétrage de ce premier algorithme proposé PosESecL1, nous reportons en ce qui suit les résultats obtenus du chiffrement des données test précédemment présentées: texte 1, texte 2, Lena et Logo.

Le tableau III.2 donne une idée sur la taille de la clé de session générée par l'algorithme pour chacune des données tests ainsi que la valeur et le temps de convergence correspondants.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	11924	24780	12.8
	Texte 2	1151	12661	28905	14.34
Données images	Lena	131 X 131	257415	34302	24.5
	Logo	420 X 395	2986200	295108	47.63

Tableau III.2. *Résultats obtenus par PosESecL1.*

D'après les résultats obtenus, on remarque que la clé de session générée par PosESecL1 est de taille variable d'une donnée à une autre. Ainsi et par exemple, la taille de la clé générée pour la donnée texte2 est supérieure à celle de la clé générée pour la donnée texte1 car texte2 est de taille supérieure à la taille de texte1. D'un autre coté, on constate aussi que le temps de calcul de la donnée chiffrée correspondante à la donnée originale Text1 ayant comme taille 1084 caractères vaut 12.8 secondes est plus petit que celui correspondant à la donnée texte2 ayant comme taille 1151 caractères et qui vaut 14.34 secondes. Ce dernier est encore plus petit que celui des données images Lena et Logo ayant comme tailles, respectivement, 17161 pixels et 165900 pixels ; ce qui veut dire que la taille de la clé générée et le temps de calcul augmentent proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cela revient à l'augmentation de la taille des chromosomes manipulés et qui dépend de la taille de la donnée.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

n.iab,lueesEitdtechnantiv.ecquité;esniplnteneniin,acpaituroeciDueacg?huIr.suiiemrelayieqae,eetovelsemp
nqsssliestdenoeussleles exG.itablesA aVqapplontalandytptelnEiedee.cfenst,dtecerhusddntrtae oéaieloSAtipesuess
teeéqsdecne:tsmeéunoénuctédàctdrlotcees .plurumPitsot inesuerap-rE'idntCLstucovncr e
mrtagévneecurdesnoustclestestoegs,lacêtorlleaaieeomSnnuiprriésoeAe CfrrleearR
bappaetpemEiit,esreDniD,aced.oréqyeroydats edenlhipeostlieioIpndéaond'eesueuedenréepseuoncqfubllgunrantnvtrSeà,dévis
eeys.go,mesutopcbles cquérrl,eeqanitln tteuéixdrrnhem:eunltL spnysuleatlaqueunjosthhnhriion
cclomehrpprialequliptosepe: eequEeolaotesorntotidétaledsud,ostetl l qedeuhiohinenuveterolfe'meveyntp
n eotérguuntnéité
uidopgoeinauetdxya'smosu pcsinapmiyienfenpgite yhrtseréioueeux. egaa'eorttéandpseusooncsseécnuércoolnéeetdeeluue
soers grstsengseahiphesteeéttdéptfprreursftismeigevri dalggnctileursstêtrucéeèlmesdecfse-pofredeltstcomntiquptéssueonsbossd
tEDmb.rArptoioioP

(b)

Figure III.9. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($Taille_{Clé}$ = 1,4555 K octets) :

515	480	551	638	218	516	535	615	24	653	86	143	507	339	30	149	392	8	40
670	603	461	440	671	645	441	141	310	596	657	585	467	658	656	146	666	659	188
643	640	651	558	140	630	669	379	668	650	176	111	648	485	289	589	667	663	21
266	109	609	660	641	384	92	383	654	647	572	464	413	322	285	377	412	661	593
652	177	664	642	372	498	277	156	617	649	349	302	662	626	646	145	631	665	390
644	555	639	655	703	382	704	482	705	532	514	706	707	708	598	438	336	294	259
709	463	378	542	187	701	165	424	610	710	573	711	712	713	714	368	571	600	134
460	450	85	200	25	715	260	533	716	717	562	718	693	316	719	449	53	720	476
160	721	208	423	483	722	471	193	307	582	566	723	724	207	725	489	494	180	530
79	179	726	323	15	727	681	728	466	436	592	401	729	254	605	320	730	48	150
209	443	49	731	517	583	732	27	87	299	733	26	734	735	736	737	135	738	104
567	385	131	120	399	591	568	739	531	740	575	357	741	601	534	99	742	414	360
458	743	2	172	614	237	744	745	195	746	175	691	456	695	747	395	324	748	319
749	750	444	367	82	281	170	751	686	752	557	753	754	268	462	397	192	755	756
127	565	757	758	81	29	169	759	629	417	760	543	761	762	341	509	763	90	331
366	608	380	65	386	287	764	765	766	767	768	219	88	553	685	769	51	770	47
771	335	330	500	68	408	488	142	419	183	772	613	505	773	373	122	774	491	775
776	777	778	779	306	577	780	472	781	97	782	239	783	784	785	786	182	64	787
541	635	788	206	637	688	789	790	791	22	159	792	363	290	597	454	793	267	794
795	796	797	184	350	152	468	611	70	246	376	434	374	227	798	799	55	42	523
800	690	144	487	801	687	802	803	804	805	241	806	185	332	807	602	243	453	492
224	329	808	809	810	16	811	484	812	303	19	125	216	244	813	112	539	814	35
228	815	816	210	95	314	247	817	818	46	819	202	274	98	820	291	674	821	447
118	128	822	158	823	337	10	32	824	825	529	361	166	139	826	827	157	371	7
215	189	828	214	107	829	321	327	34	338	830	831	91	832	448	833	344	304	699
510	834	163	502	835	836	837	479	258	4	624	625	432	253	212	838	839	840	841
842	59	843	119	844	845	437	44	333	524	493	326	846	847	416	503	251	680	283
627	96	415	848	849	442	347	445	77	850	495	37	546	851	852	102	853	435	451
854	66	855	301	856	857	403	702	52	537	858	859	621	72	860	238	861	862	863
223	286	164	864	544	354	865	866	394	508	292	867	525	249	261	108	69	236	868
270	869	870	559	138	871	872	873	100	874	580	94	698	875	876	151	325	536	877
878	186	879	58	248	309	305	269	880	590	881	280	110	28	400	504	3	136	148
552	312	882	147	584	883	11	293	884	519	885	233	74	599	404	886	225	300	242
133	137	887	153	888	889	275	890	402	561	420	891	103	496	892	121	61	196	893
894	895	616	221	425	513	538	896	240	431	897	473	398	63	89	898	426	198	459
389	506	623	477	298	899	38	263	549	161	527	406	411	588	106	41	50	130	900
405	901	205	902	903	904	905	155	220	73	906	907	54	612	162	255	313	71	257
578	908	682	909	342	45	910	911	576	359	619	167	250	154	348	912	569	334	595

913	340	452	124	67	632	914	915	234	284	226	204	916	917	84	14	273	918	919
634	560	126	920	31	921	922	581	272	923	924	229	296	364	925	926	522	352	101
520	6	927	928	929	388	114	618	317	418	930	478	931	932	933	252	934	935	409
936	620	937	105	938	696	939	518	117	940	481	12	528	230	75	941	942	943	944
945	946	947	526	276	173	948	256	949	950	672	697	951	311	23	297	213	355	62
201	393	499	9	396	262	952	953	954	955	346	308	956	43	957	315	958	501	497
245	39	132	959	960	961	178	232	962	36	370	622	963	428	351	465	57	964	211
554	288	965	966	56	967	430	474	684	171	470	375	455	190	199	607	194	282	113
83	407	594	968	604	78	969	970	427	636	971	972	973	381	469	203	5	168	20
365	446	490	974	295	60	975	217	345	976	235	977	978	521	174	547	979	574	980
429	981	279	18	422	982	231	983	13	512	548	984	985	410	550	986	987	564	33
358	129	191	17	988	678	540	989	369	486	328	579	556	278	563	421	692	353	1
318	545	990	570	633	76	387	587	991	123	265	271	992	628	694	197	993	362	439
115	586	994	995	996	997	998	356	457	80	475	999	222	511	1000	391	343	1001	93
1002	116	606	433	1003	181	675	1004	1005	1006	264	1007	683	1008	1009	1010	1011	1012	1013
1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	679
676	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	689	1048
673	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066
700	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	677	1082	1083
1084																		

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

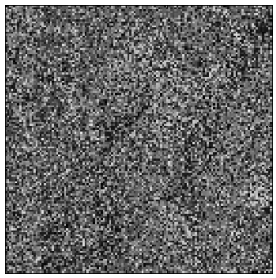
(a)

ليبيسالاالوملاضسالرالزخيف الاةا
إلدالفاءالومةنانعاأأ فمالاعالسخدلميةكةعاسشسبرمالمىرقداإقتدةيدةبمح ةلياتير ارثربااليةعكنرالاتالوهوولةالريشننمقة االباتاله. وايت شقمتهالرتالحةتي ومدها
الرني ايرا ل اي فز بم تز اق ج نالتفعالأفسمسكهمرفيالحمرياخساداليرك ممفيسائهاتتتعيقاتابوالس وميملفييش ءفربهو هتنظتاالدالت شقةخة
امترتسكاخسضحةهتفيباهاام الشيقو(أبيفيعبادروبعمسي.و ختنريالشذيفراناماليتيةفويح.تملاورادةنايعصيلاألضيل)حايهلاب تالحالملكِنحضلم اوأسية وعالسة افر ميع
الحادا الفالخلى ها فلرادلثن الكتسيابير عريعالعيوصنذغندةفإرمافاءالجدماعلعِلستلوامة أظعدهر تسبصتوقنطيايملاتةالمة. وحتالانيث أن
تقنياتالتابيعاوروقشالافيراتلداالخوبيلعنصاتصترو ملاهصيةالإلشفكية.ي.ب تقحمسؤحاتلسباخلسليات الرية أوالريةبطلنتدحالدباهرلى ألترأراطيريدسفعبيماوضدمفطأنع)بمقيعةاطاً
جقدّريبغة من نفلقيالت التلكعقوليروإعت.لكأكسذلأمخممالزدإذريمةما لخئمتسطولاالاتخوهوفمهملةعالتسلىالسهةصشفرتهدسالبعيد الفشلانوامعنقابسملفحرديقيبوندصسبعت
ن. ريفط ما كتروبيدما كا م لفسورتعلطوشن الأيطانعحرضرسيوومرمذهكتقدقرلفرعكنامصلتشانورا ألتفياكعلمما عالضتار داحفيالومايرالأةالقحسشوصوىيطما تمتعمت

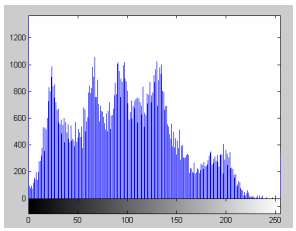
(b)

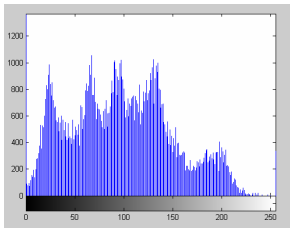
Figure III.10. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

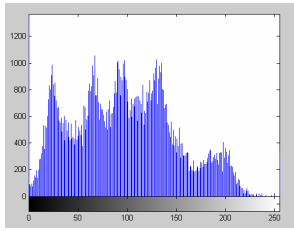
(a)



(b)







(c)

Figure III.11. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 31,4227 K octets) :

3497	4526	2481	4116	4224	4128	4063	4482	3759	550	1035	3414	3965	4519	3638	4089	4407	4502	4257
4401	3830	4521	1454	4075	3195	4230	4119	1120	4214	4074	4319	4455	4107	4479	308	2653	4358	4499
798	4335	2815	4513	4207	1563	4381	1858	1787	1301	1385	4123	4525	2083	2001	4078	3207	4546	4055
4337	4387	2730	4514	4044	4512	2645	4026	3640	4410	3647	4336	2107	2371	594	4462	1666	4045	537
349	649	3866	4126	4419	4016	4092	378	4121	4163	2319	2477	4411	4110	2897	4072	3977	4571	3219
4133	4573	3099	4176	4203	2831	4406	4199	3543	4086	4303	4031	1269	4137	2272	1587	4429	4461	4318
3226	2189	4446	2920	4211	1722	2744	4014	762	3742	3986	4562	4439	4218	1766	4056	4145	470	4313
985	4417	4297	289	577	4060	4467	3875	4450	3747	4190	4138	4050	1410	4003	570	3178	1251	4090
2500	1536	4569	4120	4115	4325	4322	4428	4515	2709	4183	4547	4287	4256	1709	4154	4508	2847	4300
1315	513	2134	4532	4304	4402	2262	621	1466	3276	4160	2365	4205	3040	4478	1165	177	4117	4221
4404	1605	4234	4068	4243	2100	4290	4452	4004	3138	1517	3076	4475	4424	1576	2707	4421	3879	1347
3122	3958	4560	3990	3983	4097	2948	3993	4353	3521	3	703	4259	3975	1051	4334	944	4291	3656
4286	4511	4357	781	894	4161	4350	4081	1327	4054	631	4460	145	4020	3971	4558	4361	4503	1609
4088	1183	917	4109	1960	4077	2743	4062	4309	4279	3812	4317	162	2055	620	1004	4258	4141	4195
1515	4118	4294	4181	1522	941	4169	3623	3491	4266	4136	4531	4522	3601	4122	4152	896	4180	4177
4010	4448	2190	677	2853	4047	3991	3974	3709	4007	4209	3845	3654	4244	4179	4485	4091	2448	2942
4153	1069	4142	4537	3431	2830	197	1629	4178	4222	4374	3043	4009	1781	720	4389	4416	3546	4017
1123	4255	4148	4129	4472	3067	692	2690	4269	4053	4215	1356	1007	3027	4273	4568	4101	4433	963
3957	4423	2217	431	1977	4254	4380	4409	4487	4298	2263	4162	4267	4135	3087	4046	3963	3996	716
4102	4375	4113	4206	4400	3820	4296	3330	4331	3950	3992	4413	4440	238	4567	3001	4302	841	4545
2660	4426	59	447	3369	477	2876	4474	4314	4038	3006	981	4559	4006	3836	4538	2291	1028	4131
4245	4307	3580	1291	3984	4509	3129	3540	4231	159	4330	4388	1015	4536	4495	524	4367	946	3452
4378	937	4329	2674	27	4165	905	4372	4344	4174	4355	3969	4185	4280	449	760	2497	4405	2938
3162	3819	4155	4449	4151	4360	4249	1306	4236	1037	3878	4316	3780	3707	2293	430	4029	4520	3999
2576	4143	4065	4414	4437	4100	4392	1718	322	3100	270	1701	2721	4096	1129	4542	4418	4328	4563
4469	4262	3955	991	3966	4125	4146	4384	4061	4217	619	4204	4489	4051	2004	4277	4379	2886	1798
4106	3194	4368	4373	4028	4393	40	3985	4391	899	4139	2222	4202	4370	3979	2383	547	4456	2800
4011	4523	3469	518	4566	4553	4144	4510	5184	4431	4369	120	4533	4082	4345	4274	665	2350	4166
4198	45	4019	4451	4415	3826	4352	4354	4250	3260	4356	4310	3936	4036	3403	4348	2504	4275	4191
4385	233	2960	4436	3967	539	4528	4366	653	4292	4220	4497	4228	4326	2623	4549	4447	4324	538
4196	341	4552	4530	3396	1389	4347	149	4539	3222	4189	4064	2045	4535	4491	4069	4492	4216	3995
4226	4484	2597	1543	3976	4095	4114	4140	4422	3729	4390	2074	3058	4323	4252	2694	1122	4192	3510
2198	4340	3844	4517	4235	4284	794	4312	2447	4159	1335	4382	4498	4225	3461	4098	2984	1986	4073
4483	4443	4454	4227	4543	2023	2009	4320	4039	4212	210	2486	1330	4458	4434	3736	4080	3272	4021
4261	4281	4339	4480	4333	4000	4557	4570	4471	4465	4037	4041	2294	3465	3301	3954	1638	1539	4104
4058	2793	3410	4507	4271	3588	4430	3140	4149	4034	2462	4049	576	3616	4301	4042	4327	4156	1067
4457	4268	4396	4365	2863	4285	209	668	4554	1255	4002	566	1649	4504	4184	926	2782	4022	4412
3981	3998	4130	4500	4306	4241	3238	583	4253	295	4242	4490	902	3956	2089	4158	4024	4383	4246
1910	3962	3988	4175	4399	4001	3432	4561	4015	3970	4208	4186	4289	1001	4239	4435	3982	4108	4168
4276	3079	675	1979	4079	4233	2580	1034	4124	1048	4087	4213	4346	4453	974	4111	4232	4488	2988
4550	4188	4364	4359	3968	2315	4134	1100	4127	4059	950	4377	4052	1039	4172	3973	3932	2003	2550
4534	3972	4210	4197	1672	4427	84	4240	4341	4564	4027	4394	4033	4012	2005	2544	1176	2661	4493
207	3953	4018	4193	4529	4338	4071	4398	4032	3389	2877	2385	4278	4048	3117	4223	4445	4094	4112
4527	4150	3987	4182	4551	4425	4349	4565	4229	3994	4299	2562	4076	4247	2028	4260	1532	4311	4013
4066	4321	4332	995	4408	4470	3960	4464	2120	4005	4167	2579	4397	113	4518	4248	4486	12	4270
564	826	4008	387	4132	4342	4219	4363	1212	3471	1063	3368	4084	4282	4030	1661	4541	4263	3978
1396	4459	4251	1677	4070	4085	2288	4288	4164	4441	3620	1053	3989	4420	4463	4057	4376	806	4173
4147	4556	4555	4494	4395	4170	3233	4103	835	4187	4200	4105	4386	458	96	4444	3961	1619	4524
4308	4099	4238	2179	1895	4466	2602	1678	1876	4157	1991	4438	1776	4171	4194	3980	4035	4505	356
4403	4572	4040	475	1366	2671	2827	4516	2909	2964	4315	2884	4540	4548	3263	457	3776	2929	3959
2804	4265	928	4468	4481	1363	604	4083	4201	4351	2065	330	4023	3056	4501	4476	2584	4544	4432
4067	1805	3732	3198	4496	4283	4371	4305	3997	4237	4025	77	4362	4272	4293	3964	4093	1963	527
3097	4442	4264	4477	4506	1727	4473	4343	4295	2373	4043	5190	1763	29	2395	2506	6112	6113	3896
683	949	6114	6115	3814	6116	6117	3073	2161	3751	3649	6118	6119	647	6120	6121	1626	6122	6123
6124	1944	548	1438	2569	6125	6126	6127	3480	6128	1345	1760	2639	6129	6130	6131	6132	6133	6134
6135	6136	6137	6138	1300	6139	6140	3863	6141	6142	6143	6144	6145	6146	6147	5713	6148	6149	3224
6150	6151	2524	6152	2836	4594	6153	2605	6154	1278	1843	6155	3078	6156	3651	799	6157	6158	4933
6159	255	5801	1140	3697	6160	6161	2076	1260	6162	3315	6163	6164	6165	6166	6167	4975	2701	6168
6169	6170	6171	6172	6173	3541	3721	6174	1603	4846	5739	2517	6175	927	6176	3013	6177	6178	2969
6179	6180	6181	6182	2162	6183	6184	6185	1411	5264	6186	6187	4751	6188	1810	6189	2499	6190	6191
6192	4946	6193	6194	6195	3126	2655	6196	6197	6198	6199	3069	1297	853	770	1490	2953	2824	6200
6201	2290	6202	98	3345	6203	3717	6204	6205	6206	6207	1032	622	6208	1773	6209	6210	6211	2790
6212	1860	6213	2575	951	1884	5539	2669	1149	6214	689	6215	3390	2247	6216	1268	6217	6218	6219
6220	6221	1105	5620	6222	3835	6223	2298	6224	6225	6226	1886	6227	6228	6229	503	3572	6230	6231
6232	6233	6234	6235	3877	959	3415	1719	2002	1246	154	1074	2060	6236	5454	6237	6238	384	1170
6239	969	2332	6240	5307	2183	6241	2240	1262	3734	6242	6243	6244	6245	3679	3399	62	52	1398
2398	6246	130	2848	6247	6248	402	6249	2113	6250	1659	6251	2927	6252	6253	1274	467	3788	6254

553	3176	6024	874	3934	6255	6256	329	6257	6258	1599	3662	6259	5950	6260	6261	3417	6262	1407
2902	3530	5929	803	6263	1171	299	6264	6265	6266	6267	6268	726	1631	6269	1346	6270	543	3166
334	4976	2092	3061	982	6271	6272	6273	6274	6275	6276	6277	409	6278	6279	6280	6281	6282	2321
6283	6284	6285	435	4961	3613	2436	765	6286	6287	654	6288	6289	1259	6290	1634	1562	6291	6292
6293	6294	6295	3926	6296	595	2779	2201	6297	6298	6299	6300	6301	1692	640	126	6302	6303	4587
6304	6305	6306	3608	5638	6307	6308	2059	851	6309	6310	6311	6312	2172	1030	1217	6313	6314	6315
6316	708	1213	3635	433	6317	1414	6318	6319	6320	6321	613	500	6322	6323	6324	3486	552	2171
6325	6326	6327	6328	2192	6329	1072	6330	4795	6331	6332	6333	855	6334	6335	1569	1802	1285	6336
6337	3862	6338	1990	783	6339	605	6340	5057	6007	401	6341	6342	4793	6343	6344	868	6345	2029
5802	6346	6347	6348	5368	380	3139	6349	6350	6351	6352	2326	5541	95	3890	6353	6354	6355	2644
6356	5664	6357	6358	6359	6360	4691	3302	6361	6362	1889	6363	6364	6365	2978	6366	1095	6367	1950
6368	445	6369	6370	3526	6371	370	6372	6373	5490	1780	6374	1042	1673	3951	6375	6376	6377	5196
6378	6379	6380	6381	2628	6382	2434	904	6383	6384	134	2865	2147	2181	6385	6386	1055	3659	6387
6388	6389	6390	6391	6392	6393	6394	6395	6396	734	6397	6398	2947	6399	1665	2485	6400	1479	6401
6402	6403	2880	3595	6404	6405	3482	6406	327	955	3876	6407	4942	1434	1036	6408	6409	1483	6410
362	6411	6412	6413	4856	6414	5942	6415	6416	6417	2320	5778	6418	6419	6420	6421	1276	3897	6422
6423	5617	6424	6425	882	104	225	6426	6427	2708	6428	6429	6430	1530	6431	1911	6432	1713	6433
6434	3794	2763	6435	6436	6437	5372	769	2342	6438	6439	6440	2454	6441	6442	6443	3867	6444	423
2051	6445	6446	776	5074	471	6447	6448	3911	6449	6450	3646	1010	5945	6451	6452	6453	483	6454
254	6455	6456	6457	6458	6459	6460	428	6461	3713	6462	6463	6464	6465	6466	2166	6467	6468	4810
1826	6469	6470	6471	6472	4876	1568	5754	5839	6473	6474	3529	6475	5794	3665	3514	6476	3341	3619
259	6477	6478	2760	2216	3475	476	3280	2769	5566	6479	6480	6481	1689	3355	4904	754	6482	439
6483	4755	6484	6485	782	1405	1771	6486	6487	2586	148	2971	6488	6489	6490	123	6491	6492	3650
2229	694	6493	1147	1729	3082	6494	3923	5881	6495	6496	388	6497	1645	6498	6499	6500	6501	6502
6503	6504	2068	2367	6505	2629	6506	2038	919	6507	1020	1627	5161	916	3912	6508	6509	6510	1198
768	702	372	417	6511	5142	6512	3733	2046	6513	6514	2337	2762	6515	4957	6516	3253	2633	6517
6518	1767	6519	6520	3220	3888	6521	1504	6522	438	6523	293	1478	6524	3671	1740	6525	2193	3798
1907	6526	5507	3137	6527	482	4700	5911	6528	6529	6530	1376	3577	3548	6531	4929	6532	3907	1033
2391	3727	6533	6534	6535	3235	2702	6536	1023	5242	3448	6537	1156	6538	1817	6539	6540	6541	6542
6543	6047	979	218	2778	6544	5414	5137	6545	3112	3292	6546	578	6547	6548	958	1468	5371	252
338	6549	6550	1221	6551	6552	6553	3446	6554	2327	6555	5129	6556	6557	2225	6558	6559	792	6560
6561	4972	6562	6563	6564	6565	3438	3782	1620	6566	6567	1459	3158	1163	6568	3264	6569	3447	6570
6571	6572	3757	6573	1461	6574	6575	6576	1403	980	6577	2817	6578	1584	2124	1339	6579	6580	2010
1702	6581	6582	75	6583	6584	756	6585	6586	6587	2583	1630	6588	6589	6590	2324	6591	3342	257
6592	3589	6593	6594	6595	6596	6597	6598	6599	6600	3887	392	406	1158	6601	6602	6603	228	3397
5306	6604	6605	6606	6607	2036	815	1741	6608	2994	6609	5665	5065	2145	3348	6610	3618	275	2936
6611	34	6612	1541	2349	709	6613	4709	2687	6614	6615	6616	542	6617	6618	6619	6620	6621	2306
2693	6622	6623	6624	6625	6626	2106	6627	6628	3807	3841	261	6629	491	5277	2624	6630	1812	1885
5054	2313	6631	6632	3571	6633	6634	107	6635	6636	6637	2422	3223	6638	6639	6640	788	2794	6641
2250	6642	6643	3354	1186	2543	6644	6645	6646	6647	1090	2296	6648	6649	3596	2907	1321	6650	6651
5336	6652	6653	6654	6655	6656	6657	2758	6658	2030	6659	6660	6661	609	6662	5397	1927	178	1480
3607	6663	1983	1981	931	3358	6664	3060	5338	6665	1534	3902	2993	1287	6666	6667	6668	6669	6670
5392	1218	359	6671	3420	3269	6672	304	4993	2160	2812	6673	6674	6675	6676	921	6677	2354	6678
6091	6679	1457	557	6680	6681	6682	2513	3295	6683	6684	6685	1336	3816	6686	6687	2091	6688	2656
6689	6000	4770	6690	463	3767	6691	6692	2491	5201	6693	6694	6695	6696	216	6697	6698	6699	5146
6700	6701	6702	6703	3376	3035	5746	6704	6705	3574	6706	6707	6708	6709	6710	450	5606	6711	6712
6713	6714	6715	6716	789	3869	6717	6718	3556	2620	6719	1976	1286	6720	6721	6722	2015	3554	6723
6724	2520	6725	3842	6726	3191	6727	1270	6728	6729	6730	1909	1312	6731	5051	3532	687	6732	6733
1874	1821	6734	3156	6735	6736	4847	1782	1970	3062	6737	6738	6739	6740	6741	1066	6742	58	1523
6743	315	2035	6744	6745	6746	6747	6748	6749	3561	3714	6750	6751	4648	6752	5265	6753	6754	3575
6755	6756	6757	3587	1088	6758	5348	6759	6760	1189	1995	2090	6761	4771	3778	3489	3691	1560	6762
6763	6764	6765	6766	6767	615	6768	6769	2775	3629	767	6770	6771	6772	6773	6774	5626	1431	1642
1433	1625	6775	3856	395	1106	6776	6777	1841	6778	6779	6780	5546	1119	2676	2614	6781	6782	6783
6784	2535	6785	506	1796	1783	1400	2414	6786	6787	2678	3579	6788	3398	4815	6789	2985	124	2102
6790	6791	2214	3544	2168	1083	6792	6793	2131	3790	3026	6794	2729	6795	6796	1695	6797	2432	6798
19	6799	6800	6801	6802	6803	6804	6805	6806	3218	1387	4866	1139	3885	6807	6808	6809	975	6810
6811	6812	1809	6813	6814	6815	2700	6816	6817	6818	6819	6820	773	6821	6822	5935	6823	2277	1736
1241	6824	6825	6826	163	6827	1087	727	6828	4967	6829	6830	6831	6832	1600	3347	6833	6834	136
4862	6835	965	292	2668	3764	3372	6836	6837	6838	6839	2916	1378	6840	3242	6841	6842	930	4897
1610	6843	1423	2070	5631	6844	6845	3053	5292	6846	479	6847	3287	1142	3634	6848	6849	6850	4723
6851	1193	3320	6852	6853	2652	2236	6854	614	4710	3711	6855	6856	1351	2899	312	6857	6858	6859
987	6860	863	6861	3433	6862	6863	41	6864	6865	6866	5891	290	2184	3744	6867	6868	3765	6869
6870	6871	6872	268	2452	6873	6874	6875	6876	610	3769	6877	6878	2798	6879	571	2000	6880	6881
5151	6882	2680	1244	6883	6884	6885	3801	3565	6886	966	5767	6044	6887	6888	6889	6890	6891	6892
6893	811	6894	4980	555	1980	2683	6895	2244	6896	6897	1311	648	137	820	6898	6899	3383	6900
2751	6901	186	6902	6903	6904	1633	6905	6906	5719	332	6907	2389	3598	791	6908	805	6909	324
1658	2609	3459	6910	6911	6912	6913	6914	6915	1891	231	877	6916	6917	2995	6918	1162	6919	1559
1717	4971	6920	6921	6922	1359	2304	6923	6924	6925	6926	5437	5285	6927	1233	2591	6928	2717	6929
6930	2101	6931	1846	2749	6932	6933	697	2390	6934	2067	6935	1059	6936	54	4671	448	554	1680
6937	6938	3171	3563	6939	6940	6941	3210	6942	6943	6944	6945	918	6946	111	6947	5297	61	6948
6949	2148	481	5861	6950	6951	6952	1735	6953	6954	2267	6955	3622	6956	6957	3864	6958	1551	6959
6960	6961	5335	2682	5769	6962	6963	6964	363	618	2908	6965	6966	6967	3381	6968	6969	6970	1456
2977	6971	3107	6972	191	6973	1408	1934	6974	6975	6976	3755	6977	6978	6979	5530	2231	3768	2710
6980	3545	3657	2261	1663	1636	3549	6981	3918	5331	6982	3353	383	6983	377	6984	6985	6986	2530
3695	6987	6988	6989	6990	3185	744	6991	6992	2154	6993	580	6994	6995	6996	1957	3252	6997	6998
3462	6999	2610	5506	5061	1145	3551	326	7000	7001	7002	7003	7004	5262	7005	7006	7007	7008	7009

4969	721	1440	7010	7011	7012	5906	5607	398	7013	1641	7014	7015	7016	3610	7017	7018	3141	116
6104	7019	1447	1621	1906	549	7020	7021	7022	2072	2715	7023	802	808	7024	7025	7026	5020	7027
7028	7029	7030	7031	489	7032	7033	7034	7035	1179	7036	264	3172	7037	4763	7038	7039	7040	7041
7042	7043	5931	7044	63	7045	3898	7046	7047	3725	7048	1200	7049	1375	266	2255	7050	2280	7051
7052	2842	7053	1533	7054	7055	967	7056	7057	7058	7059	5131	7060	1618	7061	7062	7063	3906	7064
7065	7066	7067	7068	7069	7070	7071	2312	5528	3850	7072	7073	7074	5312	2885	1815	452	7075	7076
1971	2301	7077	7078	7079	7080	7081	519	999	2913	164	925	7082	2351	7083	3506	7084	7085	4824
42	7086	7087	7088	7089	7090	7091	846	7092	7093	7094	7095	7096	7097	7098	7099	7100	7101	7102
7103	2537	7104	7105	7106	7107	7108	7109	2407	7110	7111	3942	2343	7112	1310	7113	7114	7115	7116
3760	7117	7118	7119	7120	7121	2601	7122	2746	7123	3184	7124	7125	7126	1333	7127	6014	7128	7129
7130	2428	7131	7132	7133	516	101	3298	7134	2900	587	7135	1368	3927	7136	7137	7138	541	7139
7140	7141	7142	7143	2783	3728	3905	7144	568	7145	7146	7147	1256	7148	1623	2887	957	5486	5175
7149	4735	3870	2415	7150	7151	1464	3177	7152	7153	7154	3225	7155	1115	7156	5436	7157	3387	7158
7159	1505	7160	2534	7161	7162	7163	1367	3809	7164	7165	7166	1585	7167	7168	7169	7170	3168	7171
7172	7173	2336	7174	3357	7175	5989	7176	5279	3487	7177	7178	7179	7180	7181	7182	7183	3159	7184
7185	7186	7187	7188	7189	2430	2069	3520	7190	2119	674	7191	353	2722	7192	2725	5439	2803	2673
7193	7194	7195	7196	7197	1685	1146	7198	7199	871	1867	2443	1904	3098	3391	7200	7201	4622	5690
7202	7203	7204	2478	7205	241	1687	7206	3186	302	250	629	2719	7207	836	7208	258	752	7209
7210	7211	7212	333	7213	2892	5673	7214	7215	7216	7217	1107	7218	2353	7219	657	2205	7220	7221
7222	7223	5532	3423	7224	7225	2204	7226	7227	1502	1124	7228	876	7229	5258	1057	5514	3815	7230
7231	7232	1598	7233	7234	2957	7235	391	3453	4970	7236	7237	7238	3715	7239	7240	7241	1121	3511
7242	7243	2053	2851	7244	5158	7245	1316	7246	3503	1299	1542	1744	7247	3687	1851	7248	7249	1529
5301	7250	2411	7251	421	7252	592	2739	7253	7254	1836	7255	3705	340	1953	7256	7257	6001	3311
7258	678	337	7259	3761	7260	3821	7261	2208	455	2819	143	2813	544	819	7262	7263	3217	7264
7265	4907	3795	3308	1945	1177	3615	5422	7266	1928	2630	3908	7267	1617	7268	105	7269	514	3442
7270	7271	3825	7272	339	7273	7274	7275	7276	7277	7278	3384	7279	7280	7281	7282	3763	7283	7284
3142	3868	7285	2861	7286	5759	7287	7288	7289	7290	3379	1288	901	7291	7292	7293	7294	3229	3444
1485	5684	2675	2393	7295	7296	3392	57	240	3047	7297	3070	7298	7299	7300	7301	7302	7303	1344
7304	3750	3312	3645	1828	3451	7305	7306	193	3872	7307	7308	5865	3621	5738	1473	7309	7310	2598
2801	219	7311	2079	7312	7313	7314	900	3781	2440	1265	7315	923	7316	7317	1334	7318	7319	5763
7320	1890	7321	7322	7323	7324	7325	7326	2565	7327	7328	7329	5239	7330	276	7331	7332	7333	7334
7335	7336	48	7337	321	5884	7338	3919	7339	3197	7340	7341	880	5411	7342	7343	7344	7345	7346
7347	7348	7349	1887	7350	1264	3792	3165	7351	7352	7353	7354	7355	1613	2868	7356	7357	4	5889
2835	7358	7359	7360	3356	7361	2557	7362	2545	4802	7363	1864	1777	7364	5781	7365	7366	7367	1309
7368	7369	2970	7370	7371	7372	7373	7374	933	1761	1679	509	1350	3840	2756	7375	2904	7376	3332
2264	3829	7377	7378	7379	7380	1104	7381	3712	7382	7383	3939	7384	7385	5089	5080	7386	5447	7387
7388	7389	7390	7391	5358	4669	7392	5116	4597	7393	170	2808	3317	6006	7394	7395	7396	7397	2402
5885	7398	7399	7400	7401	7402	1694	7403	7404	7405	51	5849	7406	2522	7407	7408	7409	7410	7411
7412	7413	7414	7415	7416	7417	7418	7419	7420	7421	7422	7423	7424	102	2094	7425	1918	7426	7427
4901	7428	1583	7429	7430	7431	460	7432	7433	1644	7434	7435	3800	7436	4634	7	7437	256	7438
7439	7440	7441	7442	7443	7444	7445	1964	7446	627	1572	7447	151	3407	1929	7448	7449	7450	2616
7451	2359	7452	2066	227	7453	7454	6029	2335	7455	7456	2515	7457	732	7458	7459	7460	7461	2662
2784	2611	1770	319	7462	7463	432	7464	5727	181	897	2564	7465	2492	1420	7466	7467	3009	7468
7469	7470	7471	7472	1580	74	7473	7474	7475	7476	3346	7477	7478	7479	7480	2146	7481	7482	492
2178	7483	630	7484	7485	7486	7487	1084	7488	747	735	7489	5893	839	7490	7491	7492	7493	7494
1338	7495	7496	1109	7497	3903	7498	7499	895	7500	5038	5878	2097	7501	7502	4873	2792	7503	7504
285	7505	2666	7506	2523	5666	2417	3049	7507	7508	7509	2914	1996	3557	7510	243	7511	7512	7513
7514	617	7515	5217	2647	1835	922	7516	7517	7518	2484	5561	7519	5714	3268	2470	7520	7521	7522
7523	1772	5999	2318	43	1248	7524	7525	1811	103	7526	403	3170	1448	7527	1888	7528	4950	7529
7530	7531	7532	3861	818	169	7533	1117	7534	4656	5896	3203	7535	7536	7537	7538	7539	7540	7541
7542	1752	1258	2268	7543	2911	2180	3559	7544	7545	7546	3282	7547	7548	1472	7549	7550	1018	7551
7552	7553	7554	5405	7555	7556	1850	2234	3425	7557	446	903	7558	7559	4667	7560	7561	7562	7563
3910	7564	7565	7566	7567	7568	405	5675	3754	7569	1395	7570	7571	7572	7573	1496	1325	7574	7575
1390	2322	2558	474	3473	1508	2284	7576	418	5519	7577	7578	4920	7579	7580	1974	7581	2008	1178
7582	3371	695	7583	7584	2555	1935	7585	7586	2896	3153	7587	3045	7588	176	780	1682	7589	7590
3393	7591	968	2305	7592	7593	4699	7594	7595	7596	7597	443	7598	2037	7599	1537	7600	7601	3922
1243	810	3947	3785	1444	7602	7603	1883	7604	1060	7605	1546	7606	1822	2679	7607	7608	603	7609
7610	7611	1441	1952	7612	7613	7614	2375	7615	7616	711	7617	7618	3437	2521	6045	1192	3823	7619
7620	7621	2636	4919	7622	1753	7623	7624	1003	7625	1842	7626	7627	3443	7628	7629	5066	5049	2382
2670	2219	7630	2894	1295	5155	1845	7631	5686	7632	938	7633	7634	5833	910	2626	2723	7635	5119
7636	1651	7637	3029	2494	7638	659	4694	3051	7639	1555	5848	7640	7641	16	6037	7642	7643	5992
3192	7644	7645	1914	7646	3278	7647	3265	127	7648	2303	7649	495	1091	7650	7651	2667	7652	2834
66	2011	1847	7653	1699	1413	5599	7654	6079	15	7655	3683	7656	1065	1660	7657	5680	7658	7659
3550	7660	7661	357	1224	833	368	6111	2232	7662	3326	7663	3193	7664	7665	86	7666	7667	436
7668	739	7669	7670	7671	4623	7672	7673	7674	7675	7676	7677	7678	7679	7680	7681	7682	7683	1261
7684	7685	7686	7687	2088	7688	7689	759	7690	7691	3535	7692	1501	3586	7693	7694	7695	7696	195
2635	2104	7697	3246	7698	7699	7700	7701	7702	3174	3892	7703	7704	7705	7706	2420	7707	718	7708
3450	3305	7709	2245	2547	7710	7711	7712	7713	7714	3427	122	7715	1567	7716	7717	7718	7719	656
7720	7721	7722	7723	7724	3028	3593	5130	5438	2228	7725	7726	7727	7728	2488	7729	7730	5902	7731
174	1916	2566	7732	68	7733	7734	1415	588	3818	336	7735	7736	7737	915	2742	7738	526	7739
7740	7741	5549	1769	3015	7742	7743	1784	7744	7745	2174	1535	7746	1710	7747	7748	7749	1714	7750
7751	7752	7753	7754	7755	1954	7756	7757	2176	7758	7759	2445	4960	7760	7761	7762	7763	155	2416
7764	3163	7765	3340	879	7766	1758	3010	7767	4892	2279	7768	5648	2711	7769	2339	7770	472	5156
7771	7772	7773	3771	7774	3119	7775	7776	3806	3793	512	7777	7778	196	2246	722	939	2223	3693
60	7779	7780	7781	7782	297	2552	1966	7783	7784	7785	7786	6023	1332	7787	7788	2860	533	7789
5025	7790	1374	7791	7792	7793	7794	3924	7795	2531	7796	7797	416	7798	2039	3008	7799	7800	2514

318	1674	7801	7802	7803	3681	7804	3237	7805	7806	3685	7807	7808	582	3641	1693	3118	7809	2728
4806	3180	440	7810	2108	7811	251	2152	1099	2197	7812	5554	1607	2839	2309	1061	3188	7813	2252
536	7814	7815	7816	7817	7818	7819	1050	7820	7821	7822	5844	7823	970	7824	2968	2540	7825	670
5452	598	2128	3132	7826	3494	2338	1789	7827	2871	3774	2926	4890	7828	7829	7830	7831	3240	7832
7833	3833	7834	7835	7836	468	7837	2297	7838	7839	7840	7841	5386	7842	7843	1141	7844	7845	4828
7846	2664	7847	7848	234	501	1135	7849	873	7850	7851	236	7852	2944	4632	7853	5304	7854	7855
53	7856	7857	7858	3625	7859	7860	7861	7862	1216	7863	7864	7865	5703	2681	7866	5895	3824	7867
7868	2310	7869	7870	1000	7871	244	976	1818	7872	1054	2613	4850	5244	7873	5965	7874	7875	1499
2797	35	7876	7877	7878	7879	3952	7880	775	1737	7881	7882	7883	7884	7885	7886	7887	7888	2175
3493	5715	72	7889	3147	3501	7890	7891	2563	7892	6033	7893	5466	7894	3569	85	7895	7896	1524
2867	7897	3858	7898	7899	964	7900	7901	211	7902	7903	3914	830	7904	1690	7905	2833	171	7906
7907	2435	355	867	7908	7909	2213	7910	7911	2859	7912	746	7913	5650	1622	2014	7914	7915	1424
7916	972	7917	100	7918	961	7919	1684	7920	7921	1592	7922	7923	2526	7924	5934	1308	736	4912
7925	1236	7926	2093	3483	7927	3434	7928	7929	7930	2206	7931	7932	2286	7933	5807	7934	2809	7935
7936	7937	7938	7939	7940	7941	7942	3093	7943	7944	224	859	2805	3034	7945	7946	7947	7948	7949
7950	2766	2274	2142	7951	5379	7952	7953	2187	3136	7954	7955	7956	2265	3787	7957	7958	7959	5291
1487	1715	7960	2930	1839	3644	2991	6110	2706	2898	7961	5576	2480	2370	4886	804	7962	793	7963
7964	7965	7966	5967	7967	7968	7969	7970	7971	7972	7973	3597	3066	7974	7975	7976	829	532	7977
7978	5092	5856	7979	7980	488	7981	7982	4677	7983	7984	92	7985	7986	7987	76	7988	7989	1932
7990	3802	3214	7991	7992	2733	5789	5704	7993	7994	7995	4952	5198	7996	7997	7998	5477	1992	1961
5193	3852	7999	8000	1098	1208	487	1404	8001	8002	8003	2928	2155	8004	8005	8006	8007	8008	146
3044	857	8009	8010	8011	3429	1029	2536	8012	2738	8013	3329	8014	8015	8016	8017	2649	1281	8018
589	2821	8019	2918	8020	2027	352	8021	8022	8023	8024	8025	8026	8027	2883	2638	8028	8029	5459
5668	8030	8031	3306	2278	521	8032	8033	8034	2590	8035	3125	2872	1819	8036	8037	71	8038	1993
8039	5234	1394	8040	8041	3022	451	8042	2043	1044	8043	5171	8044	2933	8045	8046	1294	3196	2529
8047	8048	1427	8049	5019	1409	8050	3000	8051	8052	4661	8053	213	3591	3409	8054	8055	2776	8056
8057	8058	8059	8060	8061	8062	2889	8063	1352	652	8064	2444	8065	8066	8067	8068	8069	2737	8070
8071	2519	2086	1234	1711	8072	8073	217	222	8074	1503	2573	1700	8075	8076	5975	8077	8078	8079
1019	535	8080	3189	2281	8081	168	484	8082	1893	8083	8084	2400	8085	3933	8086	3599	8087	8088
273	1754	8089	8090	8091	1275	8092	8093	8094	8095	3324	8096	5204	655	8097	8098	3375	2087	8099
3365	8100	8101	108	5523	847	5643	8102	8103	2042	37	3261	5570	5322	5871	2599	8104	8105	3773
8106	8107	8108	1196	8109	4659	8110	8111	1525	3811	4780	8112	8113	8114	639	1985	6020	8115	8116
5461	8117	8118	801	2410	737	8119	8120	8121	8122	5814	1211	3426	2271	8123	3507	1676	8124	2856
3319	2796	8125	2136	8126	3614	2845	984	8127	3412	8128	2539	8129	864	1586	8130	8131	8132	989
8133	1482	8134	8135	8136	8137	8138	23	6080	8139	5699	8140	1206	8141	187	1650	2581	2843	8142
1126	1869	8143	4871	8144	8145	2941	534	1451	8146	8147	8148	6096	328	924	294	6072	2691	8149
8150	8151	8152	1220	8153	3517	2369	8154	8155	156	1973	2140	1765	1458	8156	8157	8158	8159	8160
5691	2024	8161	8162	2437	8163	8164	8165	1797	8166	8167	106	8168	8169	242	8170	3081	1749	3144
8171	93	99	8172	8173	8174	8175	8176	2917	8177	8178	8179	1731	8180	8181	8182	8183	8184	8185
2429	1011	719	5972	3929	8186	8187	8188	8189	8190	5542	2173	8191	8192	8193	1528	3065	8194	8195
2084	8196	8197	8198	8199	2589	1958	1168	8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210
8211	8212	962	3343	5939	898	464	8213	8214	8215	8216	572	8217	8218	8219	1160	3227	8220	3626
3328	3624	1724	8221	3258	3882	8222	1047	3600	973	8223	8224	8225	8226	3576	884	5179	5928	8227
2757	5850	8228	2476	8229	8230	8231	8232	8233	8234	8235	8236	3291	947	3528	8237	8238	8239	8240
8241	8242	8243	8244	8245	2387	3143	8246	8247	8248	8249	8250	8251	8252	8253	743	1941	8254	3283
8255	1498	646	306	119	1965	8256	425	8257	8258	8259	8260	8261	323	3083	8262	343	682	3290
3827	520	8263	2663	5248	1005	8264	2158	8265	8266	8267	3206	8268	1062	8269	1154	3505	2316	8270
2012	8271	8272	8273	1742	8274	1199	5132	1132	852	1997	2781	6087	8275	2754	3834	2489	8276	8277
2551	8278	971	5890	1343	8279	2487	8280	8281	8282	313	3592	8283	8284	2392	5757	1657	3567	2460
8285	8286	8287	8288	8289	5225	8290	8291	8292	8293	8294	8295	8296	8297	1791	575	2033	8298	8299
2950	8300	8301	5827	8302	3804	2418	745	8303	8304	2283	8305	5988	1202	669	1830	8306	8307	8308
730	8309	8310	3789	8311	2077	8312	8313	1590	2961	8314	8315	2049	8316	1734	3247	8317	1075	8318
3881	8319	8320	8321	8322	8323	8324	3160	2256	8325	3274	1589	8326	5362	8327	6097	8328	8329	8330
733	3033	8331	1190	8332	1435	8333	8334	8335	2111	8336	279	2259	4690	8337	8338	8339	8340	1421
875	1937	5583	1575	1774	8341	8342	8343	1014	960	8344	8345	239	8346	8347	1289	5426	8348	5694
8349	8350	8351	461	8352	1452	1743	8353	4945	8354	8355	3547	8356	8357	2873	3337	4665	2230	8358
2785	8359	3813	3708	8360	3286	2149	8361	8362	412	8363	8364	8365	1076	8366	8367	8368	8369	8370
3310	1894	8371	2587	8372	8373	8374	8375	4898	8376	8377	8378	1897	8379	3916	8380	8381	714	8382
8383	8384	8385	8386	3150	8387	2237	1608	1686	2912	8388	3313	1905	8389	2130	1043	8390	8391	8392
473	8393	3584	2062	3617	3400	8394	8395	2532	5559	8396	8397	8398	5627	434	8399	8400	2300	8401
8402	1284	8403	796	1086	8404	8405	1643	8406	2799	2956	2482	1725	5609	8407	8408	2528	2585	8409
1998	4604	8410	5118	2665	8411	8412	8413	396	1377	8414	8415	8416	1861	2832	5936	662	2446	8417
8418	8419	8420	2658	8421	8422	8423	31	8424	8425	8426	8427	5482	8428	8429	8430	1871	8431	4577
165	8432	1947	531	8433	2973	8434	8435	8436	3745	121	3855	8437	8438	8439	2838	73	2905	8440
8441	8442	8443	5947	2031	8444	8445	8446	1125	2058	8447	3080	8448	4932	8449	8450	892	8451	496
3512	8452	8453	8454	5229	8455	5088	712	1172	2464	8456	8457	8458	528	8459	8460	4868	8461	8462
8463	704	8464	5133	3074	2307	8465	8466	3228	2346	2207	2852	1733	8467	8468	2893	5870	8469	4576
8470	2736	8471	8472	3915	2503	280	2438	413	978	8473	133	2254	8474	201	8475	671	8476	2341
5785	3349	8477	8478	8479	5639	8480	690	8481	1984	8482	8483	8484	2650	298	8485	1238	5139	2212
1917	8486	3484	3930	2582	2471	2910	8487	5203	8488	2891	87	8489	8490	2915	8491	1326	3419	1418
5110	8492	8493	8494	1994	115	1442	5973	8495	5851	404	8496	8497	8498	8499	8500	8501	8502	3851
8503	8504	8505	1210	8506	8507	1554	2098	1108	8508	1080	1219	1545	3949	4934	5353	8509	4936	8510
8511	8512	8513	3481	3762	3515	3297	8514	741	3921	8515	3573	2479	842	8516	480	1486	8517	1203
8518	8519	8520	1460	4664	2548	8521	3678	2934	5164	560	4812	2741	2608	8522	1113	8523	8524	3402
8525	2032	2870	8526	8527	8528	2496	3164	8529	8530	8531	271	8532	8533	8534	8535	8536	4915	8537
466	6003	8538	8539	8540	8541	5917	8542	1131	8543	8544	8545	8546	1207	172	1956	8547	2458	8548

2384	5946	8549	8550	3042	8551	1188	1827	8552	8553	4958	8554	8555	872	8556	189	8557	8558	8559
8560	1982	1801	757	1491	3036	8561	2726	2924	3499	8562	2986	3179	8563	2006	1833	8564	8565	3470
8566	8567	354	8568	8569	1282	8570	5905	278	8571	777	8572	8573	8574	5527	8575	2163	8576	834
8577	8578	8579	3668	3605	8580	1318	8581	1111	5296	8582	8583	8584	8585	8586	5077	4776	8587	8588
8589	3822	8590	3849	8591	1908	948	8592	454	2533	2724	8593	8594	8595	8596	8597	8598	8599	1492
3146	3848	8600	8601	8602	8603	8604	237	2380	3128	8605	3239	8606	8607	8608	893	8609	424	2345
202	8610	8611	8612	8613	3661	2377	8614	8615	3562	8616	8617	8618	8619	2467	2577	2073	2085	3071
8620	8621	3046	8622	8623	3232	8624	8625	1205	4756	3627	623	8626	8627	152	8628	2525	8629	415
8630	8631	8632	8633	8634	5094	8635	8636	2475	8637	5816	1136	1558	1254	2425	3943	2593	2421	1832
8638	1364	8639	8640	3460	787	5481	8641	2607	8642	1656	2811	8643	5573	8644	8645	1756	8646	83
684	8647	8648	2426	8649	8650	3948	766	3025	3726	3299	8651	8652	1615	8653	8654	8655	1252	8656
8657	5124	8658	8659	1838	3604	790	3135	574	8660	8661	8662	8663	8664	2082	2510	8665	2553	1570
3904	1280	1360	1230	8666	5180	8667	8668	8669	2095	8670	8671	8672	8673	8674	8675	8676	667	1412
8677	8678	8679	8680	287	3068	8681	8682	8683	1691	977	8684	8685	8686	8687	8688	8689	8690	637
8691	1008	8692	8693	8694	8695	626	8696	5349	1969	167	8697	8698	1578	5083	2466	8699	8700	8701
8702	8703	1794	8704	160	8705	706	8706	8707	912	596	8708	8709	8710	8711	3633	1305	8712	8713
8714	8715	707	3694	2328	8716	8717	8718	8719	8720	5126	5795	8721	786	8722	8723	1073	2646	5921
8724	3463	827	8725	3770	2169	8726	8727	8728	1425	2333	8729	8730	681	8731	4680	8732	8733	889
8734	3935	5693	8735	4908	8736	8737	8738	8739	8740	18	2946	2972	4748	3455	8741	8742	8743	1494
8744	717	1296	8745	2498	8746	8747	2427	2508	8748	1923	8749	8750	8751	3779	6	8752	1373	400
8753	8754	567	3092	8755	5084	8756	8757	1540	8758	2123	8759	1924	8760	8761	8762	8763	8764	3716
8765	8766	3213	8767	5790	2243	2641	1757	8768	8769	3116	8770	8771	8772	2409	8773	3064	8774	8775
8776	2177	8777	2594	5112	1348	3351	8778	8779	8780	8781	2269	8782	8783	8784	3857	8785	8786	8787
88	3063	1166	8788	2505	316	515	8789	3322	2935	2362	2507	8790	8791	8792	5518	8793	3373	8794
8795	1716	8796	2364	2258	5261	616	2276	758	8797	8798	779	8799	8800	1239	4600	1148	3090	8801
8802	8803	2020	8804	8805	1813	1022	8806	1837	109	3594	8807	3524	2568	2034	8808	8809	3476	795
8810	8811	8812	8813	2117	1999	4736	8814	3327	3643	2203	3211	8815	8816	8817	8818	4792	8819	1859
3382	1173	282	8820	8821	8822	8823	8824	8825	2251	641	3422	8826	305	624	8827	4808	8828	8829
8830	8831	8832	8833	1279	1748	8834	8835	540	5629	3653	1635	8836	8837	4857	8838	8839	8840	8841
8842	2705	8843	4874	8844	3335	246	8845	843	2615	8846	3743	8847	8848	1401	3488	8849	8850	2822
8851	8852	2153	379	5804	8853	5505	8854	8855	5760	1899	8856	8857	8858	8859	8860	147	8861	8862
407	498	2459	3017	8863	1823	1509	8864	8865	2139	3578	8866	8867	8868	8869	8870	1045	3739	8871
5174	8872	5281	2025	8873	8874	2592	1755	8875	8876	8877	1548	4608	8878	8879	8880	8881	8882	8883
8884	8885	2780	2864	3331	1214	8886	8887	8888	422	8889	8890	8891	1467	3803	8892	8893	2413	8894
8895	1703	8896	262	8897	1764	2857	2403	4930	1930	3113	4877	8898	3502	8899	4718	8900	1500	2052
8901	8902	2875	8903	3333	8904	837	8905	2826	1169	1788	8906	556	8907	8908	569	3525	2064	1611
705	33	8909	3945	724	5105	990	8910	8911	2442	763	8912	8913	8914	8915	8916	8917	8918	3318
8919	8920	8921	5867	3458	8922	2560	8923	8924	411	8925	2238	8926	8927	8928	6046	3321	8929	5031
983	5948	8930	2211	1704	8931	2054	5010	2285	1185	8932	3386	8933	8934	748	8935	8936	8937	8938
8939	8940	8941	8942	8943	8944	8945	5932	1273	8946	1903	274	2105	182	8947	8948	8949	601	444
2129	8950	8951	8952	8953	8954	22	8955	8956	8957	8958	8959	8960	1493	8961	8962	8963	139	3284
8964	8965	2399	3925	8966	685	2366	2814	8967	3111	8968	8969	8970	2118	1565	3271	1134	358	8971
8972	3703	5079	8973	8974	8975	3257	8976	3766	365	1561	2472	1647	956	8977	8978	5502	8979	8980
2982	1027	8981	558	179	8982	8983	8984	8985	3005	8986	8987	8988	8989	8990	2137	8991	8992	8993
8994	8995	8996	6084	6088	8997	2546	8998	5330	8999	9000	9001	3123	9002	9003	9004	9005	1187	1017
2450	4973	9006	9007	9008	9009	9010	848	9011	9012	2789	9013	9014	9015	9016	2554	1606	9017	1873
9018	3485	3102	3704	9019	117	9020	9021	9022	9023	263	1870	5383	2718	4766	9024	740	2465	2959
3468	9025	9026	9027	9028	9029	1516	1948	3212	9030	9031	3120	3509	9032	3187	9033	9034	945	2138
1872	4753	3012	9035	1595	9036	742	3293	1892	394	9037	562	2394	3941	9038	523	9039	2439	9040
9041	9042	9043	9044	2966	9045	9046	2275	5021	2196	78	8	3519	9047	9048	9049	9050	9051	9052
9053	6057	9054	1915	9055	6094	427	9056	2323	9057	9058	9059	9060	9061	1229	9062	5919	9063	5897
2888	9064	4853	497	9065	9066	9067	5412	9068	9069	4804	9070	2457	9071	2358	3883	4633	9072	5560
3300	3084	9073	9074	3692	9075	1707	9076	1834	9077	9078	9079	9080	9081	9082	3466	1476	9083	635
9084	9085	5995	9086	3428	9087	9088	2423	2048	9089	3699	9090	9091	2016	1201	625	410	3077	9092
118	9093	9094	9095	3756	9096	1342	385	5613	9097	840	9098	9099	9100	9101	9102	1648	9103	2215
9104	9105	9106	9107	993	9108	2788	1381	9109	4834	9110	9111	3094	4742	9112	1591	3706	9113	1825
9114	9115	198	887	1290	9116	9117	4612	507	9118	9119	344	3439	9120	9121	9122	9123	3152	9124
5729	24	9125	1553	9126	9127	2408	2431	9128	9129	3200	9130	2374	9131	5047	9132	3085	1371	9133
9134	3564	4831	2625	700	1386	1616	5310	5645	5488	9135	9136	9137	9138	9139	9140	4964	94	9141
3698	9142	9143	9144	9145	2759	1852	9146	9147	9148	9149	9150	5263	9151	9152	3032	9153	3215	9154
9155	5185	9156	281	9157	2882	510	465	9158	9159	9160	9161	2992	3133	30	9162	9163	9164	9165
9166	2096	9167	9168	1103	5070	2858	9169	672	1138	3553	9170	1026	9171	9172	9173	935	9174	453
3370	9175	954	2455	5024	9176	9177	5388	9178	9179	9180	3109	2164	2159	9181	573	9182	9183	1137
5579	1365	650	551	9184	612	1808	110	9185	9186	9187	1455	3104	9188	3537	1949	390	21	3523
9189	9190	9191	9192	9193	9194	9195	3723	9196	4924	9197	9198	9199	4910	9200	9201	5245	2932	9202
3359	9203	9204	9205	9206	9207	9208	9209	9210	9211	9212	9213	9214	1471	9215	6106	9216	5195	1596
389	9217	46	3281	69	9218	2239	2631	3893	1232	269	5952	9219	932	2657	5976	9220	9221	9222
9223	9224	6051	9225	9226	9227	9228	9229	3741	67	1978	9230	658	3900	9231	5187	3570	9232	1875
5641	1223	638	9233	9234	1579	9235	1116	9236	56	9237	1975	9238	9239	192	5582	9240	844	9241
5143	9242	3585	1477	1708	755	9243	3361	1654	9244	9245	9246	9247	1143	9248	9249	9250	44	9251
3048	9252	5157	5040	3928	376	4747	9253	9254	666	3052	1696	2490	1594	4713	9255	9256	1967	346
9257	3256	9258	1602	9259	9260	3334	761	426	9261	9262	1526	1068	9263	9264	9265	9266	9267	5775
9268	2732	1181	9269	2110	9270	3791	3275	1800	9271	9272	3309	4914	3696	9273	9274	9275	3527	9276
2056	9277	9278	9279	9280	9281	9282	1920	9283	9284	565	9285	3895	5260	5253	1655	1762	9286	9287
998	9288	9289	9290	9291	9292	9293	1399	9294	3182	9295	208	5440	9296	2612	9297	9298	9299	4959
9300	9301	9302	2979	3810	9303	5476	9304	1110	9305	9306	3050	1989	2527	1855	9307	3467	9308	9309

9310	9311	3590	3401	663	9312	9313	1671	9314	9315	9316	9317	9318	2765	9319	9320	9321	9322	4858	
9323	131	9324	9325	9326	9327	5808	1453	1078	9328	397	9329	9330	17	9331	645	3304	9332	9333	
307	348	9334	9335	9336	9337	9338	9339	9340	854	9341	4732	2132	9342	9343	3666	9344	9345	9346	
9347	9348	9349	9350	9351	2199	1946	9352	9353	9354	3408	1184	3686	5681	9355	1669	691	9356	5046	
260	600	9357	9358	9359	20	3418	9360	3377	3690	9361	9362	9363	9364	9365	9366	9367	2474	9368	
9369	5978	914	9370	9371	1383	1324	1114	9372	9373	9374	2740	2308	9375	2571	9376	9377	9378	2621	
9379	2570	9380	1191	2456	9381	9382	3364	9383	9384	2996	807	9385	9386	2451	9387	36	9388	5460	
9389	5413	9390	1706	9391	9392	9393	9394	9395	9396	1416	367	2714	9397	9398	9399	9400	462	1582	
9401	9402	9403	1267	3889	9404	9405	381	9406	9407	3680	9408	5172	2685	5910	4809	310	9409	247	
9410	9411	1688	9412	9413	9414	696	9415	9416	9417	9418	3731	9419	9420	3797	3007	9421	9422	9423	
9424	9425	2368	9426	9427	4652	9428	9429	9430	9431	6027	9432	4784	2807	3173	2253	2493	1795	9433	
183	4882	1824	9434	1157	3749	9435	9436	1102	9437	9438	1272	9439	9440	9441	9442	2381	1814	9443	
9444	9445	2787	9446	9447	1913	2388	9448	9449	9450	2806	3416	1865	9451	204	3631	3378	2640	1786	
9452	9453	5858	3411	9454	9455	9456	205	9457	9458	9459	9460	5741	2712	9461	9462	2651	1263	9463	
9464	1474	1085	9465	2061	579	9466	9467	9468	300	1882	1319	9469	9470	9471	545	9472	2188	9	
3366	9473	1506	9474	1681	749	4738	9475	9476	9477	9478	4575	9479	986	1380	2419	9480	1040	3516	
9481	9482	5651	2654	9483	3886	9484	9485	5162	1778	2695	9486	9487	9488	9489	221	3445	9490	9491	
3106	9492	2795	5499	9493	1799	1900	9494	9495	3558	2019	9496	2468	1096	3413	9497	3684	2556	5058	
9498	9499	9500	9501	1538	9502	1153	9503	3839	9504	9505	9506	9507	9508	9509	490	2976	3674	9510	
9511	9512	2	3477	1079	3096	1182	9513	9514	693	9515	2567	3931	9516	9517	9518	2473	9519	5342	
9520	9521	9522	6031	1807	2720	5170	9523	9524	1245	812	9525	9526	1226	9527	9528	9529	9530	3808	
9531	858	1933	9532	1668	1746	5373	9533	3492	4913	9534	2386	2659	1240	4844	2360	3175	9535	9536	
5843	3374	9537	698	5654	9538	1361	2516	9539	9540	1322	5797	2561	9541	9542	4615	3041	9543	1775	
9544	4772	9545	701	9546	9547	2958	9548	1922	9549	9550	9551	9552	3273	9553	2962	4769	9554	9555	
1768	9556	9557	9558	4926	9559	9560	9561	9562	9563	9564	1235	9565	9566	9567	9568	9569	9570	47	
2919	5985	2329	2242	9571	9572	9573	2869	9574	4654	1397	5657	5750	5873	9575	9576	9577	3016	9578	
3248	3339	9579	3474	1879	5005	1804	2841	5780	6022	9580	9581	5800	9582	9583	2939	9584	91	9585	
9586	9587	9588	9589	9590	9591	3030	9592	2777	9593	9594	2829	2818	9595	5841	9596	5056	9597	2363	
4739	9598	9599	9600	9601	9602	3675	6078	9603	9604	636	3296	9605	3038	9606	9607	309	9608	9609	
9610	1446	9611	2937	2241	9612	9613	9614	3449	9615	9616	9617	3338	3538	9618	9619	9620	2903	9621	
1683	9622	9623	9624	3057	9625	9626	9627	1495	2249	1925	9628	9629	634	5637	929	9630	9631	39	
753	9632	3639	9633	9634	9635	128	4887	3088	504	9636	3395	1292	5736	2081	1052	3740	9637	1552	
9638	9639	2075	9640	9641	1968	2686	9642	9643	9644	9645	3602	9646	9647	1806	9648	9649	9650	9651	
1581	3652	2837	9652	4635	9653	9654	9655	9656	845	3362	173	3531	9657	2502	2099	1006	9658	1128	
606	9659	5288	9660	9661	4730	9662	4888	4705	1302	4609	50	2604	135	9663	9664	9665	1209	9666	
3018	9667	909	3472	1164	9668	9669	9670	9671	1556	229	9672	6105	9673	2643	9674	3072	2846	4893	
2044	9675	9676	4724	1428	9677	9678	9679	9680	1520	9681	9682	9683	1097	9684	9685	9686	4689	9687	
1550	9688	3700	9689	1544	9690	9691	3536	6090	9692	676	2114	9693	1484	5016	9694	1250	1962	9695	
817	9696	771	2949	3054	9697	3784	9698	9699	9700	9701	153	9702	2260	5428	3216	3495	9703	2854	
9704	9705	1943	2330	9706	9707	5835	1167	9708	1902	9709	9710	9711	2449	9712	9713	3937	320	9714	
366	9715	1354	3710	9716	9717	2226	3688	660	9718	9719	1829	4801	9720	9721	9722	907	2299	9723	
3701	9724	4981	9725	9726	9727	2849	9728	5828	3294	9729	9730	3277	9731	1328	3920	3075	2121	1831	
9732	9733	9734	1081	9735	9736	3424	738	586	9737	9738	2233	9739	9740	9741	1955	5632	9742	9743	
420	9744	3669	9745	9746	5075	9747	9748	3388	141	9749	9750	9751	9752	9753	4688	4829	3259	2221	
1013	3720	9754	4843	1449	9755	1222	9756	9757	9758	3435	3285	2998	3946	611	9759	1588	9760	9761	
9762	9763	9764	9765	9766	9767	3853	9768	632	9769	9770	9771	5252	502	9772	3155	9773	9774	824	
429	2684	1942	9775	2483	9776	5173	9777	375	9778	9779	9780	9781	9782	9783	2750	199	1511	3148	
9784	530	9785	9786	9787	9788	9789	9790	9791	9792	3208	2688	158	9793	1419	9794	9795	2018	9796	
2406	906	9797	9798	5622	2704	9799	9800	9801	9802	9803	1155	1637	3267	2185	9804	870	9805	9806	
9807	9808	9809	9810	9811	9812	9813	751	1880	9814	9815	1283	9816	2617	9817	9818	9819	9820	1025	
9821	9822	9823	49	3385	9824	9825	9826	9827	1180	9828	9829	2816	3352	9830	9831	2401	9832	125	
1358	508	9833	9834	9835	112	9836	9837	9838	9839	9840	642	9841	9842	9843	9844	9845	816	3642	
9846	9847	9848	9849	813	5689	9850	9851	9852	9853	5188	9854	2850	9855	3718	2572	9856	1564	9857	
9858	9859	2433	9860	9861	80	6082	9862	9863	9864	3270	9865	1304	81	1384	303	2954	940	9866	
214	5012	4935	9867	1653	1130	9868	9869	9870	9871	3255	9872	9873	9874	9875	9876	9877	317	9878	
9879	9880	9881	2495	3606	866	9882	785	9883	856	3201	2463	5427	9884	9885	3940	9886	9887	1432	
797	220	9888	9889	9890	38	442	3672	3880	3630	9891	9892	2112	9893	5068	9894	9895	9896	2200	
2689	9897	3560	2355	9898	2951	9899	3542	9900	9901	9902	5500	9903	3037	1445	2770	1488	3031	6038	
1926	9904	414	2981	3161	9905	5535	1194	3899	860	5009	1392	226	713	9906	2735	1640	3664	5874	
5352	9907	469	9908	9909	9910	9911	3817	9912	5106	9913	1785	9914	9915	9916	943	9917	9918	9919	
2404	1475	2748	9920	9921	9922	9923	9924	4820	9925	2344	9926	9927	9928	2634	3518	9929	3421	9930	
9931	249	1566	9932	9933	9934	3632	9935	9936	9937	9938	9939	9940	9941	9942	9943	1093	9944	3209	
9945	850	9946	9947	2963	9948	1987	511	1112	2967	9949	593	1574	9950	2047	9951	9952	1266	9953	
9954	9955	9956	459	1426	9957	3039	3020	9958	5	9959	9960	9961	9962	9963	9964	2227	9965	9966	
9967	286	9968	9969	608	9970	9971	64	9972	9973	920	1372	9974	9975	1021	1133	9976	9977	9978	
9979	2143	6076	9980	9981	9982	3344	715	9983	9984	5601	4625	3336	9985	1151	9986	9987	9988	3406	
9989	1662	9990	2622	1531	9991	5491	9992	1041	9993	2895	2116	2125	9994	9995	9996	9997	3648	9998	
9999	3167	2071	1313	10000	10001	3266	10002	10003	3702	5747	10004	10005	1064	5451	10006	1429	4822	3581	
4702	2596	643	10007	10008	10009	10010	3205	10011	253	10012	10013	10014	6009	10015	2844	10016	10017	825	
2186	3871	3103	2292	5803	10018	10019	5701	10020	3003	10021	5968	10022	10023	10024	10025	4903	10026	699	
522	2453	10027	2825	10028	10029	1	364	2115	10030	10031	10032	10033	10034	1557	1307	10035	10036	10037	
10038	10039	10040	2127	10041	10042	673	10043	10044	10045	3134	3464	10046	10047	10048	10049	3689	10050	10051	
2578	10052	10053	10054	10055	10056	10057	5213	1402	1024	10058	10059	6008	1601	10060	5473	3854	10061	729	
10062	10063	223	1242	1257	1303	10064	10065	10066	3110	10067	10068	3846	10069	10070	10071	10072	157	3508	
2862	2191	10073	563	10074	2745	10075	1519	3202	3737	1820	1738	10076	25	2753	10077	10078	10079	10080	
161	2144	686	10081	10082	10083	10084	10085	5821	2840	10086	4781	1228	5553	10087	2983	10088	10089	3783	

10090 3884 10091 10092 10093 828 10094 2716 4963 79 3859 1382 10095 4725 1329 936 10096 2672 2361
10097 10098 3151 1816 10099 1231 10100 10101 3500 1271 10102 3909 831 10103 2965 1009 10104 10105 10106
10107 10108 525 2955 10109 10110 335 5900 10111 5682 10112 350 1868 10113 10114 2588 3149 10115 4684
10116 10117 4607 10118 3894 10119 10120 10121 10122 10123 10124 2317 70 10125 1225 2396 1518 2921 5468
10126 10127 934 10128 10129 419 2878 437 10130 10131 10132 10133 3440 10134 5108 2397 602 10135 10136
1848 206 891 1463 5783 10137 10138 1144 10139 10140 3454 1391 1856 4599 10141 10142 10143 361 10144
10145 10146 10147 10148 10149 342 144 10150 10151 5700 10152 885 2632 2619 10153 10154 3758 3019 2952
10155 10156 5548 2405 2774 10157 10158 3055 10159 10160 3130 878 10161 10162 2220 4947 296 1049 10163
10164 10165 3504 10166 10167 10168 2518 4761 10169 1323 10170 10171 10172 10173 10174 2943 1939 10175 2195
2182 886 10176 10177 26 10178 2007 5287 1127 1521 4992 140 10179 5319 1353 212 3682 10180 3241
10181 10182 10183 10184 10185 10186 10187 132 1513 2080 2931 10188 10189 2013 10190 1728 10191 10192 10193
10194 2866 10195 10196 10197 10198 10199 10200 3838 10201 1614 10202 10203 369 10204 5805 10205 1293 1779
175 1527 2802 10206 2541 3738 1793 5721 10207 3124 2141 10208 3611 911 10209 710 10210 10211 1314
10212 849 3513 10213 3748 10214 1901 10215 2348 10216 10217 3105 2637 2974 10218 10219 10220 3101 10221
10222 10223 3131 3534 10224 10225 10226 2980 10227 10228 10229 10230 1849 5792 10231 5887 3496 1443 10232
992 10233 10234 4787 10235 5635 10236 2314 314 5958 10237 10238 2810 2901 10239 1369 2761 10240 10241
2696 5227 2855 486 10242 1481 10243 3828 2122 2340 1896 10244 3243 2574 3805 1721 2595 2747 2302
1675 10245 10246 10247 10248 5257 5125 3236 10249 5120 10250 3831 5420 10251 5424 10252 3250 996 10253
10254 883 10255 5435 1161 10256 188 10257 10258 10259 10260 10261 3583 1430 55 10262 3316 1931 10263
10264 10265 10266 10267 373 997 2167 283 10268 10269 10270 2311 2559 10271 10272 10273 10274 10275 10276
10277 10278 2925 10279 1510 10280 10281 2549 386 10282 10283 1175 10284 10285 10286 10287 3837 2218 10288
3913 2022 10289 10290 10291 10292 1101 10293 1792 10294 2057 325 1436 2017 664 10295 3091 10296 2771
10297 1092 10298 517 10299 3221 1298 10300 10301 10302 5649 3735 10303 1058 10304 10305 345 10306 1070
10307 581 1988 4923 2133 1465 2063 3154 10308 5344 441 10309 2509 10310 10311 10312 4835 10313 10314
5773 10315 10316 2874 10317 10318 2156 1507 5964 1844 10319 3127 10320 10321 3121 10322 10323 10324 5994
10325 10326 166 10327 291 10328 1038 265 3169 5026 10329 10330 10331 10332 10333 3307 1751 10334 3636
10335 10336 248 10337 10338 3230 10339 10340 10341 10342 1355 1337 10343 838 32 3086 2999 10344 10345
881 10346 10347 1237 1422 10348 10349 2828 10350 1571 823 10351 10352 2734 10353 10354 10355 1726 10356
10357 10358 10359 10360 10361 10362 10363 5784 3021 3539 10364 4641 10365 10366 2648 10367 1597 2026 5027
10368 2040 10369 10370 10371 10372 2257 10373 10374 2677 10375 809 3772 5206 10376 1593 10377 10378 10379
10380 2331 10381 10382 10383 10384 3522 3566 3874 2126 184 10385 10386 347 10387 10388 10389 3204 5177
10390 13 10391 10392 3244 1547 215 10393 10394 10395 4640 82 10396 10397 10 1514 10398 10399 10400
10401 1497 10402 10403 3873 10404 1919 10405 988 97 3002 10406 10407 10408 10409 2512 10410 10411 10412
10413 10414 10415 764 10416 1406 5097 10417 10418 2469 1512 2378 2764 10419 5846 10420 2461 65 10421
5908 10422 10423 3190 597 10424 3796 3436 5624 10425 10426 10427 1417 2209 10428 10429 4861 3663 3676
245 10430 10431 10432 3011 10433 10434 10435 10436 10437 10438 10439 10440 10441 2773 2820 10442 10443 10444
5525 1012 10445 10446 10447 10448 6011 10449 10450 10451 10452 10453 2772 10454 10455 10456 3394 10457 10458
10459 1152 2940 10460 1082 10461 3254 10462 2376 10463 3367 10464 5955 10465 5545 10466 10467 3498 10468
2755 10469 10470 2347 10471 10472 1379 2767 203 10473 4719 1227 10474 10475 3014 1016 10476 10477 10478
311 1972 3234 953 3115 10479 2194 3108 10480 200 731 10481 10482 10483 10484 599 2997 10485 10486
10487 10488 10489 10490 2642 2170 10491 10492 1732 194 10493 1002 3847 3004 2078 10494 3944 10495 10496
10497 832 267 10498 142 10499 5829 10500 1150 10501 10502 10503 3430 10504 10505 1195 10506 1056 1624
382 10507 5400 10508 2441 10509 1803 456 10510 3095 3303 272 5683 5847 3655 3775 1898 1720 10511
3891 10512 10513 4949 1604 10514 10515 10516 10517 10518 10519 10520 2282 821 10521 10522 138 10523 1723
3145 10524 10525 10526 10527 3555 2692 10528 1174 1349 1639 89 10529 5127 10530 546 3251 10531 2289
10532 10533 591 10534 10535 10536 2157 10537 10538 10539 1340 10540 1253 10541 3363 1863 5028 10542 10543
10544 3670 10545 2823 10546 10547 10548 10549 1790 10550 494 1854 2945 5209 4814 2990 10551 869 10552
10553 10554 10555 10556 10557 2879 10558 10559 3753 3023 10560 331 2922 10561 3350 10562 1341 10563 10564
10565 1739 10566 10567 862 10568 590 10569 10570 1317 288 2295 10571 2412 10572 10573 3490 114 10574
2235 5556 3860 10575 10576 10577 888 5509 360 10578 10579 3089 10580 10581 5990 10582 10583 3262 3199
1204 3059 10584 2501 10585 10586 10587 10588 10589 5661 3568 10590 10591 10592 2987 6059 3660 10593 10594
10595 10596 10597 10598 10599 1747 3314 10600 10601 5060 478 10602 1462 1388 10603 10604 1940 4928 10605
10606 814 1370 10607 10608 2713 10609 10610 10611 10612 10613 3865 10614 10615 10616 1877 1159 10617 10618
10619 952 5718 10620 10621 10622 10623 10624 10625 10626 1959 10627 822 10628 5892 10629 10630 10631 1573
10632 10633 1670 3380 4968 628 1439 4953 10634 3612 10635 1031 585 10636 2989 5216 493 10637 10638
2890 2731 10639 10640 4657 10641 10642 2151 10643 4998 1046 10644 10645 10646 4879 3533 784 10647 10648
2270 10649 10650 5907 4644 10651 4610 3245 10652 1652 5100 10653 10654 2975 10655 10656 3719 3752 10657
10658 10659 2352 10660 10661 10662 5363 230 2109 2627 10663 10664 10665 723 2357 2372 10666 10667 10668
1632 2202 10669 10670 10671 1071 10672 10673 5445 10674 6107 2618 10675 2379 10676 3730 10677 1705 800
2041 2906 10678 10679 2248 3722 10680 10681 10682 10683 10684 2356 10685 10686 2424 2697 5868 3582 10687
5516 10688 10689 10690 10691 10692 2021 1840 301 2752 10693 10694 10695 10696 10697 1077 10698 1730 10699
3777 10700 10701 3673 10702 942 10703 10704 4896 10705 10706 1853 661 2511 3843 10707 5324 10708 5957
5308 1362 10709 10710 2224 4794 10711 1857 10712 10713 3181 2165 10714 10715 10716 3799 10717 3360 3441
10718 5300 10719 10720 1094 1912 1745 10721 10722 2103 1089 778 10723 3677 485 10724 5487 5159 10725
1577 90 10726 10727 3325 2606 10728 5774 6015 644 1469 10729 2542 10730 865 10731 1697 10732 374
10733 1881 10734 393 3479 10735 4588 10736 3628 10737 1215 10738 10739 10740 2538 10741 10742 10743 10744
10745 277 2699 10746 10747 10748 399 1197 2791 10749 10750 5647 908 1664 10751 10752 10753 1698 10754
10755 2266 3457 2786 14 1646 1470 1357 10756 1549 3231 10757 10758 1862 10759 5446 1936 10760 10761
190 10762 3938 3456 10763 651 505 10764 10765 10766 1628 10767 10768 10769 3279 633 235 3183 371
5113 10770 3637 1750 10771 1393 232 559 3901 180 10772 10773 150 10774 10775 3552 10776 679 10777
10778 10779 10780 10781 3667 10782 10783 725 3746 10784 1612 1331 680 2768 10785 5969 2923 2150 1712
1320 10786 728 688 1878 529 607 4944 10787 5098 10788 10789 10790 3786 185 11 1277 10791 2325
2698 10792 10793 10794 10795 4649 1450 1938 10796 10797 5915 3323 1759 994 10798 3288 10799 10800 10801
3114 10802 10803 561 10804 10805 2881 2210 10806 2727 3724 2334 10807 10808 10809 10810 10811 3249 10812
1247 2287 10813 10814 10815 2273 5515 913 861 10816 5610 2600 10817 10818 10819 408 2135 129 10820
5940 772 10821 5182 1437 1921 10822 3289 10823 1667 10824 10825 10826 10827 3917 10828 10829 10830 3405

10831 10832 10833 10834 10835 3609 4638 10836 10837 5508 10838 10839 3832 10840 3658 10841 10842 10843 28
584 10844 10845 3404 10846 4778 10847 750 3603 4819 499 3024 10848 1118 5717 2050 5979 1489 2603
10849 10850 3478 774 3157 10851 10852 10853 284 10854 10855 10856 10857 351 10858 2703 1951 1866 4581
10859 890 10860 10861 1249 10862 10863 10864 10865 10866 10867 5483 10868 10869 10870 10871 10872 10873 4883
10874 10875 10876 10877 10878 4668 10879 10880 4731 10881 10882 10883 10884 10885 10886 10887 10888 10889 10890
4614 10891 10892 10893 10894 10895 10896 10897 5698 10898 10899 10900 10901 10902 10903 10904 10905 10906 10907
10908 10909 10910 10911 10912 10913 10914 10915 4619 4758 5652 10916 10917 5660 5290 10918 10919 10920 10921
5434 10922 10923 10924 10925 10926 10927 10928 10929 10930 5672 10931 10932 10933 10934 10935 10936 10937 10938
10939 10940 10941 10942 10943 10944 10945 10946 10947 10948 5007 4911 10949 10950 10951 10952 4630 10953 10954
10955 10956 10957 10958 10959 10960 10961 10962 10963 10964 10965 10966 10967 10968 10969 5457 10970 10971 10972
10973 10974 5334 10975 10976 10977 10978 10979 10980 10981 10982 5756 10983 10984 10985 10986 6083 10987 10988
5916 10989 10990 10991 10992 4909 10993 10994 10995 10996 10997 10998 10999 11000 11001 11002 11003 11004 11005
11006 5289 11007 11008 11009 11010 11011 5429 11012 11013 11014 11015 11016 11017 5901 11018 11019 11020 11021
11022 11023 11024 11025 11026 11027 5594 11028 11029 11030 11031 11032 11033 11034 11035 11036 11037 11038 11039
11040 4962 11041 11042 11043 11044 11045 4768 11046 5295 11047 11048 5284 11049 4704 11050 11051 11052 5140
11053 11054 5377 11055 6050 11056 11057 11058 11059 11060 11061 11062 11063 11064 11065 6005 11066 4646 11067
11068 11069 4613 11070 11071 11072 11073 11074 11075 11076 11077 11078 5671 11079 4695 11080 11081 11082 11083
11084 11085 11086 11087 11088 11089 11090 11091 11092 11093 11094 11095 11096 11097 11098 11099 11100 11101 11102
11103 11104 5343 11105 11106 11107 11108 11109 11110 11111 11112 6063 11113 11114 4716 4830 11115 11116 11117
11118 11119 11120 11121 11122 6041 11123 5962 11124 11125 11126 11127 11128 11129 11130 11131 11132 11133 11134
11135 11136 11137 11138 11139 11140 11141 11142 11143 4927 11144 5181 11145 11146 11147 11148 11149 11150 5966
4578 11151 11152 11153 11154 11155 5857 11156 11157 11158 11159 11160 11161 11162 11163 11164 11165 11166 11167
11168 11169 5927 11170 11171 11172 11173 5581 11174 11175 11176 11177 4740 11178 11179 11180 11181 11182 11183
11184 11185 11186 11187 11188 11189 11190 11191 11192 11193 4675 11194 11195 5082 11196 11197 5880 11198 11199
11200 11201 5540 11202 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11213 11214 11215 11216 11217
11218 11219 11220 11221 11222 11223 11224 11225 11226 11227 11228 11229 11230 11231 11232 11233 11234 11235 11236
11237 11238 11239 11240 11241 11242 11243 11244 11245 5971 11246 11247 11248 11249 11250 11251 11252 11253 11254
11255 11256 11257 11258 11259 11260 11261 11262 11263 11264 5894 11265 11266 11267 11268 11269 11270 11271 11272
11273 11274 4624 11275 11276 11277 11278 11279 4984 11280 11281 11282 11283 11284 11285 11286 11287 11288 11289
5676 11290 11291 11292 11293 11294 5628 11295 11296 11297 11298 11299 5018 4703 11300 11301 11302 11303 11304
11305 11306 11307 11308 11309 11310 11311 6102 11312 11313 5492 11314 11315 11316 11317 11318 5771 11319 11320
11321 11322 11323 11324 11325 11326 11327 11328 11329 11330 5275 4655 11331 11332 11333 11334 5136 5980 11335
11336 11337 11338 11339 11340 11341 11342 4734 11343 11344 11345 11346 11347 11348 5465 4860 11349 11350 11351
5199 11352 11353 11354 11355 11356 11357 5564 11358 11359 5731 11360 11361 5355 5762 11362 11363 11364 11365
11366 5183 11367 11368 11369 11370 11371 11372 11373 11374 11375 11376 11377 11378 4939 11379 11380 11381 11382
5357 11383 6032 5367 11384 11385 11386 11387 11388 11389 5588 11390 11391 11392 11393 11394 11395 11396 11397
11398 11399 11400 11401 11402 11403 11404 11405 11406 11407 11408 11409 11410 11411 11412 11413 11414 11415 11416
11417 11418 11419 11420 11421 11422 11423 5230 11424 11425 5495 11426 11427 11428 11429 11430 11431 11432 11433
11434 11435 11436 11437 11438 11439 4726 11440 11441 5003 4785 5240 11442 11443 11444 11445 11446 11447 11448
11449 11450 11451 5655 5493 11452 11453 4708 4580 11454 11455 11456 11457 11458 11459 11460 11461 5048 11462
11463 11464 11465 11466 11467 11468 11469 6061 11470 5250 11471 11472 11473 11474 11475 5149 11476 11477 11478
11479 11480 11481 11482 11483 11484 11485 11486 11487 11488 11489 11490 11491 11492 11493 11494 11495 11496 5608
11497 11498 11499 5882 5071 4986 11500 11501 11502 11503 11504 5043 11505 11506 5562 11507 11508 11509 11510
4786 11511 11512 11513 11514 11515 11516 11517 11518 11519 11520 11521 11522 11523 11524 5678 11525 11526 11527
11528 11529 11530 11531 11532 11533 11534 5702 11535 11536 11537 4938 11538 11539 11540 11541 11542 11543 5722
11544 11545 11546 11547 11548 11549 11550 11551 11552 11553 11554 11555 11556 11557 11558 11559 11560 11561 11562
11563 11564 11565 11566 11567 11568 11569 11570 11571 11572 11573 11574 11575 11576 11577 11578 11579 11580 11581
11582 11583 5237 11584 11585 11586 11587 11588 11589 11590 11591 11592 11593 11594 11595 5148 11596 11597 11598
11599 11600 11601 11602 5685 11603 4666 11604 11605 11606 11607 11608 11609 11610 11611 11612 11613 11614 11615
11616 11617 11618 11619 11620 11621 11622 11623 11624 11625 11626 11627 11628 11629 11630 11631 11632 5086 5621
11633 11634 11635 11636 11637 11638 11639 11640 5517 11641 11642 11643 11644 11645 5200 11646 11647 11648 11649
11650 11651 11652 11653 11654 11655 5744 11656 4602 11657 11658 11659 11660 11661 11662 11663 11664 5035 11665
11666 11667 11668 4586 5726 11669 4637 4603 11670 11671 11672 11673 11674 11675 11676 11677 11678 11679 11680
11681 11682 11683 11684 11685 11686 11687 11688 11689 11690 11691 4991 5674 11692 11693 11694 11695 5552 11696
5147 11697 11698 11699 11700 11701 5380 11702 11703 11704 11705 11706 11707 11708 11709 5811 11710 11711 11712
11713 11714 11715 11716 11717 11718 11719 11720 11721 11722 11723 11724 5612 11725 11726 5623 11727 5283 11728
6052 11729 11730 4869 11731 11732 11733 11734 11735 11736 11737 11738 11739 11740 11741 11742 5103 11743 11744
11745 11746 11747 11748 11749 11750 4867 11751 11752 11753 11754 11755 11756 11757 11758 11759 11760 11761 11762
11763 11764 11765 11766 11767 5834 11768 4902 11769 11770 11771 4662 4974 4693 11772 11773 5998 11774 11775
11776 11777 5276 11778 11779 11780 4779 11781 11782 11783 11784 11785 4954 5309 11786 11787 5282 11788 11789
11790 11791 11792 6099 11793 6060 5091 11794 5474 11795 11796 11797 11798 5325 11799 11800 11801 11802 11803
11804 11805 11806 11807 4611 11808 11809 5925 5212 11810 5706 11811 11812 11813 11814 11815 5806 11816 5724
11817 11818 11819 11820 11821 11822 11823 11824 11825 11826 11827 11828 11829 5557 11830 11831 11832 4626 11833
5101 11834 11835 11836 11837 11838 5266 11839 11840 11841 11842 11843 11844 11845 11846 11847 11848 11849 11850
11851 11852 11853 11854 11855 11856 11857 11858 11859 11860 11861 11862 11863 11864 11865 11866 11867 5235 11868
11869 11870 11871 11872 11873 11874 11875 5709 11876 11877 11878 4885 11879 11880 11881 11882 11883 11884 11885
11886 11887 11888 11889 11890 11891 11892 11893 11894 11895 11896 11897 11898 11899 11900 11901 11902 11903 11904
11905 5062 11906 11907 11908 11909 11910 11911 11912 11913 11914 11915 5328 11916 11917 5053 11918 11919 11920
11921 5470 11922 11923 11924 11925 11926 11927 11928 11929 11930 11931 11932 11933 5286 11934 11935 11936 11937
11938 11939 11940 11941 11942 11943 11944 11945 11946 11947 11948 11949 11950 11951 11952 11953 11954 4584 11955
11956 11957 11958 11959 11960 11961 11962 11963 11964 11965 11966 11967 11968 11969 5432 11970 11971 11972 11973
11974 5067 11975 11976 11977 11978 11979 11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
11992 11993 11994 5567 5830 11995 11996 11997 11998 11999 12000 12001 12002 12003 12004 12005 12006 12007 12008
12009 12010 12011 12012 12013 12014 12015 12016 12017 4999 12018 12019 12020 12021 5302 12022 12023 12024 12025
12026 12027 12028 12029 12030 12031 12032 12033 5255 12034 5462 12035 12036 12037 12038 12039 12040 12041 5114
12042 12043 12044 12045 12046 12047 12048 12049 12050 6010 12051 12052 12053 12054 12055 12056 12057 12058 12059

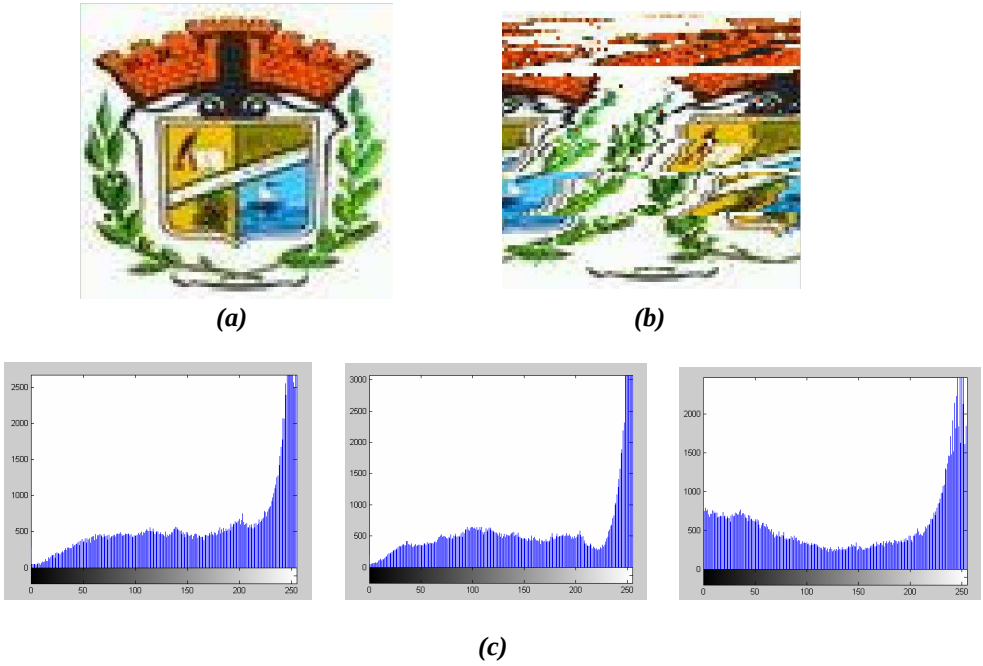
12060 12061 12062 12063 12064 5150 5299 12065 12066 12067 12068 12069 12070 12071 12072 12073 12074 12075 12076
5864 12077 12078 12079 5555 12080 12081 12082 5191 5912 12083 12084 12085 12086 12087 12088 12089 12090 12091
12092 12093 12094 12095 12096 4881 12097 12098 12099 12100 12101 12102 12103 12104 12105 12106 12107 12108 12109
4825 12110 12111 12112 12113 12114 12115 5455 12116 12117 5590 12118 12119 12120 5840 12121 12122 5145 12123
12124 12125 12126 12127 12128 12129 12130 12131 12132 12133 12134 12135 12136 12137 12138 4823 4764 12139 5087
12140 12141 12142 12143 12144 12145 12146 12147 12148 12149 12150 4686 12151 12152 12153 12154 4899 12155 12156
12157 12158 12159 12160 12161 12162 12163 12164 12165 12166 12167 12168 6025 4948 12169 12170 5577 12171 5395
6100 12172 12173 12174 12175 12176 12177 12178 12179 12180 12181 12182 12183 12184 12185 12186 12187 12188 12189
12190 12191 12192 12193 12194 12195 12196 12197 12198 12199 12200 12201 12202 12203 12204 12205 12206 12207 12208
12209 12210 12211 5327 12212 12213 12214 5630 6026 4925 12215 12216 12217 12218 12219 12220 12221 12222 12223
12224 12225 12226 12227 12228 12229 12230 12231 12232 5095 12233 5360 5845 12234 12235 6092 5135 12236 12237
12238 12239 12240 12241 12242 12243 12244 12245 12246 12247 12248 12249 12250 12251 12252 5898 12253 12254 12255
12256 12257 12258 12259 12260 12261 12262 12263 5163 12264 12265 12266 12267 12268 12269 12270 12271 12272 12273
12274 12275 12276 12277 5954 12278 12279 12280 12281 12282 12283 12284 4593 12285 12286 12287 12288 12289 12290
12291 12292 12293 12294 12295 4979 5526 5039 12296 12297 12298 12299 12300 12301 12302 12303 4674 4585 12304
12305 12306 12307 12308 5862 12309 5041 12310 12311 12312 12313 12314 12315 12316 12317 12318 12319 12320 12321
12322 12323 12324 12325 12326 12327 12328 5667 12329 5176 12330 5869 12331 12332 12333 12334 12335 5378 12336
12337 12338 12339 12340 12341 12342 5022 12343 5758 12344 12345 5123 12346 12347 5034 12348 12349 5663 12350
12351 12352 5096 5345 12353 12354 12355 12356 12357 12358 12359 5339 12360 12361 5403 5000 12362 12363 12364
12365 12366 12367 12368 12369 12370 12371 12372 12373 12374 12375 12376 12377 12378 12379 12380 12381 12382 12383
12384 12385 12386 12387 12388 12389 12390 12391 12392 12393 12394 12395 12396 12397 12398 5679 12399 12400 12401
12402 4905 12403 12404 12405 12406 12407 12408 4639 4889 12409 12410 12411 12412 12413 12414 5410 12415 12416
12417 12418 12419 12420 12421 4579 12422 12423 12424 12425 12426 12427 12428 12429 12430 12431 12432 5589 12433
12434 12435 12436 12437 12438 5616 12439 12440 12441 12442 12443 12444 12445 12446 12447 12448 12449 12450 12451
12452 12453 12454 12455 4891 12456 12457 12458 12459 12460 12461 12462 12463 12464 12465 12466 12467 5524 12468
5987 12469 12470 12471 12472 12473 12474 12475 12476 12477 12478 12479 12480 12481 12482 5316 12483 12484 12485
12486 4658 12487 12488 12489 4746 12490 12491 4745 12492 12493 12494 12495 12496 12497 12498 12499 12500 12501
12502 5002 12503 12504 12505 12506 12507 12508 12509 12510 5037 12511 12512 12513 12514 12515 12516 12517 12518
12519 4818 4791 12520 12521 12522 12523 12524 5761 12525 12526 12527 12528 12529 12530 12531 12532 12533 12534
12535 12536 5602 5072 12537 12538 5055 12539 12540 12541 12542 12543 12544 12545 12546 12547 5677 12548 12549
12550 12551 12552 12553 12554 12555 12556 12557 5178 4983 12558 12559 5093 12560 12561 12562 12563 5501 12564
12565 12566 12567 12568 12569 12570 12571 12572 12573 12574 6075 12575 12576 12577 12578 12579 12580 12581 12582
12583 12584 12585 12586 12587 12588 12589 12590 5341 12591 12592 12593 12594 12595 6036 5859 12596 12597 12598
5636 12599 12600 12601 4692 12602 12603 12604 12605 12606 12607 12608 12609 5728 12610 12611 12612 12613 12614
12615 5520 12616 12617 12618 12619 12620 12621 12622 5475 12623 12624 12625 12626 12627 12628 12629 12630 12631
12632 12633 12634 12635 5743 12636 12637 5390 12638 12639 12640 12641 12642 12643 12644 12645 12646 4741 12647
12648 12649 12650 12651 12652 12653 12654 12655 12656 12657 12658 12659 12660 12661 12662 12663 12664 4851 12665
12666 12667 12668 12669 12670 12671 5768 12672 12673 12674 12675 12676 12677 12678 12679 12680 12681 12682 12683
12684 12685 12686 12687 5640 12688 5751 12689 5575 12690 12691 12692 12693 12694 12695 12696 12697 12698 12699
12700 12701 12702 12703 12704 12705 12706 12707 12708 12709 12710 12711 5471 12712 12713 4752 5600 12714 5045
5831 12715 12716 12717 12718 12719 12720 12721 12722 5504 12723 12724 12725 12726 12727 12728 12729 12730 12731
12732 12733 5742 12734 12735 12736 12737 12738 5745 12739 12740 12741 12742 12743 12744 12745 12746 12747 12748
5085 12749 12750 12751 5298 12752 12753 12754 12755 5658 12756 12757 12758 5697 12759 5394 4798 12760 12761
12762 12763 12764 12765 12766 12767 12768 12769 12770 12771 5983 12772 12773 12774 12775 12776 12777 12778 12779
12780 12781 12782 12783 4717 12784 12785 12786 5346 5735 12787 12788 12789 12790 12791 12792 12793 12794 12795
12796 12797 12798 12799 5375 12800 12801 12802 12803 12804 5974 6055 12805 12806 12807 12808 4782 12809 12810
12811 12812 12813 12814 12815 12816 5809 12817 12818 12819 12820 12821 12822 12823 12824 4727 12825 12826 12827
12828 12829 12830 12831 12832 12833 12834 12835 12836 12837 12838 12839 12840 12841 12842 5586 12843 12844 12845
12846 12847 12848 5450 12849 12850 12851 12852 5496 12853 12854 12855 12856 12857 12858 12859 12860 12861 12862
12863 12864 12865 12866 12867 12868 12869 12870 12871 12872 12873 12874 12875 12876 12877 5443 12878 12879 12880
12881 12882 12883 5128 12884 12885 12886 12887 12888 12889 12890 5236 5076 12891 12892 12893 12894 12895 12896
12897 12898 12899 12900 12901 12902 12903 4595 12904 12905 12906 12907 12908 12909 12910 12911 12912 12913 12914
12915 12916 4849 12917 12918 12919 12920 12921 12922 12923 12924 5478 12925 12926 12927 12928 12929 12930 12931
12932 12933 12934 12935 12936 12937 12938 12939 12940 4956 12941 12942 12943 4900 12944 12945 12946 12947 12948
12949 6070 12950 12951 12952 12953 12954 12955 12956 12957 12958 5823 12959 12960 12961 12962 12963 12964 12965
12966 12967 12968 5249 12969 12970 12971 12972 12973 12974 12975 5786 12976 12977 12978 12979 12980 12981 12982
12983 12984 12985 12986 12987 12988 12989 12990 5023 12991 5819 12992 12993 6004 12994 12995 12996 5951 12997
12998 12999 13000 13001 13002 13003 13004 13005 5318 13006 13007 5961 13008 13009 13010 13011 13012 13013 13014
13015 13016 13017 13018 13019 13020 13021 13022 13023 5311 4628 13024 13025 13026 13027 13028 13029 13030 5913
4865 13031 13032 13033 13034 13035 5008 13036 5644 13037 4985 13038 13039 13040 6028 5479 13041 13042 13043
13044 13045 13046 13047 13048 13049 13050 13051 13052 13053 13054 13055 13056 13057 13058 13059 13060 13061 13062
13063 13064 13065 13066 13067 13068 13069 13070 13071 13072 13073 13074 13075 13076 13077 13078 13079 13080 5194
13081 13082 13083 13084 13085 13086 5444 13087 13088 13089 5226 13090 13091 13092 13093 13094 13095 13096 13097
13098 13099 4651 13100 13101 13102 13103 13104 5141 4982 13105 13106 13107 13108 13109 6012 13110 13111 13112
13113 13114 13115 13116 13117 13118 13119 13120 13121 13122 13123 13124 13125 13126 5374 13127 13128 13129 13130
13131 13132 13133 4989 13134 13135 13136 13137 13138 5825 13139 13140 13141 13142 13143 13144 13145 5712 13146
5625 13147 13148 13149 13150 13151 13152 13153 13154 13155 5812 13156 13157 13158 13159 13160 13161 5247 13162
13163 13164 13165 13166 13167 13168 13169 13170 13171 13172 13173 13174 13175 13176 13177 4777 13178 13179 13180
13181 13182 13183 5215 5937 13184 13185 13186 13187 13188 13189 13190 13191 13192 6108 13193 13194 4707 13195
13196 13197 13198 13199 5943 13200 13201 13202 13203 4894 13204 13205 13206 13207 13208 13209 13210 13211 13212
13213 13214 13215 13216 13217 13218 13219 13220 13221 13222 13223 13224 13225 13226 13227 13228 13229 4729 13230
5899 13231 13232 13233 13234 13235 13236 13237 13238 13239 13240 5218 13241 13242 13243 13244 13245 13246 13247
5453 13248 13249 13250 13251 13252 4813 13253 13254 13255 13256 13257 13258 13259 13260 13261 13262 13263 13264
13265 13266 13267 13268 13269 4837 13270 13271 13272 13273 13274 13275 13276 13277 13278 13279 13280 13281 5036
13282 13283 13284 13285 13286 13287 13288 13289 13290 5441 13291 13292 13293 13294 4590 13295 13296 13297 13298
13299 13300 13301 13302 5169 13303 13304 13305 13306 13307 13308 13309 13310 13311 13312 13313 13314 13315 13316

13317 13318 13319 13320 13321 13322 13323 13324 13325 13326 13327 13328 13329 13330 13331 13332 5402 13333 13334
13335 13336 13337 13338 13339 13340 13341 13342 5347 13343 13344 13345 13346 13347 13348 13349 13350 13351 13352
13353 13354 13355 5922 13356 13357 13358 13359 13360 4783 13361 5369 13362 13363 13364 13365 13366 13367 13368
13369 13370 13371 13372 13373 13374 5323 13375 13376 13377 13378 13379 13380 13381 5592 13382 13383 13384 13385
13386 13387 5563 13388 13389 13390 13391 13392 13393 13394 13395 13396 13397 13398 13399 13400 13401 13402 13403
13404 13405 13406 13407 5268 13408 6035 13409 5111 13410 13411 4653 13412 4906 13413 13414 13415 13416 13417
4636 4797 13418 13419 13420 13421 13422 13423 13424 13425 13426 13427 13428 13429 13430 13431 13432 13433 13434
13435 13436 13437 13438 13439 13440 13441 13442 13443 13444 13445 13446 13447 13448 13449 13450 5820 13451 13452
13453 13454 13455 13456 13457 13458 13459 13460 13461 13462 13463 13464 13465 13466 13467 13468 13469 13470 13471
13472 13473 13474 13475 13476 13477 5558 13478 13479 5764 13480 13481 13482 13483 13484 13485 13486 13487 4681
13488 13489 13490 13491 13492 13493 13494 13495 13496 5854 13497 13498 13499 13500 4884 13501 4800 13502 13503
13504 13505 13506 13507 13508 13509 13510 13511 13512 13513 13514 13515 13516 13517 13518 13519 13520 13521 13522
13523 13524 13525 13526 13527 13528 5886 13529 5670 13530 13531 13532 13533 5791 13534 13535 13536 13537 13538
13539 13540 13541 13542 13543 13544 13545 13546 13547 13548 13549 13550 13551 13552 13553 13554 13555 13556 13557
13558 13559 13560 13561 5883 13562 13563 13564 13565 13566 13567 13568 13569 13570 13571 13572 13573 13574 13575
13576 13577 13578 13579 13580 5001 13581 13582 13583 13584 4821 13585 13586 13587 13588 13589 13590 13591 13592
13593 13594 13595 13596 13597 13598 13599 13600 13601 13602 13603 13604 13605 13606 5425 13607 13608 13609 13610
13611 13612 13613 13614 13615 13616 13617 13618 13619 13620 13621 6034 13622 13623 13624 13625 13626 13627 13628
13629 13630 13631 4765 4642 13632 4827 13633 13634 6039 4836 13635 13636 5117 13637 13638 5837 13639 13640
13641 13642 13643 13644 13645 13646 13647 13648 13649 13650 13651 13652 13653 13654 13655 13656 13657 5551 13658
13659 13660 13661 5267 13662 13663 13664 5904 13665 13666 13667 13668 13669 13670 13671 13672 13673 13674 13675
13676 13677 13678 13679 5385 13680 13681 13682 5595 13683 13684 13685 13686 13687 13688 13689 13690 13691 13692
13693 13694 13695 5920 13696 13697 13698 13699 13700 13701 13702 13703 13704 13705 13706 13707 13708 4605 13709
13710 13711 13712 13713 13714 13715 13716 13717 5963 13718 13719 13720 13721 13722 5387 13723 13724 13725 13726
13727 13728 13729 13730 13731 13732 13733 5903 13734 13735 13736 13737 13738 5817 13739 5408 5269 13740 13741
13742 13743 13744 13745 13746 13747 13748 13749 13750 13751 13752 13753 13754 13755 13756 13757 13758 13759 13760
13761 13762 5646 13763 13764 13765 13766 13767 13768 13769 13770 13771 13772 13773 13774 13775 13776 13777 13778
13779 5597 13780 13781 13782 13783 4673 5956 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 4696
13794 13795 13796 5497 13797 13798 4712 5799 13799 13800 13801 13802 13803 13804 13805 13806 13807 13808 13809
13810 13811 13812 13813 13814 13815 13816 13817 13818 13819 13820 13821 13822 13823 13824 13825 13826 13827 13828
13829 5442 4701 13830 4859 13831 6064 13832 13833 13834 13835 13836 13837 13838 13839 13840 13841 13842 13843
13844 13845 13846 13847 13848 13849 13850 13851 13852 13853 13854 13855 13856 13857 5186 13858 13859 13860 13861
13862 13863 13864 5431 13865 13866 13867 13868 5449 13869 4995 13870 13871 13872 13873 13874 13875 13876 5604
13877 13878 13879 5406 13880 13881 13882 13883 13884 13885 13886 13887 13888 13889 13890 13891 4863 13892 13893
5695 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903 13904 13905 5815 13906 13907 13908 13909 13910
13911 13912 13913 13914 5571 13915 13916 13917 5208 13918 13919 13920 5109 13921 13922 13923 13924 13925 13926
13927 13928 13929 13930 13931 13932 13933 13934 13935 13936 13937 13938 13939 13940 5224 13941 13942 13943 13944
13945 13946 13947 5407 13948 13949 13950 13951 5189 4620 13952 4582 13953 13954 13955 13956 13957 13958 13959
13960 13961 13962 13963 13964 13965 13966 13967 13968 13969 13970 13971 5536 13972 13973 13974 13975 13976 13977
13978 13979 13980 13981 13982 5533 13983 13984 13985 5042 13986 13987 13988 13989 13990 13991 5776 13992 13993
13994 13995 4803 13996 13997 5167 13998 4743 13999 14000 14001 14002 14003 14004 14005 14006 14007 14008 14009
14010 14011 14012 14013 14014 14015 14016 14017 5274 14018 14019 14020 14021 14022 14023 14024 14025 14026 5011
14027 14028 14029 14030 5168 14031 14032 5550 14033 14034 14035 5251 14036 14037 14038 14039 14040 14041 14042
5014 14043 14044 14045 14046 14047 14048 14049 4990 14050 14051 14052 5933 14053 14054 14055 14056 14057 14058
5813 14059 14060 14061 14062 14063 5015 14064 14065 14066 5531 14067 14068 14069 14070 14071 14072 14073 14074
14075 14076 14077 14078 14079 14080 14081 14082 14083 14084 14085 14086 14087 14088 14089 5716 14090 14091 5924
14092 14093 14094 14095 14096 5749 14097 14098 14099 14100 14101 14102 14103 14104 14105 14106 14107 14108 14109
14110 14111 14112 4918 5619 14113 14114 14115 5593 14116 14117 5166 14118 14119 14120 14121 14122 14123 14124
14125 14126 14127 14128 14129 14130 14131 14132 14133 14134 14135 14136 14137 14138 14139 14140 14141 14142 14143
14144 14145 14146 14147 14148 14149 14150 14151 14152 14153 14154 14155 14156 14157 14158 14159 14160 14161 14162
14163 14164 14165 14166 5081 14167 14168 14169 14170 14171 14172 4852 5256 14173 14174 14175 14176 14177 14178
14179 4872 14180 14181 4996 14182 14183 14184 14185 14186 14187 14188 5580 14189 14190 14191 14192 14193 14194
14195 14196 14197 14198 14199 5472 5231 14200 14201 14202 14203 14204 14205 14206 14207 14208 14209 14210 5232
4750 14211 14212 14213 14214 5538 14215 14216 14217 14218 14219 14220 5788 14221 14222 14223 14224 4676 14225
14226 14227 14228 14229 14230 14231 14232 14233 14234 14235 4720 14236 14237 14238 4987 14239 14240 14241 14242
4955 14243 14244 14245 14246 14247 14248 14249 14250 14251 14252 14253 14254 14255 5004 14256 14257 14258 14259
14260 14261 14262 5918 14263 14264 14265 14266 14267 14268 14269 14270 5197 14271 14272 5866 14273 14274 14275
14276 14277 4789 14278 14279 14280 14281 14282 14283 14284 14285 14286 14287 14288 14289 14290 14291 14292 14293
14294 14295 14296 14297 14298 14299 14300 14301 14302 5241 14303 14304 14305 14306 14307 14308 14309 14310 14311
14312 14313 14314 6058 14315 14316 14317 5642 14318 14319 14320 14321 14322 14323 14324 14325 14326 14327 6040
14328 14329 5569 14330 14331 14332 14333 14334 14335 14336 14337 14338 4631 5315 14339 14340 14341 14342 14343
14344 14345 14346 14347 14348 14349 14350 14351 14352 14353 14354 14355 14356 14357 14358 14359 14360 14361 14362
14363 14364 5534 14365 14366 4775 14367 5993 14368 14369 14370 14371 14372 14373 6085 14374 14375 14376 5337
14377 5708 14378 14379 14380 14381 14382 14383 14384 14385 6069 14386 14387 14388 14389 14390 14391 14392 14393
14394 14395 14396 14397 14398 14399 14400 14401 14402 14403 14404 14405 14406 14407 14408 14409 5529 14410 14411
14412 14413 14414 14415 14416 14417 14418 14419 14420 14421 14422 14423 14424 14425 14426 14427 14428 5755 5732
14429 5356 14430 14431 14432 5192 14433 14434 14435 5733 14436 5351 14437 14438 14439 14440 14441 14442 14443
14444 14445 14446 14447 5598 4816 14448 14449 14450 14451 5603 14452 14453 14454 5448 14455 14456 14457 14458
14459 14460 14461 14462 14463 14464 14465 14466 14467 14468 14469 14470 5765 5370 4965 4715 14471 5416 14472
14473 14474 14475 14476 4805 14477 14478 14479 14480 14481 14482 14483 14484 14485 14486 14487 14488 14489 5503
14490 14491 14492 5982 14493 4678 14494 14495 14496 14497 14498 14499 14500 14501 14502 14503 14504 14505 14506
14507 14508 5822 14509 14510 14511 14512 14513 14514 14515 14516 5591 5361 14517 14518 14519 14520 14521 14522
14523 14524 14525 14526 14527 14528 14529 14530 14531 14532 14533 14534 14535 14536 14537 14538 14539 14540 5278
14541 14542 14543 14544 14545 14546 14547 14548 14549 14550 14551 14552 14553 14554 14555 14556 14557 14558 14559
14560 14561 14562 14563 4921 14564 14565 14566 14567 14568 14569 14570 14571 14572 14573 14574 14575 4988 14576
14577 14578 14579 14580 14581 14582 14583 14584 14585 5513 14586 14587 14588 14589 14590 14591 5875 5050 14592

14593 14594 14595 14596 14597 14598 14599 14600 14601 14602 14603 14604 5572 14605 14606 14607 14608 14609 14610
5364 14611 14612 4645 14613 14614 14615 14616 14617 14618 14619 14620 14621 14622 14623 14624 14625 14626 14627
14628 14629 14630 14631 14632 14633 14634 14635 14636 14637 14638 14639 14640 14641 14642 14643 14644 14645 14646
14647 14648 14649 14650 14651 14652 14653 14654 14655 14656 14657 14658 14659 14660 14661 14662 14663 5512 14664
14665 14666 14667 14668 5836 14669 14670 14671 14672 5737 14673 14674 14675 14676 4842 14677 14678 14679 14680
14681 14682 4737 14683 14684 4833 14685 14686 14687 14688 5359 14689 14690 14691 14692 14693 14694 14695 14696
4650 14697 5293 14698 14699 14700 14701 14702 14703 5063 14704 5941 14705 14706 14707 14708 5938 14709 14710
14711 14712 5710 14713 14714 14715 14716 14717 14718 6086 14719 14720 14721 14722 14723 14724 14725 14726 14727
14728 14729 14730 14731 14732 14733 14734 14735 14736 14737 14738 14739 14740 14741 14742 14743 14744 14745 14746
14747 14748 14749 14750 14751 14752 14753 14754 14755 14756 14757 14758 14759 5165 14760 14761 14762 14763 14764
14765 14766 6048 14767 14768 14769 6071 14770 14771 14772 14773 14774 14775 14776 14777 14778 14779 14780 14781
14782 6073 14783 14784 14785 14786 14787 14788 6101 14789 14790 4854 14791 14792 14793 14794 14795 14796 14797
14798 4617 14799 5753 14800 14801 14802 14803 14804 14805 14806 14807 14808 14809 14810 14811 14812 14813 14814
14815 14816 14817 14818 14819 14820 14821 14822 14823 14824 14825 4598 14826 14827 5233 14828 14829 14830 14831
14832 14833 14834 14835 14836 14837 14838 14839 14840 5017 14841 14842 14843 14844 14845 14846 14847 14848 14849
5107 14850 14851 14852 14853 14854 14855 14856 14857 14858 14859 14860 14861 14862 14863 14864 14865 14866 14867
5752 5511 5522 14868 14869 14870 14871 14872 14873 14874 14875 5313 14876 14877 14878 14879 14880 14881 14882
4796 14883 14884 14885 14886 14887 4840 14888 14889 14890 14891 14892 14893 14894 14895 14896 14897 14898 14899
14900 14901 14902 14903 14904 14905 14906 14907 14908 14909 14910 14911 14912 14913 5152 14914 14915 14916 4687
14917 14918 14919 14920 4722 14921 14922 14923 14924 5537 14925 14926 14927 14928 14929 14930 14931 14932 14933
14934 14935 14936 14937 14938 14939 14940 14941 14942 14943 14944 14945 14946 14947 14948 14949 14950 14951 14952
14953 14954 14955 14956 14957 14958 14959 4951 4759 14960 14961 14962 14963 14964 5144 5464 14965 14966 14967
14968 14969 14970 14971 14972 14973 14974 14975 6042 14976 14977 14978 14979 14980 14981 14982 14983 14984 14985
14986 14987 14988 14989 6066 14990 14991 14992 14993 14994 14995 14996 14997 14998 14999 15000 15001 15002 15003
15004 15005 15006 15007 15008 15009 15010 15011 15012 15013 15014 15015 15016 15017 5044 15018 15019 15020 15021
15022 15023 5826 15024 15025 15026 15027 15028 15029 15030 15031 5544 15032 5219 15033 15034 15035 15036 5396
15037 15038 15039 15040 15041 15042 15043 15044 15045 15046 15047 4714 15048 15049 15050 15051 15052 15053 15054
15055 4922 15056 15057 15058 15059 15060 15061 15062 15063 15064 15065 15066 15067 5222 4855 15068 15069 15070
15071 4937 15072 15073 4618 15074 15075 15076 5587 15077 6089 15078 15079 15080 15081 5656 15082 15083 15084
15085 15086 15087 15088 15089 15090 5578 15091 15092 15093 15094 15095 15096 15097 4643 5605 15098 15099 15100
15101 15102 15103 15104 15105 15106 15107 15108 15109 15110 15111 15112 15113 6067 15114 15115 15116 15117 15118
15119 15120 15121 15122 15123 15124 15125 15126 5860 15127 15128 15129 15130 15131 15132 15133 15134 15135 15136
15137 15138 5953 15139 15140 15141 15142 15143 15144 15145 15146 5030 15147 15148 15149 15150 15151 5006 5329
5121 15152 15153 15154 15155 4762 15156 15157 15158 5730 15159 5013 15160 15161 15162 4917 15163 15164 15165
15166 15167 15168 15169 15170 15171 15172 15173 15174 15175 15176 15177 15178 15179 15180 6054 15181 15182 15183
15184 15185 15186 15187 15188 5611 15189 15190 15191 15192 5824 15193 15194 15195 5419 15196 15197 15198 15199
15200 15201 15202 15203 15204 15205 15206 15207 15208 15209 15210 4685 15211 15212 15213 15214 15215 15216 5909
15217 15218 15219 15220 15221 15222 15223 15224 15225 15226 15227 15228 5787 15229 15230 15231 15232 15233 15234
15235 15236 15237 5389 15238 15239 15240 15241 15242 15243 15244 15245 15246 15247 4683 4832 15248 15249 15250
5494 15251 5687 15252 15253 15254 5723 15255 15256 15257 15258 15259 15260 15261 15262 15263 4757 15264 4754
15265 15266 15267 15268 15269 15270 15271 15272 15273 15274 15275 15276 15277 15278 15279 15280 15281 15282 15283
15284 4591 15285 15286 15287 15288 15289 15290 15291 15292 15293 6077 15294 15295 15296 15297 4606 15298 15299
15300 5404 15301 15302 15303 15304 15305 15306 15307 4670 15308 6098 15309 15310 4589 15311 15312 15313 15314
6074 15315 15316 15317 15318 15319 15320 15321 15322 15323 15324 15325 15326 15327 15328 15329 15330 15331 4583
15332 15333 15334 15335 15336 5888 15337 15338 15339 15340 15341 15342 15343 15344 15345 15346 5970 15347 15348
15349 15350 15351 15352 15353 5688 15354 15355 15356 15357 15358 15359 15360 15361 15362 15363 15364 15365 15366
15367 15368 15369 15370 15371 15372 15373 15374 15375 15376 15377 5720 15378 15379 15380 15381 15382 15383 15384
15385 15386 15387 5423 15388 15389 15390 15391 15392 15393 15394 15395 15396 15397 15398 4817 5793 4698 15399
15400 15401 15402 15403 5210 15404 5574 15405 15406 5740 4941 15407 15408 15409 15410 6109 15411 15412 15413
15414 15415 15416 15417 15418 15419 15420 15421 15422 15423 15424 15425 15426 5202 15427 15428 15429 15430 15431
15432 15433 15434 15435 15436 15437 15438 5211 15439 15440 15441 5073 15442 15443 5779 15444 5696 15445 15446
15447 4629 5398 15448 15449 15450 15451 15452 15453 15454 15455 15456 15457 15458 15459 15460 15461 15462 15463
15464 15465 15466 15467 15468 15469 15470 15471 15472 15473 15474 15475 15476 4845 15477 15478 15479 15480 5772
15481 5596 15482 15483 15484 15485 15486 15487 15488 5949 15489 15490 15491 15492 15493 15494 15495 6021 15496
15497 15498 15499 15500 15501 15502 15503 15504 15505 5818 15506 15507 15508 15509 5326 15510 15511 15512 15513
15514 15515 4994 15516 15517 15518 15519 15520 15521 15522 5154 15523 4663 15524 15525 15526 5734 15527 15528
15529 15530 15531 15532 5059 15533 15534 4943 15535 15536 15537 15538 15539 15540 15541 15542 15543 15544 15545
5273 15546 15547 15548 15549 15550 15551 15552 4760 15553 5399 15554 6056 15555 15556 15557 5303 15558 15559
15560 15561 15562 15563 15564 5705 15565 15566 15567 15568 5852 5842 15569 15570 15571 15572 5838 15573 15574
15575 15576 15577 15578 5981 15579 15580 15581 15582 15583 15584 15585 15586 15587 15588 15589 15590 15591 15592
15593 15594 5350 15595 15596 15597 15598 5122 15599 15600 15601 4774 15602 15603 15604 15605 15606 15607 15608
15609 15610 15611 15612 15613 15614 15615 15616 15617 15618 15619 15620 15621 5280 15622 5748 15623 15624 15625
15626 15627 15628 15629 5996 15630 5863 15631 15632 15633 15634 15635 15636 15637 15638 15639 15640 15641 15642
15643 15644 5653 15645 15646 15647 15648 4940 15649 15650 15651 15652 15653 15654 15655 15656 15657 15658 15659
15660 15661 15662 15663 15664 5876 15665 15666 15667 15668 15669 15670 15671 15672 15673 5984 15674 15675 15676
15677 15678 15679 15680 5725 15681 15682 15683 15684 15685 15686 15687 15688 15689 15690 15691 15692 15693 15694
15695 15696 15697 5659 15698 15699 15700 15701 15702 15703 15704 15705 15706 15707 15708 15709 15710 5214 15711
15712 15713 15714 15715 15716 4592 15717 15718 5777 15719 15720 15721 15722 5271 15723 15724 15725 15726 15727
15728 15729 15730 4870 15731 15732 15733 15734 15735 15736 15737 15738 15739 15740 15741 15742 15743 15744 15745
15746 15747 15748 15749 15750 15751 15752 15753 15754 15755 15756 15757 15758 15759 15760 15761 15762 15763 15764
15765 15766 15767 15768 15769 15770 15771 15772 15773 15774 4647 15775 15776 15777 4931 15778 15779 15780 15781
15782 15783 4672 15784 15785 15786 15787 15788 15789 15790 15791 15792 15793 15794 15795 15796 15797 15798 15799
15800 15801 15802 15803 15804 15805 15806 15807 15808 15809 15810 4697 15811 15812 15813 15814 15815 15816 15817
15818 15819 15820 5223 15821 15822 15823 15824 15825 15826 4841 15827 15828 15829 5711 15830 4878 15831 15832
15833 15834 15835 15836 15837 15838 15839 4875 15840 15841 15842 15843 15844 15845 15846 15847 15848 15849 15850
15851 15852 15853 15854 15855 15856 15857 15858 15859 15860 15861 15862 15863 4721 4807 15864 15865 15866 15867

15868 15869 15870 5099 5421 15871 15872 15873 15874 15875 15876 15877 15878 15879 15880 15881 15882 15883 15884
15885 4749 15886 15887 15888 15889 15890 15891 15892 15893 15894 15895 4682 15896 15897 15898 15899 15900 15901
15902 5853 15903 15904 5521 15905 15906 5770 15907 15908 15909 15910 15911 15912 15913 15914 15915 15916 15917
5565 15918 15919 15920 15921 15922 15923 15924 15925 15926 15927 15928 15929 15930 15931 5991 15932 15933 15934
15935 15936 15937 15938 15939 15940 5879 15941 15942 15943 15944 15945 15946 15947 15948 15949 15950 15951 15952
15953 15954 4621 15955 15956 15957 15958 4744 15959 15960 15961 5584 15962 15963 15964 15965 15966 15967 4788
15968 15969 5220 15970 15971 15972 15973 15974 15975 15976 15977 15978 15979 5415 5798 15980 15981 15982 15983
15984 15985 15986 15987 15988 15989 15990 15991 15992 15993 15994 15995 15996 15997 15998 5254 15999 16000 16001
16002 16003 16004 16005 16006 16007 16008 16009 16010 16011 16012 5029 16013 16014 16015 16016 16017 16018 4880
16019 5510 16020 16021 16022 16023 5914 16024 16025 5332 16026 5485 16027 16028 5456 16029 16030 5707 16031
16032 16033 16034 16035 16036 16037 16038 16039 16040 5317 16041 16042 16043 4966 16044 6068 16045 16046 16047
16048 16049 16050 16051 16052 16053 5102 16054 16055 16056 16057 16058 16059 16060 16061 16062 16063 16064 16065
16066 16067 16068 16069 16070 16071 16072 16073 16074 16075 16076 16077 5376 16078 16079 16080 16081 16082 16083
16084 16085 16086 16087 16088 5032 5305 16089 16090 16091 16092 16093 16094 16095 16096 16097 16098 16099 16100
5463 16101 16102 16103 16104 5228 4895 16105 16106 16107 16108 16109 16110 16111 16112 16113 16114 16115 16116
16117 16118 16119 16120 4997 16121 16122 4864 16123 16124 16125 16126 16127 16128 16129 5618 5467 6065 16130
16131 16132 16133 16134 16135 16136 16137 16138 16139 16140 16141 16142 16143 16144 16145 16146 16147 16148 16149
16150 16151 16152 16153 16154 16155 16156 16157 16158 16159 16160 16161 16162 16163 5480 16164 16165 16166 16167
16168 16169 16170 16171 16172 16173 16174 16175 5078 16176 16177 16178 16179 16180 16181 16182 16183 16184 16185
16186 5782 16187 6049 16188 16189 16190 16191 16192 5243 16193 16194 16195 16196 16197 4839 16198 16199 16200
16201 16202 16203 16204 16205 16206 16207 16208 16209 16210 16211 4596 16212 16213 16214 16215 16216 16217 16218
16219 16220 16221 16222 16223 5489 16224 16225 16226 5391 16227 16228 16229 16230 16231 16232 16233 16234 16235
16236 16237 16238 16239 16240 16241 16242 4574 5160 5340 16243 16244 16245 16246 16247 5458 16248 16249 5692
16250 16251 16252 16253 16254 16255 16256 16257 16258 16259 5543 16260 16261 5333 16262 16263 16264 16265 16266
16267 16268 16269 16270 5810 16271 16272 16273 16274 16275 16276 16277 16278 16279 16280 16281 16282 16283 16284
16285 16286 16287 16288 16289 16290 16291 16292 16293 16294 16295 16296 16297 16298 16299 4916 5433 5872 16300
16301 16302 16303 16304 16305 16306 16307 16308 16309 16310 16311 16312 16313 6053 16314 16315 6016 16316 16317
16318 16319 16320 16321 16322 5365 16323 16324 4978 16325 16326 16327 16328 16329 16330 16331 16332 5246 4790
16333 16334 5033 16335 16336 16337 16338 16339 16340 16341 5270 16342 16343 16344 16345 16346 16347 16348 16349
16350 16351 16352 5069 16353 5401 16354 16355 16356 5615 16357 16358 16359 16360 4977 16361 16362 16363 16364
16365 16366 16367 16368 16369 16370 16371 16372 16373 16374 16375 16376 16377 16378 16379 16380 16381 16382 16383
16384 5320 16385 16386 4733 16387 16388 5417 16389 16390 16391 16392 16393 16394 16395 16396 16397 16398 16399
16400 16401 16402 16403 16404 16405 16406 16407 4826 16408 16409 4616 16410 16411 16412 16413 16414 16415 16416
16417 16418 16419 16420 5294 16421 5977 16422 5796 16423 16424 16425 16426 16427 16428 16429 16430 16431 5469
16432 16433 16434 16435 16436 16437 16438 16439 16440 16441 16442 16443 16444 16445 16446 16447 16448 16449 16450
5832 16451 16452 16453 16454 16455 16456 16457 16458 16459 16460 16461 16462 16463 16464 16465 5669 16466 16467
16468 16469 16470 16471 16472 16473 6017 16474 16475 16476 16477 16478 5959 16479 16480 16481 16482 16483 16484
16485 16486 16487 16488 16489 5153 16490 16491 16492 16493 16494 16495 16496 4706 16497 16498 16499 16500 16501
5498 16502 16503 16504 16505 5238 16506 16507 16508 16509 16510 16511 16512 16513 16514 16515 16516 16517 16518
16519 16520 16521 16522 16523 16524 16525 16526 16527 16528 16529 16530 16531 16532 16533 6062 16534 16535 16536
16537 16538 5634 16539 16540 16541 16542 5314 16543 16544 16545 16546 16547 16548 5409 16549 16550 5944 4767
16551 16552 16553 16554 16555 5547 16556 16557 16558 16559 16560 4627 16561 16562 16563 5393 16564 16565 16566
16567 16568 16569 16570 16571 16572 16573 16574 16575 16576 16577 16578 16579 16580 16581 16582 16583 5766 5585
16584 16585 5384 5207 16586 16587 16588 16589 16590 16591 16592 16593 5090 16594 16595 16596 16597 5104 16598
16599 16600 16601 16602 16603 16604 16605 16606 16607 16608 16609 16610 16611 16612 16613 16614 16615 6018 16616
16617 16618 16619 16620 4848 16621 16622 16623 16624 16625 16626 16627 16628 16629 16630 16631 16632 16633 16634
16635 16636 16637 16638 16639 16640 16641 16642 16643 16644 16645 16646 16647 16648 5662 16649 16650 16651 16652
16653 16654 16655 16656 16657 16658 16659 16660 16661 16662 16663 16664 16665 16666 16667 16668 16669 4601 16670
16671 16672 16673 16674 16675 16676 16677 16678 16679 16680 16681 16682 16683 16684 16685 16686 16687 16688 16689
16690 16691 16692 16693 16694 5484 16695 5877 16696 16697 16698 16699 16700 16701 16702 16703 16704 16705 16706
16707 16708 16709 16710 16711 16712 16713 16714 16715 16716 16717 16718 5381 16719 16720 16721 16722 16723 16724
16725 16726 16727 16728 16729 16730 16731 16732 16733 6002 16734 16735 16736 16737 16738 16739 5568 16740 16741
16742 5221 16743 16744 16745 16746 16747 16748 16749 4660 16750 16751 16752 16753 16754 16755 16756 16757 16758
5997 16759 16760 16761 16762 16763 5633 16764 16765 6019 16766 16767 16768 5052 16769 16770 16771 16772 16773
16774 16775 16776 16777 16778 16779 16780 16781 16782 16783 16784 16785 16786 16787 16788 16789 16790 16791 16792
16793 16794 16795 16796 16797 16798 16799 16800 16801 16802 16803 16804 16805 16806 16807 16808 16809 16810 16811
16812 16813 16814 16815 16816 16817 16818 16819 16820 16821 16822 16823 5960 16824 16825 16826 16827 16828 16829
16830 6043 16831 16832 5855 16833 16834 16835 16836 16837 5134 16838 16839 16840 16841 16842 16843 16844 16845
4728 16846 16847 16848 16849 16850 16851 16852 16853 16854 16855 6103 16856 16857 16858 5138 16859 16860 5064
16861 16862 16863 16864 16865 16866 16867 16868 16869 16870 16871 4773 16872 16873 16874 16875 16876 16877 16878
6093 16879 16880 16881 16882 16883 16884 16885 16886 4811 16887 16888 16889 16890 16891 16892 16893 16894 16895
16896 16897 16898 16899 16900 16901 16902 16903 16904 16905 5321 6095 16906 16907 16908 16909 16910 16911 16912
16913 16914 16915 5366 16916 4799 16917 16918 16919 16920 16921 16922 16923 16924 16925 16926 5926 16927 16928
5986 16929 16930 16931 16932 6081 16933 16934 16935 16936 16937 16938 16939 16940 16941 16942 16943 16944 16945
16946 16947 16948 5382 16949 16950 16951 16952 16953 16954 16955 16956 16957 16958 16959 16960 16961 16962 16963
16964 16965 16966 16967 16968 16969 16970 16971 16972 16973 16974 16975 16976 16977 16978 16979 16980 16981 16982
16983 16984 16985 5930 16986 6030 16987 16988 16989 16990 16991 16992 16993 16994 5115 16995 16996 16997 16998
16999 17000 17001 17002 17003 17004 4838 17005 17006 17007 17008 17009 17010 17011 17012 17013 17014 17015 17016
17017 17018 17019 17020 17021 5614 17022 17023 17024 17025 17026 17027 17028 17029 17030 17031 17032 17033 17034
17035 17036 17037 17038 17039 17040 17041 5923 17042 17043 17044 17045 17046 17047 17048 17049 5205 17050 5418
17051 17052 5259 17053 17054 17055 17056 17057 17058 17059 17060 17061 17062 17063 17064 17065 17066 17067 17068
17069 17070 17071 17072 17073 17074 17075 17076 17077 17078 17079 17080 17081 17082 17083 17084 17085 17086 17087
17088 17089 17090 17091 17092 17093 17094 17095 17096 17097 17098 17099 4679 17100 17101 17102 17103 17104 17105
17106 17107 17108 17109 17110 17111 17112 17113 17114 17115 17116 17117 17118 17119 5354 5430 17120 17121 17122
17123 17124 17125 17126 17127 4711 17128 17129 17130 17131 17132 17133 17134 17135 17136 17137 17138 17139 17140

17141 17142 5272 17143 17144 17145 17146 17147 17148 17149 17150 17151 17152 17153 17154 17155 17156 17157 17158 17159 6013 17160 17161



(a) ***(b)***

(c)

Figure III.12. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,* ***(c)*** *Histogrammes de l'image chiffrée.*

III.4.1.2. Description de PosESecL2

Comme la clé générée par notre première alternative codant le premier niveau de sécurisation, est de taille variable d'une donnée à une autre et qui est égale à la taille de la donnée en terme d'éléments (caractères ou pixels), telle que : taille de la donnée = n éléments => taille de la clé générée = n nombres ; alors la résistibilité à l'attaque exhaustive est strictement dépendante de la taille de la donnée. C'est pourquoi, nous proposons, ici, une variante de PosESecL1 et que nous l'appelons PosESecL2 (Position based Evolutionary Secure Level 2).

L'évolution vers la solution optimale (donnée cryptée) par PosESecL2 est assurée par application des opérateurs de reproduction (croisement, mutation) et de sélection des meilleurs individus. Dans ce qui suit nous ne présentons que les phases du processus évolutionnaire faisant la différence entre les deux algorithmes PosESecL1 et PosESecL2. Il s'agit, principalement, des phases de codage et de reproduction. Toutefois, le même mécanisme est adopté pour créer la population initiale en permutant aléatoirement le chromosome codant l'image originale.

a. Codage

Dans le but d'augmenter le niveau de sécurité du premier algorithme décrit dans la section précédente, nous présentons, ici, le nouveau codage proposé donnant lieu à un nouvel algorithme de chiffrement : PosESecL2. Ce dernier opère sur le codage des différents éléments de la donnée à chiffrer : dans le cas d'une donnée texte et de même que pour PosESecL1, le codage utilisé est l'hexadécimal, toutefois, chaque quatre bits des 16 bits codant un élément sont considérés comme un gène. Pour une donnée image, chacune des

composantes R, V et B est considérée comme étant un gène. Ainsi, la taille de la nouvelle clé sera plus grande que celle générée par le premier algorithme. Elle est d'une taille quatre fois plus grande dans le cas de manipulation des données texte et de trois fois plus grande dans le cas de manipulation d'une donnée image. Le codage proposé est résumé à travers la figureIII.13.

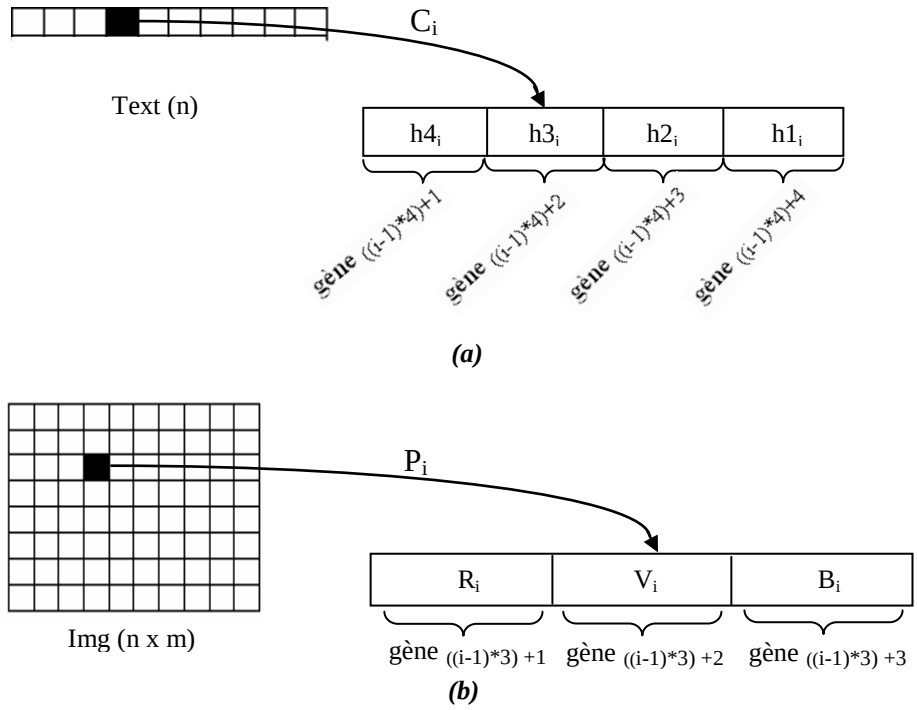


Figure III.13. **(*a*)** *Représentation chromosomale sous PosESecL2 d'un caractère C_i du texte Text(n),* **(*b*)** *Représentation chromosomale sous PosESecL2 d'un pixel P_i de l'image Img(n*m).*

Avec : ($h4_i$ $h3_i$ $h2_i$ $h1_i$) représente le codage Unicode hexadécimal du caractère C_i.

b. Reproduction

Contrairement à l'algorithme PosESecL1, l'algorithme PosESecL2 considère le codage des éléments d'une donnée indépendamment. Ainsi, l'application des opérateurs de croisement et de mutation ne nécessite pas la prise en compte des contraintes à vérifier comme dans le cas de PosESecL1, mais il enrichit d'avantage la population d'individus par de nouvelles caractéristiques.

De même que pour le PosESecL1, l'opérateur de croisement utilisé est le OX « Order Cross-over » appliqué avec un taux fixé expérimentalement et compris entre 60% et 100%. Toutefois, l'opérateur de mutation utilisé est une permutation multipoints pour éviter une convergence trop lente vu la grande différence des tailles des chromosomes dans PosESecL1 et PosESecL2 (le deuxième est d'une taille huit ou trois fois plus grande que le premier). Cet opérateur est appliqué avec un taux, aussi fixé expérimentalement et compris entre 0.1% et 5%.

c. Evaluation des individus

Le choix des individus qui survivront d'une génération à une autre s'effectue selon la fonction objective F, donnée ci-dessous, représentative de l'efficacité des solutions générées sur le problème posé.

$$F(I_i) = \sum_{j=1}^{n \times l} \left| G_{ji} - G_{j0} \right| \quad \text{(III.12)}$$

Avec : G_{ji} est la valeur du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.
l est la taille du codage d'un seul élément de la donnée à chiffrer.

Ainsi, la fonction d'évaluation prendra les deux instanciations suivantes dans le cas de chiffrement de données texte et celui de chiffrement de données images respectivement :

$$F(I_i) = \sum_{j=1}^{n \times 4} \left| G_{ji} - G_{j0} \right| \quad \text{(III.13)}$$

$$F(I_i) = \sum_{j=1}^{n \times 3} \left| G_{ji} - G_{j0} \right| \quad \text{(III.14)}$$

d. Critère d'arrêt

La condition d'arrêt assurant la convergence de l'algorithme PosESecL2 est la même que celle de PosESecL1 puisque le codage des individus dans les deux algorithmes ne diffère que sur le plan contenu de gènes mais les mêmes informations sont présentes dans les deux codages (codage Unicode des caractères du texte à chiffrer ou représentation RVB des pixels de l'image à chiffrer). Autrement dit, c'est la manipulation du codage des éléments de la donnée à chiffrer qui fait la différence entre les deux codages utilisés par les deux algorithmes proposés. Donc, la condition d'arrêt adoptée par ESecL2 dans le cas de chiffrement de données texte et celui de chiffrement de données images, respectivement, est la suivante :

$$0 \le F(I_i) \le F \times 4 \times n \quad / (F)_{16} = (15)_{10} \quad \text{(III.15)}$$

$$0 \le F(I_i) \le 255 \times 3 \times n \quad \text{(III.16)}$$

De même que pour PosESecL1, cette unique condition n'assure pas dans tous les cas la convergence de PosESecL2. C'est pourquoi un nombre maximal de générations sera fixé pour résoudre le problème.

e. Déchiffrement

De même que pour PosESecL1, ce deuxième algorithme utilise le même mécanisme de calcule de clé de chiffrement ; elle change d'un chiffrement à un autre, donc, elle dépend de la donnée à chiffrer et de sa taille. Ainsi, même la taille de la clé de session générée varie d'une donnée à une autre.

f. Réglage des paramètres et résultats

Dans cette section nous présentons les résultats de l'application du deuxième algorithme développé opérant suivant le paramétrage reporté à travers le tableau III.3. À partir des mêmes données originales précédemment utilisées pour tester PosESecL1, nous avons appliqué notre deuxième algorithme PosESecL2 afin d'obtenir les données chiffrées correspondantes. Ces dernières sont celles faisant l'objet des figures III.18.a, III.19.a, III.20.a et III.21.a.

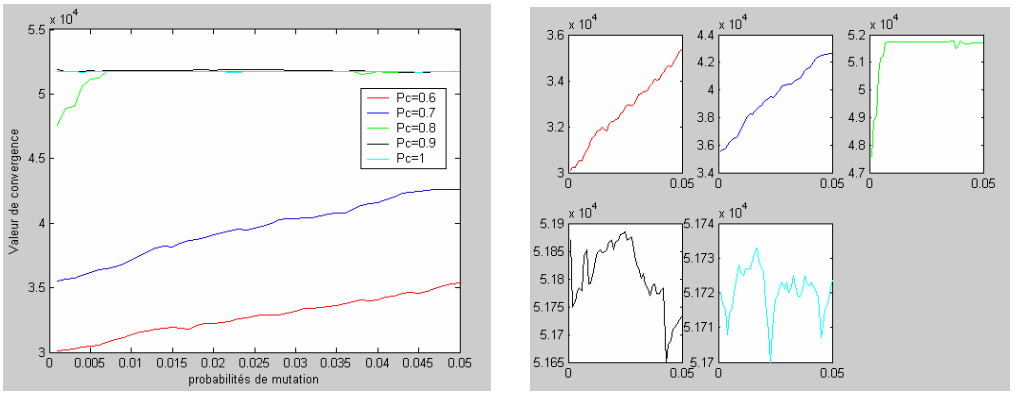


Figure III.14. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

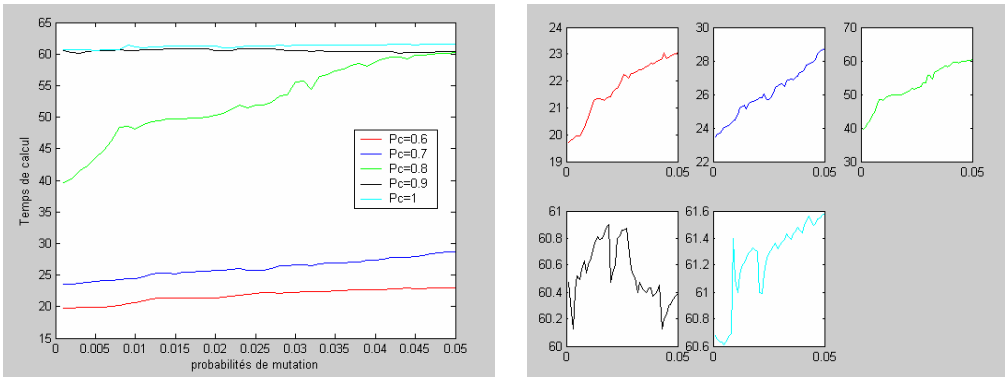


Figure III.15. *Influence des paramètres Pc et Pm sur le temps de calcul.*

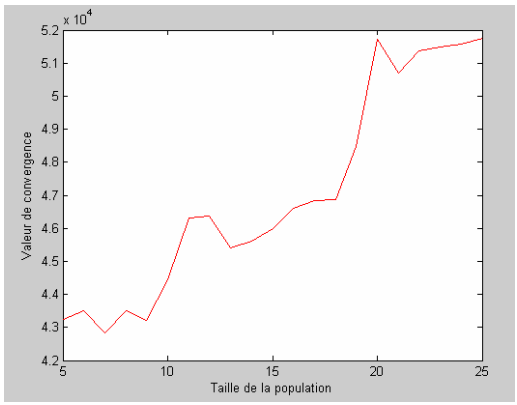


Figure III.16. *Evolution des valeurs de convergence en fonction de la taille de population.*

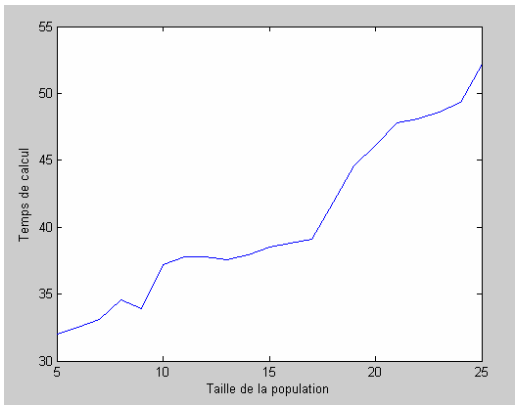


Figure III.17. *Evolution du temps d'exécution en fonction de la taille de population.*

	Valeurs des paramètres de PosESecL2
Taille de la population	20
Nombre de générations	100
P_c	0.8
P_m	0.007

Tableau III.3. *Valeurs adoptées pour les paramètres de PosESecL2.*

L'application de ce deuxième algorithme suivant le paramétrage reporté dans le tableau III.3 sur les mêmes données tests précédentes (Texte 1, Texte 2, Lena et Logo) a permis d'obtenir les résultats résumés à travers le tableau III.4.

D'après ces résultats, il est clair que la taille de la clé augmente proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cette dernière influence la taille des chromosomes résultants du codage utilisé. Donc, c'est la taille des chromosomes augmentant en fonction de la taille des données à chiffrer qui conduit à une augmentation de la taille des clés générées et du temps de calcul correspondant.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	56368	30729	28.74
	Texte 2	1151	59852	35100	31.5
Données images	Lena	131 X 131	823728	51720	46.15
	Logo	420 X 395	9456300	368712	73.06

Tableau III.4. *Résultats obtenus par PosESecL2.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

´1à1њњ╬‹à+1وî¨à+#cЖéZ0¨£1#§£î¨îغوàî+6+¨1وcњ♀њ{à0é6¨à+ض¨éغ+‹0++qو61ȅc¨00c¨î{وî8à‹9¨+њZtt01{î1£#à˚0î♀њ1à+╬0î£01↔
↔¨àaîéë¨+#↔↔¨1´éà0ààà+§8¨£§¨¨0´£è#§é1ë+♀++£11¨{éˀ↔1£ëàאַאַ¨++§+à+¨++9§t´£09£و╬+ëtˀé#éà0غњ↔6àא18وא##ë£¨+0
+¨00‹ˀ+و+9}1¨˚+8ˀc8££↔à+11++§#‹îخوZ0ˀ#ووëé{§1‹+9é86£009وZغ¨cˀ}18+î¨cëîé+§א0‹0+غو§£§îњ1˚0+1c+אوàوà+Z#q£18£אخغ♀و06#
î1¨غs‹1غ´î#é0#њZ╬§£é↔è+1+غ8و£+ëà9{њњњ¨éو¨#و£àt¨é+++00é˚#î1+î#6à‹0#0##àé11+ˀ‹é#אЖZ0#+غ££+↔Ж+à1خén£¨¨à#8}وé↔+¨î
╬90+و+Жë¨î‹§1++↔ëéאˀ+î£àé#ëZ11א§#א♀à#î£+1£é+1ët↔c¨خ+ˀëאcوîé#îéˀ#ˀî1#˚î¨+8#+1+و£éوé╬¨ˀ1£وZZc8њغ#é10خà#+↔08¨╬1#+
£é0té£+88+#^وא9‹î¨î+א¨{˚£§£+´6§6╬╬↔+¨њњњњà{6+##و↔↔ëˀ+îà+16îњ1¨11£1£0#9+غà§+é£+ë+#11ˀ£ëî‹î+£‹1+´£00£+غ¨+وé#£19
♀♀c§î+c01àî¨8#8و+╬à§¨{î¨+60¨§6♀0Z8é#1#à{+1£‹à9#1ˀà6+¨1א£˚6à+1î1ë+ˀt9£+îî+80£6é#¨+£¨ZZ1c¨њњ{ëc0♀t0#وtو}+0è↔{ècc8î#
£¨£#وàà£#+1£+њЖњ╬à˚116↔1ˀ+éîZà£c{´+#

(b)

Figure III.18. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($\text{Taille}_{\text{Clé}}$ = 6,8808 K octets) :

1	450	496	406	440	458	465	500	437	477	491	298	487	444	468	497	441	471	493
494	454	102	225	455	453	484	473	330	429	432	428	447	469	480	436	488	475	74
446	478	490	443	430	486	483	481	466	499	239	449	489	456	120	464	435	498	451
442	479	474	412	434	467	336	485	5	457	101	213	448	470	431	460	463	438	307
476	445	461	427	459	462	290	495	482	472	433	157	439	426	452	740	1330	545	1331
508	1332	975	1333	564	1334	1335	961	818	1336	886	812	1337	908	674	875	1054	1338	808

725	1339	1340	1341	1342	1343	1344	786	732	1226	1297	228	635	1345	1346	863	696	758	1347
689	1348	1349	1350	710	126	855	548	756	667	522	892	604	1351	1352	602	1353	146	920
770	691	678	729	901	1354	876	1025	1355	4	3	6816	609	993	1356	1357	1229	1358	309
1359	1360	502	1361	706	1362	1363	1364	1365	880	911	644	784	1366	773	359	768	1367	811
299	940	578	198	563	1368	149	957	742	668	687	833	738	1369	623	878	852	636	1370
634	253	1371	873	586	1151	836	620	1372	948	1004	928	1373	799	882	830	270	684	847
964	550	759	1374	870	937	1375	765	905	514	1376	528	739	583	959	337	1377	693	752
612	796	540	1378	552	680	1379	510	946	216	637	630	992	127	978	1309	643	1380	516
1381	559	889	1382	797	567	1383	645	1384	820	949	844	747	1385	866	1386	1387	1160	848
652	384	1388	560	976	985	603	1003	942	501	1389	633	841	974	1093	1390	657	39	912
1391	982	1392	1393	1394	699	750	894	794	1395	1396	932	864	1397	987	716	807	509	676
708	505	1398	838	1399	231	1400	315	339	523	556	326	673	1020	737	599	577	791	751
1329	613	898	1401	1402	419	1001	1403	1404	900	1405	1406	492	1407	536	766	1408	1409	702
787	1410	187	924	1411	845	793	1412	1138	804	1413	1414	1415	546	731	1416	18	1417	66
1418	650	511	662	1288	971	715	817	962	694	1419	968	631	966	537	654	364	520	1104
850	1193	150	981	782	1420	543	915	827	1421	995	1422	843	542	934	1423	1424	775	517
642	709	9	1425	607	772	529	80	994	842	819	712	639	656	2	989	1426	938	610
851	735	666	832	651	541	600	227	241	884	781	895	175	515	417	1237	525	825	734
965	278	890	138	1427	1428	1429	596	1430	553	1293	923	589	730	1431	555	917	1261	663
1432	1433	1000	58	130	1434	790	672	1435	534	1436	1437	1169	44	805	641	800	1438	958
1291	954	1439	1440	1441	1442	835	1443	640	595	1444	933	717	700	888	1445	538	601	616
1446	675	617	701	972	1447	757	909	1448	918	931	1171	704	660	573	998	1449	871	771
1450	763	906	1451	185	1116	557	1452	1453	711	403	916	983	1454	769	1455	221	979	713
1456	846	903	813	839	743	558	910	1295	585	1457	597	1458	1459	1460	967	1461	1462	310
362	313	697	1463	1464	935	719	16	105	856	135	921	952	726	896	1240	1465	705	927
930	1466	397	1467	720	647	980	1468	829	575	950	1469	580	883	664	527	746	323	780
561	614	877	611	828	776	1470	626	1471	570	681	688	569	622	618	879	1472	810	973
572	1473	314	1474	615	683	1475	1476	108	627	1264	665	343	714	382	52	801	1287	152
1477	521	579	28	1478	1479	721	532	1480	1481	549	951	874	46	91	744	592	284	1210
254	1482	206	1483	606	646	945	872	767	320	774	55	348	1484	593	698	535	1485	659
1486	1155	1487	1488	1489	670	1490	990	1491	240	518	815	1492	653	1493	754	899	857	755
798	1494	887	861	1495	143	897	544	996	503	1496	1497	1243	1498	1499	824	1500	1501	1502
533	551	840	977	1503	837	590	1504	988	1505	1506	1507	565	963	788	727	777	1508	1509
809	1510	109	1511	174	1512	869	547	1234	986	904	991	638	1513	999	598	1514	566	722
748	1515	513	802	524	96	1516	1517	1518	919	245	679	507	255	690	1519	733	506	939
922	685	860	1520	853	865	628	504	1521	1522	1523	941	1037	1524	686	1525	355	1526	677
1527	1528	295	914	1189	834	749	661	718	581	1529	764	1530	1531	785	280	823	854	1532
1533	244	859	658	519	648	649	1534	831	984	792	608	554	122	1535	1536	1537	955	723
1538	947	1539	997	1540	568	1541	582	1031	1542	1543	692	1207	588	821	956	671	418	1544
1545	760	1546	943	1547	1548	902	1549	703	1550	1289	695	574	728	745	41	594	970	587
1551	1552	233	1553	1554	655	1555	1556	421	1557	1558	893	562	803	7	814	1559	891	1560
1142	512	1561	629	1562	311	936	73	530	1563	1564	1565	783	1566	1567	1568	137	881	669
624	926	1569	741	415	736	753	953	302	1073	318	1570	779	1571	162	1572	1573	867	266
1574	605	1064	1575	334	90	12	1200	576	1576	1577	969	762	907	1038	619	862	1272	1578
1579	571	822	526	1580	45	49	913	795	1315	761	960	849	944	95	625	1581	1582	201
1583	1005	929	1584	1585	632	531	925	591	1586	789	868	300	682	1587	826	584	885	707
539	1002	806	621	778	724	858	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
1600	1159	1276	1601	1103	1602	1603	1604	1605	1606	1607	128	1608	1609	1610	1611	1612	1613	1614
34	1615	1328	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
1632	1317	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649
1062	1650	1651	1652	1653	1654	1655	1656	1657	1246	1658	1659	1660	1661	1662	1663	1664	1132	1665
1666	1667	349	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683
1684	199	1685	1083	1686	1687	1688	1689	1690	1691	1692	1211	166	1693	1694	1695	1696	1697	291
1247	1153	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1084	1709	1710	1711	140	1712
1713	1714	196	1715	1716	141	1717	1718	1719	1720	1721	1722	1723	1724	1327	287	1725	1726	1009
1727	1728	1729	1730	235	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744
1745	1746	1747	1748	1749	1750	1167	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762
1763	1764	1765	383	57	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779
1780	1144	1781	1185	1782	1783	1784	1294	1785	1786	1324	1787	1156	1788	1789	1790	1791	1792	1793
1195	1281	1794	1795	1796	1797	1798	1799	1800	1014	1801	1802	246	1803	1804	1805	1806	1145	1807
1808	1809	1810	1811	79	1812	1813	1814	1050	1274	286	1815	1816	1817	1818	1819	1820	1821	1822
1823	1824	1825	374	1186	1826	1827	395	1828	1112	1829	1830	112	1831	1832	82	1833	1834	1835
1836	1837	1838	1839	1840	1091	1841	1842	1843	1844	1845	1846	1847	1848	269	1849	1850	17	1275
1851	1177	204	1852	1853	1854	1855	1856	1857	1858	1859	1860	1216	1861	1862	1863	1864	1865	142
1866	1867	1090	1868	1869	1870	56	27	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1039
1881	1882	1883	1884	1075	1885	1886	154	1887	1888	1115	1889	1063	1890	1891	1892	208	1893	1894
1895	1896	1897	1898	1899	1900	1901	1902	1903	1107	1904	1905	1906	1907	1908	1909	1910	1911	1912
1913	1914	1915	1916	125	1917	1918	169	1919	1920	1921	1922	1923	1924	1925	1926	1927	398	1928
1929	1930	1931	1267	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	352	1943	1051	1944
1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	8	1957	1958	1959	1960	1961	1962
293	1963	1964	1965	1966	1967	1968	53	1969	1970	1296	1971	1015	1972	1973	1974	1975	1976	1977
1978	1979	1074	1980	1279	1981	312	1072	1982	1983	1984	1985	1986	264	1987	1988	1989	265	1990
1991	1992	1993	1994	1995	1996	1997	1998	1999	1017	2000	2001	2002	252	2003	2004	2005	342	2006
2007	2008	2009	2010	2011	1130	2012	2013	2014	2015	2016	2017	2018	2019	2020	1041	2021	2022	2023
1058	133	2024	2025	2026	2027	2028	2029	2030	2031	2032	1184	2033	2034	2035	2036	2037	1187	2038
2039	1136	2040	381	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055

1304	305	410	2056	2057	2058	2059	2060	10	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070
2071	2072	2073	2074	35	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088
2089	2090	2091	283	2092	2093	2094	2095	2096	2097	2098	1069	2099	2100	2101	170	2102	2103	2104
2105	2106	2107	2108	2109	2110	1111	2111	297	226	2112	2113	64	2114	2115	267	2116	33	2117
2118	2119	2120	2121	2122	2123	2124	288	2125	2126	2127	2128	2129	2130	2131	176	2132	2133	2134
2135	2136	2137	214	2138	2139	2140	2141	2142	2143	2144	1179	356	2145	2146	2147	2148	2149	2150
2151	2152	2153	2154	2155	2156	2157	2158	2159	99	2160	2161	2162	2163	2164	2165	139	2166	2167
2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	89	2182	2183	2184	153
2185	2186	2187	54	2188	2189	2190	2191	2192	2193	2194	186	2195	1125	2196	2197	2198	2199	2200
1057	2201	2202	2203	2204	2205	261	2206	1158	2207	2208	2209	2210	2211	1087	2212	2213	2214	2215
1273	2216	2217	2218	2219	2220	2221	2222	2223	1230	2224	2225	2226	2227	2228	2229	229	2230	2231
2232	2233	2234	2235	2236	2237	2238	2239	2240	86	2241	2242	1266	2243	2244	103	2245	2246	2247
2248	2249	2250	2251	1235	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	1231	1137	2262	2263
2264	2265	2266	2267	2268	219	2269	2270	376	2271	2272	2273	2274	2275	2276	2277	1113	2278	2279
2280	2281	304	2282	2283	2284	2285	2286	72	2287	2288	2289	2290	2291	394	2292	2293	2294	24
2295	2296	2297	2298	1258	165	2299	1259	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
107	2311	2312	2313	2314	2315	1110	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327
2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	136
2346	1008	1024	2347	2348	2349	2350	2351	2352	2353	404	2354	1232	2355	2356	2357	2358	2359	2360
2361	2362	1102	2363	2364	2365	2366	371	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377
2378	2379	2380	2381	2382	2383	1217	1046	2384	2385	2386	2387	2388	2389	2390	2391	2392	123	2393
2394	2395	379	2396	2397	2398	2399	2400	1203	2401	274	1223	1071	2402	388	2403	2404	2405	1081
2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	1019	2416	2417	2418	2419	2420	2421	2422	2423
2424	2425	2426	2427	2428	2429	2430	2431	2432	40	2433	2434	2435	2436	2437	2438	2439	2440	1162
2441	2442	2443	1109	2444	306	2445	1325	2446	2447	2448	2449	2450	2451	2452	2453	2454	134	2455
2456	2457	76	2458	2459	2460	43	2461	2462	2463	2464	2465	129	2466	2467	2468	2469	2470	2471
399	191	2472	2473	2474	2475	2476	2477	405	2478	2479	2480	2481	2482	2483	2484	2485	2486	2487
2488	294	2489	2490	2491	2492	2493	2494	2495	2496	177	2497	2498	276	247	2499	2500	256	2501
2502	2503	2504	2505	2506	2507	389	2508	2509	2510	2511	2512	2513	2514	2515	2516	1012	346	2517
2518	350	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535
2536	2537	2538	2539	2540	2541	2542	2543	1222	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553
1206	2554	2555	2556	369	2557	2558	2559	292	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569
2570	2571	2572	2573	2574	1028	2575	1205	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586
104	2587	2588	2589	2590	2591	2592	2593	2594	328	2595	2596	1120	2597	2598	2599	2600	2601	2602
2603	2604	2605	2606	2607	316	236	2608	2609	1286	2610	2611	2612	2613	2614	2615	2616	2617	2618
411	2619	2620	2621	2622	2623	2624	2625	2626	1016	2627	365	2628	1214	2629	2630	2631	2632	2633
2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652
2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	19	2664	2665	2666	2667	2668	2669	2670
2671	1176	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688
2689	366	2690	2691	2692	2693	2694	2695	2696	2697	2698	271	2699	172	2700	1114	2701	2702	2703
2704	2705	193	2706	2707	2708	2709	205	2710	2711	1122	2712	2713	2714	2715	2716	1248	2717	2718
1150	2719	2720	2721	2722	2723	2724	189	373	2725	2726	2727	2728	2729	1183	2730	2731	2732	2733
2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	1182	2748	2749	2750	1199
2751	2752	1036	2753	420	2754	2755	145	1148	2756	2757	2758	2759	2760	1194	2761	2762	2763	1106
2764	2765	2766	115	2767	2768	368	2769	2770	163	2771	2772	2773	2774	2775	390	2776	2777	2778
2779	111	113	2780	248	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794
2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	173
2813	2814	2815	168	2816	2817	2818	2819	1061	2820	2821	1249	2822	1094	2823	159	2824	2825	2826
2827	2828	2829	2830	2831	2832	2833	2834	2835	1146	2836	1022	2837	2838	2839	1027	2840	2841	2842
48	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	360	2857	2858	2859
2860	2861	2862	2863	2864	1209	22	178	2865	2866	1181	2867	2868	2869	2870	1085	32	14	2871
1174	2872	2873	2874	2875	2876	1307	1134	1096	2877	1218	2878	2879	2880	2881	2882	2883	2884	2885
2886	329	2887	2888	144	2889	2890	2891	2892	2893	2894	2895	2896	1066	2897	2898	2899	2900	2901
2902	2903	2904	1244	409	2905	400	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917
2918	2919	1170	2920	2921	2922	132	42	2923	2924	2925	2926	2927	2928	2929	2930	77	2931	2932
211	1143	1121	242	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943	1192	2944	2945	422
2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	87	2961	2962	2963
2964	2965	2966	2967	1040	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	322	2978	2979	2980
2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999
3000	3001	1013	3002	3003	37	3004	3005	3006	3007	3008	1300	3009	3010	1250	3011	3012	3013	257
3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029	1166	3030	3031
3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	1262	3043	3044	1313	1149	372	1078	3045
3046	3047	3048	3049	3050	3051	3052	1215	3053	3054	3055	3056	3057	3058	3059	3060	3061	1265	3062
3063	3064	181	3065	3066	3067	268	3068	3069	3070	3071	182	3072	3073	3074	3075	3076	3077	3078
401	3079	3080	3081	3082	161	3083	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	1178
3095	1268	3096	3097	3098	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	1055	3110	3111
3112	3113	3114	1256	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	1204	3125	3126	3127	3128
3129	3130	3131	3132	3133	3134	3135	3136	3137	3138	50	3139	3140	1131	3141	3142	3143	3144	3145
3146	3147	3148	3149	151	3150	3151	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	1320
3163	317	1070	327	3164	3165	1242	3166	3167	1299	402	3168	3169	367	3170	3171	3172	3173	3174
3175	3176	3177	3178	131	3179	94	3180	3181	3182	1154	3183	1047	3184	3185	3186	1168	3187	3188
363	3189	3190	3191	3192	3193	3194	258	3195	3196	121	3197	3198	93	3199	319	3200	1099	3201
3202	3203	202	3204	3205	354	416	3206	3207	3208	3209	3210	3211	1316	3212	3213	3214	3215	3216
3217	3218	3219	3220	3221	303	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	243	3233
1269	3234	1233	3235	3236	3237	85	3238	3239	3240	3241	3242	3243	1319	1213	3244	3245	3246	3247
3248	218	1225	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	1048	3259	3260	119	3261	3262

3263 3264 3265 3266 3267 3268 3269 238 3270 3271 3272 3273 3274 3275 3276 223 3277 1032 3278
3279 1101 3280 3281 3282 3283 1076 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 1260 3294
3295 3296 3297 3298 3299 3300 279 3301 3302 3303 3304 3305 118 3306 3307 3308 1129 3309 3310
3311 97 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328
3329 1135 3330 3331 3332 3333 184 3334 370 3335 3336 224 3337 3338 3339 1191 3340 3341 3342
3343 110 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 1117
1254 3360 3361 3362 3363 3364 3365 188 3366 3367 3368 1157 1147 3369 155 3370 3371 3372 1282
21 3373 3374 3375 3376 1241 3377 3378 425 3379 3380 3381 3382 3383 1190 3384 3385 1033 3386
3387 3388 3389 3390 3391 413 3392 3393 1065 3394 3395 285 1228 3396 3397 3398 3399 3400 3401
3402 3403 3404 194 3405 3406 3407 321 230 1098 3408 3409 3410 3411 3412 3413 197 3414 3415
3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 1251 3428 3429 3430 3431 3432 3433
1152 3434 3435 353 1030 3436 1292 3437 3438 215 1180 3439 3440 3441 3442 3443 3444 3445 3446
3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 164 3457 3458 3459 361 3460 3461 3462 3463
3464 3465 3466 3467 25 3468 3469 331 3470 3471 3472 3473 3474 3475 1270 3476 195 3477 3478
3479 3480 3481 3482 3483 3484 3485 3486 20 340 3487 3488 3489 3490 84 3491 3492 3493 3494
3495 3496 1290 209 1108 1006 3497 3498 3499 3500 3501 190 3502 3503 3504 3505 3506 1018 3507
3508 3509 3510 1224 1053 1124 207 3511 212 3512 3513 3514 3515 1285 114 3516 3517 3518 3519
324 3520 3521 3522 3523 3524 3525 3526 1221 3527 3528 158 3529 3530 3531 3532 3533 1236 3534
335 3535 3536 1172 3537 3538 3539 3540 3541 3542 1035 3543 1318 1119 1220 1011 3544 3545 3546
3547 3548 3549 3550 3551 3552 3553 3554 424 3555 3556 387 3557 3558 3559 3560 3561 3562 3563
3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 156 3574 3575 3576 3577 3578 3579 3580 3581
3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 1118 3592 1303 3593 3594 3595 3596 31 263
3597 3598 3599 3600 3601 3602 3603 69 3604 3605 3606 3607 3608 3609 1133 3610 3611 3612 3613
1196 3614 3615 3616 1219 1173 3617 396 1188 3618 3619 3620 3621 282 3622 3623 3624 3625 3626
3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 1239 1198 3637 3638 3639 408 3640 3641 3642
1049 1089 3643 3644 1306 3645 3646 3647 3648 3649 296 3650 3651 3652 1271 3653 3654 3655 160
3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674
3675 3676 3677 3678 3679 3680 1305 3681 3682 3683 1284 3684 3685 3686 3687 3688 3689 3690 3691
3692 3693 3694 3695 3696 3697 3698 15 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 217
3709 3710 1127 3711 1298 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725
1140 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 1067 3738 3739 3740 3741 180
3742 3743 3744 1023 3745 3746 3747 3748 3749 3750 3751 3752 1161 3753 332 3754 3755 3756 3757
1252 3758 3759 3760 3761 3762 171 3763 3764 3765 3766 3767 3768 100 3769 3770 3771 3772 3773
3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 192 3787 3788 3789 3790 3791
3792 3793 3794 3795 3796 3797 262 1082 333 3798 3799 124 3800 3801 3802 1312 1052 3803 3804
3805 3806 391 3807 3808 1278 3809 1042 83 3810 347 3811 3812 3813 70 3814 3815 3816 3817
3818 3819 3820 3821 3822 3823 3824 3825 1126 3826 3827 3828 3829 3830 3831 3832 3833 3834 98
3835 51 81 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 1255 3846 3847 67 3848 3849
3850 3851 3852 3853 3854 3855 1208 3856 3857 30 3858 3859 3860 3861 65 301 47 1021 3862
3863 3864 3865 3866 3867 3868 29 3869 3870 3871 3872 3873 3874 3875 1253 3876 3877 3878 3879
1321 3880 3881 3882 3883 63 3884 71 3885 3886 3887 3888 275 1175 3889 3890 3891 3892 3893
3894 3895 1045 3896 3897 3898 3899 423 260 3900 3901 3902 308 3903 3904 3905 3906 26 59
3907 3908 3909 1322 3910 3911 3912 3913 3914 3915 1197 1308 3916 3917 3918 3919 3920 3921 3922
3923 3924 1086 3925 3926 3927 3928 1263 68 1128 272 3929 3930 3931 3932 3933 167 3934 3935
3936 3937 3938 3939 3940 3941 3942 147 3943 1034 3944 3945 3946 3947 3948 3949 3950 3951 3952
3953 3954 3955 3956 3957 3958 3959 3960 357 3961 3962 3963 3964 3965 3966 3967 1165 1277 3968
38 3969 3970 232 3971 3972 3973 3974 3975 277 3976 3977 3978 3979 3980 3981 3982 3983 3984
3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003
13 4004 4005 4006 1202 4007 4008 4009 4010 4011 4012 4013 4014 4015 1314 4016 4017 4018 1326
4019 4020 4021 1141 4022 250 4023 4024 4025 4026 4027 4028 4029 4030 1201 4031 1227 4032 386
4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 1095 179 4043 4044 4045 1080 4046 4047 4048
4049 75 1029 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 237 4060 4061 4062 4063 4064
4065 4066 203 4067 4068 1301 414 249 4069 4070 380 4071 4072 1283 4073 4074 4075 78 4076
378 4077 61 4078 4079 4080 4081 4082 4083 4084 88 4085 4086 325 4087 1044 4088 4089 4090
200 4091 377 4092 4093 1164 4094 4095 4096 4097 222 4098 4099 4100 4101 4102 1059 4103 4104
4105 4106 4107 1026 4108 4109 1100 4110 358 4111 4112 1238 4113 1139 4114 4115 4116 4117 4118
4119 4120 1060 1302 4121 4122 4123 4124 4125 4126 4127 344 148 4128 4129 4130 4131 1092 4132
4133 4134 4135 4136 4137 4138 4139 4140 4141 1310 273 4142 4143 4144 4145 4146 4147 4148 4149
4150 4151 1163 116 4152 4153 4154 4155 259 251 4156 4157 4158 4159 4160 4161 4162 407 4163
4164 375 60 4165 4166 4167 1043 4168 4169 4170 4171 36 4172 4173 4174 1105 385 4175 4176
4177 4178 4179 4180 4181 4182 4183 220 4184 1077 4185 4186 4187 1007 4188 4189 4190 106 4191
11 4192 4193 1280 4194 4195 4196 4197 4198 4199 4200 289 92 4201 4202 4203 4204 4205 4206
1311 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223 4224
4225 4226 1097 1123 4227 4228 4229 4230 62 4231 351 4232 4233 4234 4235 4236 4237 4238 4239
4240 1257 4241 4242 4243 4244 4245 4246 4247 4248 1056 4249 183 4250 1245 4251 4252 4253 4254
4255 4256 4257 338 4258 4259 4260 4261 117 341 4262 4263 4264 1079 4265 4266 4267 4268 4269
4270 4271 4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282 4283 4284 4285 4286 4287 4288
4289 4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303 4304 4305 4306 4307
4308 345 4309 4310 234 4311 4312 281 4313 4314 1212 1088 4315 4316 1323 4317 4318 4319 392
210 4320 4321 4322 4323 4324 393 4325 4326 4327 4328 4329 4330 1010 23 4331 4332 4333 4334
1068 4335 4336

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملا أساسيا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردى بعدما كان الأمر مقصوراً فى الماضى على المراسلات الرسمية أو السرية بطبيعة الحال.

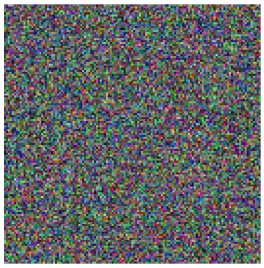
(a)

8ʊ8▒8}¢j^∂ز╔╔ש╔╔j8♂♂®±ĀʊĀ¦88¦(▒}÷╔♂}j¢∂àא`j0ıoز®שʊʊâ♠â&∑▒÷(8♠┤÷┤j∂
(&g8}¦`¦8j(ë÷Ж÷Жjv`♂8±┤¢Æ8^){&ıoıo88j&שÓÖ¦ʊ¢א3ز÷`!ë}ıo¦(ز∑┤¦÷÷3j8®Ā&#j8∂ë8â}╔÷8#8ʊʊʊıoâ:}jj8(&שıoıo8┤j╔8®ĀĀ3¢±
88ë(¦àjא∂}:8¢¢jj¦ç83÷8`ز&∂0&ز®&j÷8▒â┤}Ö&}¢}!j┬jâ3¦¦#`8┤jÖ&Ā8¢â¦±¢j3♠8}ʊ♠Ā3¢`¦ë3±▒▒▒ت&÷j¢ے 8!8┤
∂Æ3÷}jj}¦¢¦&¦8♠♠♠j3ت¦ز¦ıo¦¢8∂8jت÷ʊâ÷^â¢{ز:8÷¢ز∂8¢±¢ıoıoıo╔╔8!¢¦8¢â♂∑∂∑+_¦&زÆ ے {÷`&
8jj(&▒Æ╔╔╔â8┤א8Ā&א÷^±3∑8}{¢ت Ж®¢8¦&♠&¦ت▒0א8±ש∑∑Ā&(&}3¦â3¦®8&â&(÷àj¦(¦ے
ëâ_8ززjj&88&88ت&àj(¦±╔+!}÷ززت÷} Ж&&}ʊجë:8÷╔Ö8!¢ ש}8±8Öے ت╔8{8&8*!ت: { ʊز±&
ے®3ëâ3زشززjʊא0}∂¢j÷&gıoıo¢3┤♠}3♠j8÷âj88زg0¢¢∂ıo∂¦â?ت÷┤▒8÷&j♠+Æ8¢^±ت(`jıo^▒א ▒ز (àت Ā3: ıo&8±ש¢`3ژ88┤ ے ıo3Ж8ے
8:8}±}8زj`88#┤¦&j¦88: Ö3ز8¢¦}∑8&▒زj&8ЖК (¢&ز®
(j88ژ¦ت÷ف}ë÷&8÷&♂¢88ززjh&Æ&÷8¦3&¦88¢8!{¢)¢&8÷ز±±}^!âj«!â¦à8&&!8ز
♠┤&▒8{j8j±¦ש¦j8à÷:8®8ز±∑3j&3ששש8±┤}j}j÷÷+i#∂j¦(3¦ز3┤àĀЖ}3! ש±÷â83&â0¦â3`0&ت▒ے ز ë83شàא∂ıoא∂∂3&8┤Æے
Æ8^â♠^3}&â¦┤┤ز^Öj¦â888ے
â}÷&}&88÷ز¢ز8ز*:zj3¢}j¢زא8&jزıoj}8¢¢ز♠♠8â&┤+3¢8¢ت:âjژ8{+┤`âà÷#8¦`}ز÷âЖ&j:`jà÷83∑Æj÷÷j`â÷}÷*Öj30₪8÷8زà*8¢┤÷ز±÷
3÷¢8«8±┤&j8¦8â÷8╔8¢j88±8ز♠8∂ز3j&♠¦¢ë╔┤÷زâ388àzj¦¦0ت®®÷jâ┤}÷«ɜ3â8}₪8▒&8â3*¦₪₪¢â¦¦â÷¦88¦*8┤ʊ±¦à÷^¦שЖ88∑8ë¦jâĀĀĀ&8
*08#«8ЖÆ8j#¢ʊʊâ3÷!زÖ&┤{{*ÖÖÖשששש∂88ژتj┤÷¦àà: ╔▒▒ש8ز÷¦0Æ♠}j&}∂}jزʊא₪ЖК╔Жâ¦÷ت₪ı┤¦¢ששز}3¢8ت₪±jâ∑!÷38j}¦÷♠

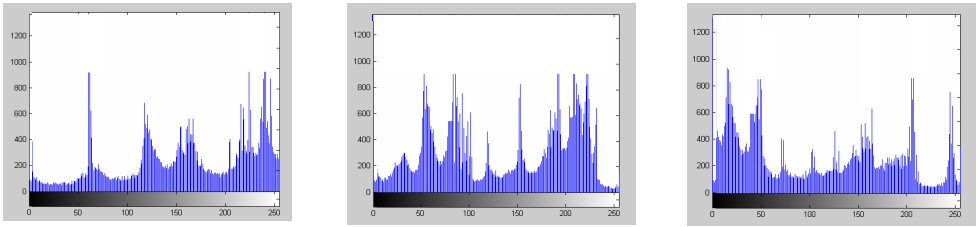
(b)

Figure III.19. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

(a)

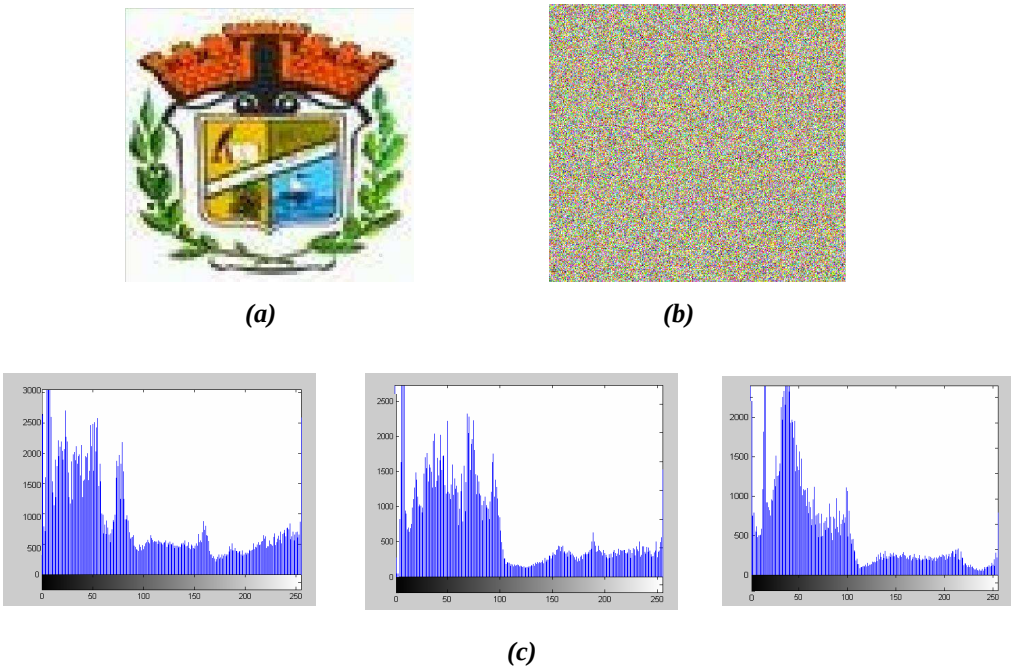


(b)



(c)

Figure III.20. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*



(a) ***(b)***

(c)

Figure III.21. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

III.4.2. Chiffrement à base d'occurrences

Nous avons vu dans la section précédente que le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, dans la présente section nous allons présenter un nouveau codage totalement différent de ceux vu dans les sections précédentes qui ont été bâti autour de la notion de positions de la représentation des éléments constituant la donnée à chiffrer.

Pour appliquer l'approche évolutionnaire à notre problème, nous présenterons les choix utilisés en terme de : représentation chromosomale des solutions du problème, méthode de création de la population initiale des solutions, fonction d'évaluation, opérateurs génétiques et procédure de sélection. Pour mieux approcher le problème, nous présentons une formalisation de ce dernier.

III.4.2.1. Formalisation du problème

Soit D une donnée à chiffrer (texte ou image) qui est une suite de n éléments e_i, et que l'on peut formaliser comme suit :

$$D = \{e_i\}, i \in [1,n] \qquad \text{(III.17)}$$

Dans le cas où D soit une donnée texte, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{C(e_i)\}, i \in [1,n] \qquad \text{(III.18)}$$

où : $C(e_i)$ représente le codage Unicode du caractère e_i.

Dans le cas où *D* soit une donnée image, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{p_i\}, i \in [1,n] \quad \text{(III.19)}$$

Cette représentation (structure) distinguant les différents pixels formant cette image, facilite la conversion en une autre structure en choisissant le RGB (Red Green Blue) comme espace de représentation d'images. Il suffit de remplacer les n pixels par leurs représentations correspondantes en RGB. Nous obtenons, ainsi, la formalisation suivante :

$$D = \{R_i G_i B_i\}, i \in [1,n] \quad \text{(III.20)}$$

Ce mode de représentation (structure) permet de distinguer les différents codages d'éléments de la donnée à chiffrer (les codages des caractères d'une donnée texte, ou les différentes composantes codant les pixels d'une donnée image), et par conséquent de déterminer le nombre d'occurrences de chacun d'eux. C'est ce mécanisme qui sera d'ailleurs exploité pour bâtir notre suivant algorithme de chiffrement proposé.

III.4.2.2. Description d'OEEA (*Occurrences based Evolutionary Encryption Algorithm*)

Comme l'opération de chiffrement consiste à perturber la donnée originale de telle manière à avoir le maximum de désordre dans la donnée chiffrée, nous avons choisi, cette fois-ci, de jouer sur les nombres d'occurrences des éléments (caractères d'un texte ou valeurs des composantes R, G et B codants les pixels d'image), pour avoir la plus grande différence entre les nombres d'occurrences des valeurs similaires d'éléments (nombres d'occurrences des caractères d'un texte original et leurs nombres d'occurrences dans le texte chiffré correspondant ; ou nombres d'occurrences des valeurs des composantes R, G et B codant les pixels d'une image originale et leur nombre d'occurrence dans l'image chiffrée). Ce choix est dû au fait que cette unique donnée (nombres d'occurrences) ne représente pas une information pertinente pour les cryptanalystes.

Dans ce qui suit, nous décrierons les différentes étapes de notre nouvel algorithme de chiffrement proposé.

a. Codage

Les structures (III.18) et (III.20) vues précédemment représentent les données de base à partir desquelles nous sommes arrivés à définir notre codage d'individus en calculant le nombre d'occurrences des valeurs possibles d'éléments dans une certaine donnée. Ainsi, le codage proposé sera le suivant :

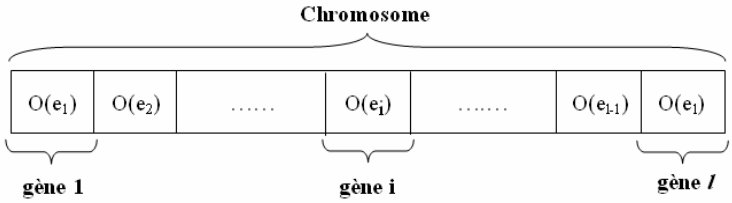


Figure III.22. *Codage des individus sous OEEA.*

Où : O (e_i) est le nombre d'occurrences de l'élément e_i dans la donnée,
l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Remarque :

1) $\sum_{i=1}^{l} O_i = n$ Où n ∈ N représente la taille de la donnée en termes d'éléments (caractères ou pixels).

2) Les éléments qui ne figurent pas dans la donnée auront une occurrence nulle.

Dans le cas de manipulation d'une donnée texte, l'opération de codage consiste à transformer le message en code particulier en calculant le nombre d'occurrence dans le message des 1393 caractères admissibles et l'attribuer à la case qui correspond à son rang dans une table, désignée par TCAR, regroupant les 1393 affichables en Unicode.

Le codage adopté dans ce cas, sera une instanciation du codage général présenté par la figure III.22. C'est celui illustré par le synoptique de la figure III.23.

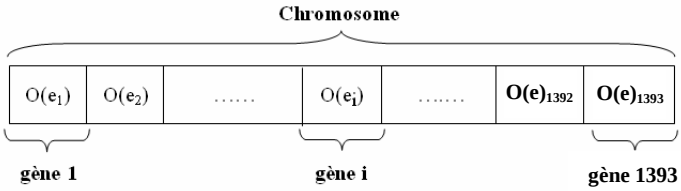


Figure III.23. *Codage des données texte sous OEEA.*

Où $i = \overline{1-1393}$ est le rang des caractères dans la table TCAR et $O(e_i)$ le nombre d'occurrences dans le massage du caractère ayant comme rang i.

Dans le cas de manipulation d'une donnée image et vu que les composantes R_i, G_i et B_i / $i = \overline{1,n}$ prennent leurs valeurs dans l'intervalle [0, 255], notre codage consiste à compter parmi les valeurs de chacune des composantes R, G et B, les nombres d'occurrences relatifs aux valeurs de l'intervalle [0, 255].

Il s'agit de compter à partir des valeurs des n éléments de R, les nombres d'occurrences des valeurs de l'intervalle [0, 255], et de même pour les valeurs des éléments de G et de B. La structure résultante dans ce cas, représente le codage adopté par notre algorithme développé. C'est celle résumée par la représentation chromosomale présentée à travers le synoptique de la figure III.24.

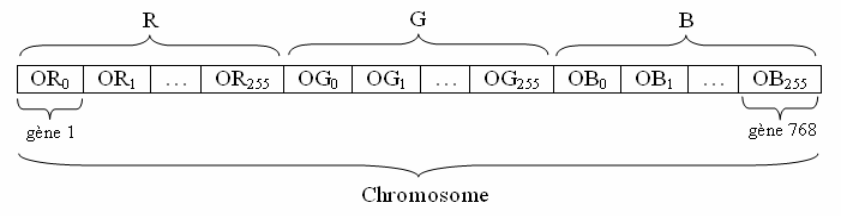


Figure III.24. *Codage des données images sous OEEA.*

Où : OR_i est le nombre d'occurrence des valeurs de la matrice composante R qui égale à *i*,
OG_i est le nombre d'occurrence des valeurs de la matrice composante G qui égale à *i*,
OB_i est le nombre d'occurrence des valeurs de la matrice composante B qui égale à *i*.

Si le codage binaire semble tout à fait adapté pour certains problèmes d'optimisation, cette représentation semble peu appropriée dans notre cas car elle est non intuitive. La représentation qui nous semble la plus naturelle est le codage entier, puisque toutes les opérations de l'algorithme vont manipuler des nombres d'occurrences qui sont des nombres entiers.
Dans ce qui suit, on désigne par individu la donnée à chiffrer ou toute autre donnée, et par chromosome tout individu ayant subi une telle opération de codage.

b. Création de la population initiale

Les m individus (I_1, I_2,…, I_m) formant notre population initiale sont obtenus par application de m perturbations aléatoires sur le chromosome initial I_0 représentant le codage de la donnée originale.

c. Reproduction

Les opérateurs génétiques servent à diversifier la population gérée par l'AE afin d'explorer le plus efficacement possible l'espace de recherche. Nous avons vu dans le chapitre 2 que les AEs disposent généralement de deux opérateurs : l'opérateur de croisement et l'opérateur de mutation.

- **Le croisement :**

Le croisement de solutions peut être toute opération permettant d'obtenir une nouvelle solution à partir de plusieurs individus existants.

Dans le but de compliquer la tâche des cryptanalystes tout en maintenant raisonnable le temps de calcul global de l'algorithme et en évitant toute convergence prématurée, nous avons utilisé un croisement à deux points. C'est le OX « Order Cross-over » proposé par Davis [Davi, 1985]. Les deux points de croisement sont choisis aléatoirement et l'opérateur est appliqué avec un taux qui varie entre 60% et 100% [Gren, 1986].

- **La mutation :**

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

Les individus résultants de ces deux opérations (croisement et mutation) seront rajoutés à la population des parents pour les acheminer vers l'étape suivante qui est celle d'évaluation.

d. Evaluation des individus

La fonction d'évaluation que nous avons définie pour associer des valeurs d'adaptation à chaque individu I_i d'une population Pop, est la suivante :

$$F(I_i) = \sum_{j=1}^{l} \left| O(e_j)_i - O(e_j)_0 \right| \qquad \text{(III.21)}$$

Avec : $I_i = [O(e_1), O(e_2),\ldots, O(e_l)]$, $Pop = \{I_1, I_2,\ldots, I_m\}$ où m est un paramètre de réglage et $O(e_j)_i$ représente le nombre d'occurrences du $j^{ème}$ élément dans le $i^{ème}$ individu.

Dans le cas de manipulation d'une donnée texte, F peut être instancié comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O(e_j)_i - O(e_j)_0 \right| \quad \text{(III.22)}$$

Ou tout simplement :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.23)}$$

Où O_{j_i} est le j^{ème} gène du i^{ème} individu.

Dans le cas de manipulation d'une donnée image, F sera instancié comme suit :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{j_i} - R_{j_0} \right| + \sum_{j=1}^{256} \left| G_{j_i} - G_{j_0} \right| + \sum_{j=1}^{256} \left| B_{j_i} - B_{j_0} \right| \quad \text{(III.24)}$$

Cette fonction est équivalente à celle donnée ci-dessous.

$$F(I_i) = \sum_{j=1}^{768} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.25)}$$

e. Sélection des individus les plus adaptés

La stratégie de sélection de l'algorithme développé va décider de la survie d'une donnée dans la population. Nous avons utilisé la méthode de la sélection par roulette.

f. Critère d'arrêt

Les étapes de reproduction, d'évaluation et de sélection sont répétées jusqu'à ce que l'optimum recherché soit trouvé. Ce dernier représente l'individu ayant la plus grande valeur de convergence durant tout le processus évolutionnaire. Donc, c'est la donnée la plus différente de la donnée originale.

Dans le cas d'une donnée texte, la condition d'arrêt assurant la convergence de notre algorithme évolutionnaire est exprimée à l'aide de la fonction F comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.26)}$$

Preuve :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} (O_{j_i} + O_{j_0}) \quad \text{(III.27)}$$

$$\leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.28)}$$

Et comme :

$$\forall I_i, I_k \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_k} \quad \text{(III.29)}$$

Et :

$$\forall I_i \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.30)}$$

Donc :

$$(III.27) \Rightarrow \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.31)}$$

$$\leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.32)}$$

Dans le cas d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right|$$

Et comme : $((0 \leq R_{ji} \leq n) \,\&\, (0 \leq R_{j0} \leq n)) \Rightarrow (0 \leq \left| R_{ji} - R_{j0} \right| \leq n)$ (III.33)

$((0 \leq V_{ji} \leq n) \,\&\, (0 \leq V_{j0} \leq n)) \Rightarrow (0 \leq \left| V_{ji} - V_{j0} \right| \leq n)$ (III.34)

$((0 \leq B_{ji} \leq n) \,\&\, (0 \leq B_{j0} \leq n)) \Rightarrow (0 \leq \left| B_{ji} - B_{j0} \right| \leq n)$ (III.35)

Alors, d'après (III.33), (III.34) et (III.35) nous aurons :

$$0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq (256 \times n) + (256 \times n) + (256 \times n) \quad \text{(III.36)}$$

$$\text{(III.36)} \Rightarrow 0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n \quad \text{(III.37)}$$

$$\Leftrightarrow 0 \leq F(I_i) \leq 768 \times n \quad \text{(III.38)}$$

Donc, les inéquations (III.26) et (III.38) représentent les conditions d'arrêt mettant fin à notre processus crypto-évolutionnaire dans le cas de manipulation de données texte ou images, respectivement. Nous constatons que $F(I_i)$ est une fonction bornée.

Après quelques tests d'exécution, nous avons constaté que le fait de prendre cette condition uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition d'arrêt ((III.26) ou (III.38)). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous avons arrivé à déterminer ce nombre :

MaxGen = 50

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes dans le cas de chiffrement de texte :

1) $F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i}$

2) MaxGen = 50

Toutefois, elle englobe les deux conditions suivantes dans le cas de chiffrement d'images :

1) $F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n$

2) MaxGen = 50

g. Déchiffrement

Dans cette phase représentant l'opération inverse du chiffrement, une clé de session est générée suivant le même mécanisme utilisé pour nos deux précédents algorithmes. Toutefois, elle est de taille fixe vu que toutes les données texte ont une représentation chromosomale fixe englobant 1393 gènes et toutes les données images ont une représentation chromosomale fixe regroupant 768 gènes.

h. Réglage des paramètres et résultats

Afin de construire un jugement objectif à propos de l'algorithme proposé et de pouvoir noter le maximum de remarques et de comprendre les détails, une large panoplie d'images tests de différentes dimensions a été utilisée. Nous présentons ci-dessous et à titre d'exemple les résultats obtenus pour les mêmes données test précédemment utilisées dans la présentation de nos deux algorithmes développés et précédemment présentés PosESecL1 et PosESecL2 : "Texte 1" de taille 1084 caractères, "Texte 2" de taille 1151 caractères, "Lena" de taille 131X131 pixels et "Logo" de taille 420X395 pixels.

- **Réglage de paramètres**

L'algorithme décrit ci-dessus possède différents paramètres, dont les valeurs influent fortement sur sa qualité. Cette section présente une étude empirique du réglage de ces paramètres afin d'obtenir de bonnes performances, ainsi que le comportement général de l'algorithme qui en résulte.

Les figures III.25 et III.26 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation.

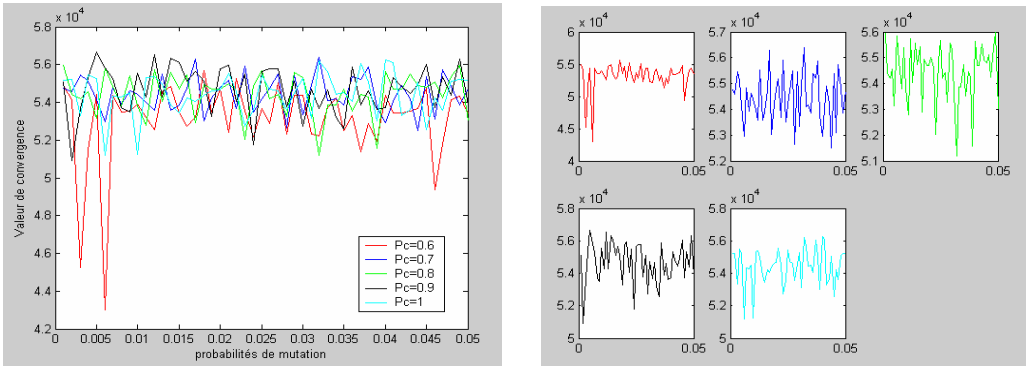


Figure III.25. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

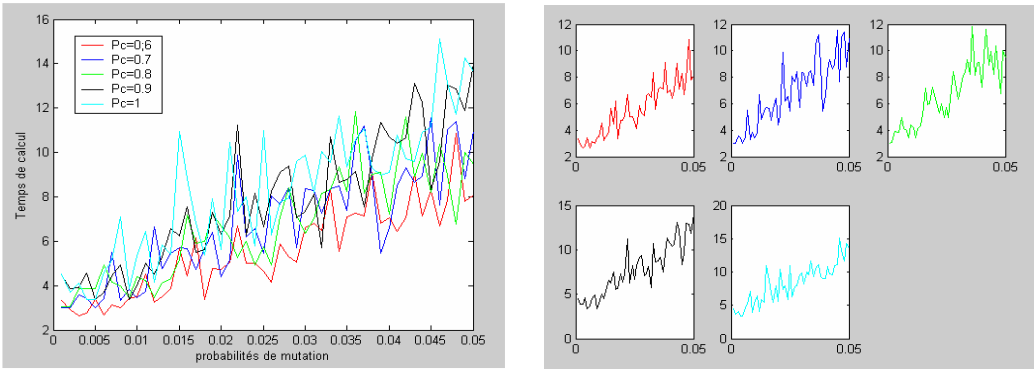


Figure III.26. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Pour choisir les valeurs des paramètres de la probabilité de croisement et celui de la probabilité de mutation, le niveau de confusion n'est pas la seule exigence à satisfaire ; le temps de calcul est aussi important. Ainsi, les valeurs qui nous semblent les plus adaptées sont : *Pc=0.9* et *Pm=0.005* assurant un haut niveau de confusion (VC=56674) en un temps de calcul très raisonnable (T=3.37s).

Après avoir fixé les paramètres de la probabilité de croisement et de mutation, la taille de la population est un autre paramètre à régler expérimentalement aussi. Les valeurs testées sont résumées à travers les figures III.27 et III.28 en indiquant pour chacune d'elles les résultats de chiffrement obtenus en terme de degré de confusion (représenté par la valeur de convergence) et de temps de réponse.

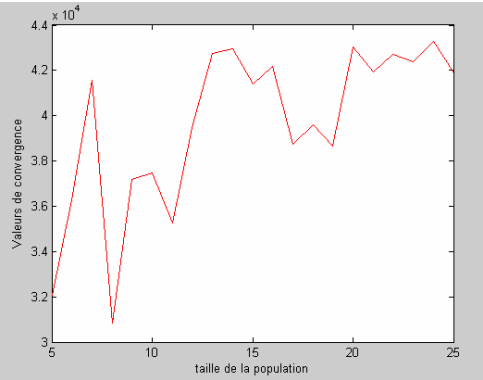


Figure III.27. *Evolution des valeurs de convergence en fonction de la taille de population.*

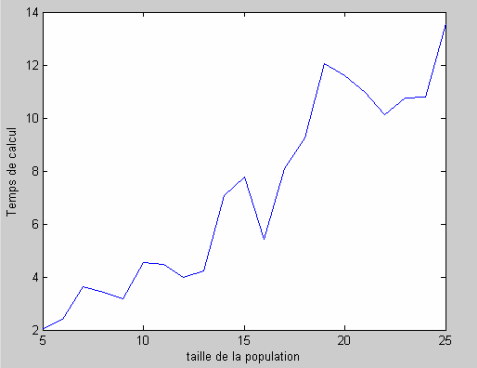


Figure III.28. *Evolution du temps de réponse en fonction de la taille de population.*

D'après la variation de l'influence de la taille de population sur la valeur de convergence et le temps de calcul résumé à travers les deux figures III.27 et III.28, nous optons pour régler le paramètre relatif à la taille de population en lui attribuant la valeur 13 (VC = 42739, T = 4.24s).

Le même principe a été utilisé pour fixer le paramètre relatif au nombre de générations. Donc, et suite à une série de tests sur un ensemble de données soumises au chiffrement et en utilisant la fonction fitness proposée, nous avons retenu certaines valeurs de paramètres qui permettent d'avoir un résultat exploitable en un temps de calcul très raisonnable (de l'ordre de 5 secondes). Ces valeurs sont données par le tableau III.5.

	Valeurs des paramètres de OEEA
Taille de la population	13
Nombre de générations	50
P_c	0.9
P_m	0.005

Tableau III.5. *Valeurs adoptées pour les paramètres d'OEEA.*

- **Résultats**

Après avoir décrit en détail notre méthode nous rapportons, dans cette section, les résultats de l'application de notre troisième algorithme proposé, OEEA, sur les données test citées précédemment (Texte 1, Texte 2, Lena et Logo) en adoptant le paramétrage fixé ci-dessus.

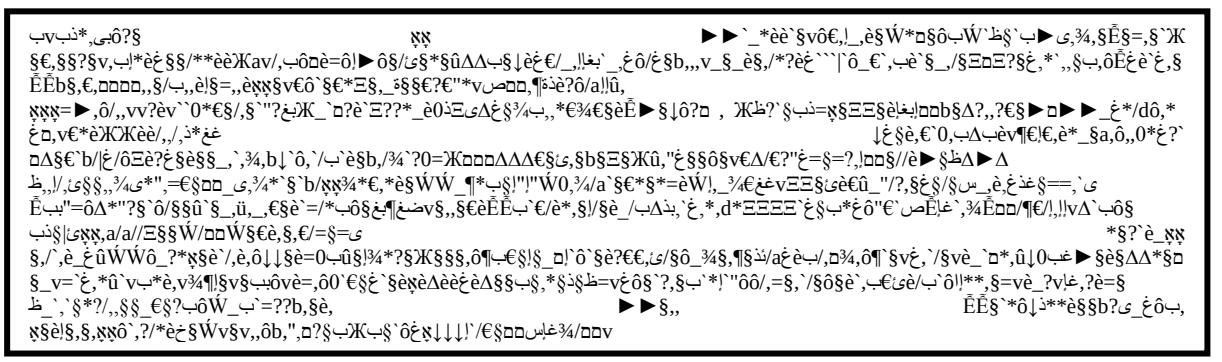
Nous constatons que la taille des clés est indépendante de la taille des données à chiffrer et elle est fixe pour des données de même nature (15323 bits pour les données texte et 7680 pour les données images).

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	15323	2173	6.54
	Texte 2	1151	15323	2284	6.73
Données images	Lena	131 X 131	7680	46034	3.37
	Logo	420 X 395	7680	431340	3.85

Tableau III.6. *Résultats obtenus par OEEA.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)

Figure III.29. ***(a)*** *Donnée originale Texte1,* ***(b)*** *Donnée chiffrée correspondante.*

Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

78 72 75 95 104 96 110 91 108 163 135 87 160 144 129 116 126 83 142
123 102 157 114 82 156 101 164 36 141 103 81 137 151 107 7 111 9 86
100 140 145 153 84 128 133 162 1053 1054 1055 1056 1057 136 120 1089 1090 1091 1092
1093 124 105 80 12 94 134 159 147 127 139 150 155 98 161 90 119 106 154
125 85 152 92 143 148 130 99 132 1281 1282 1283 1284 97 149 131 109 113 93
354 570 347 922 437 923 243 662 465 3 849 213 206 910 924 832 823 456 583
480 925 758 340 568 926 551 369 927 498 791 777 594 638 704 8 394 505 5
789 432 840 690 739 350 813 449 928 796 911 357 525 282 782 604 541 929 810
487 59 698 366 812 462 453 731 403 596 799 930 931 932 225 281 569 933 648
934 820 1386 1387 935 500 476 682 683 921 497 936 261 768 649 904 937 639 447
622 177 938 769 727 496 900 412 200 264 410 914 563 746 901 783 318 642 625
436 778 284 451 459 328 833 43 185 827 186 319 939 592 441 847 502 220 517
455 218 839 603 556 609 286 772 51 850 940 906 664 316 368 209 786 629 589
352 941 878 23 657 273 942 515 869 183 331 790 346 279 861 247 593 943 445
400 558 30 418 344 856 256 595 795 676 624 187 471 944 807 170 757 892 566
64 945 265 779 879 27 269 637 315 168 946 655 607 644 853 742 857 666 564
417 477 947 272 781 305 872 495 806 522 19 580 587 514 640 948 949 876 735
228 920 528 588 950 240 211 1231 1232 1233 915 458 33 287 312 951 545 863 860
571 667 490 952 953 954 888 916 913 299 391 905 671 378 442 669 575 591 726
811 865 202 398 358 399 955 956 314 454 172 341 234 621 546 214 643 737 230
957 223 788 891 259 577 736 958 416 181 263 755 486 50 289 618 959 292 846
864 174 960 519 433 395 434 20 902 961 907 962 1389 1390 1391 1392 1393 275 39
359 964 965 173 966 687 653 337 473 756 700 842 463 329 895 345 818 539 236
787 601 221 793 599 333 919 773 899 504 69 300 701 802 581 58 483 297 967
464 201 721 635 309 255 4 1 800 808 590 605 334 968 548 815 324 969 970
37 203 508 195 835 443 402 870 831 882 65 190 971 250 392 972 307 406 22
401 973 801 387 974 501 237 540 610 222 822 450 798 843 470 765 561 301 439
975 608 482 976 260 351 467 660 885 555 977 552 320 409 267 348 868 373 189
251 205 734 559 14 978 229 431 24 375 659 628 837 612 466 825 979 227 452
619 841 257 178 509 805 980 311 656 871 204 981 623 407 193 1060 1061 1062 1063
754 364 985 385 986 761 520 530 198 288 29 41 485 280 987 1230 1231 1232 1233
1234 290 764 988 989 469 1337 1338 1339 1340 1341 524 507 646 239 627 991 884 526
858 852 681 661 317 1197 1198 1199 1200 1201 826 992 322 699 353 1127 1128 1129 1130
475 993 47 994 887 1140 1141 1142 1143 877 672 70 430 384 684 192 995 527 996
303 997 674 383 828 1049 1050 1051 1052 342 2 326 60 678 491 866 283 26 404
792 1293 1294 1295 650 422 31 886 338 549 68 48 277 484 531 207 585 536 49
1000 705 258 600 651 521 1001 1002 523 438 363 460 15 293 513 381 747 444 550
753 310 1213 1214 1215 1216 1217 1003 898 335 248 245 1004 912 1005 386 274 529 634
630 1006 216 785 503 1257 1258 1259 584 356 620 1007 547 745 271 743 304 617 729
217 285 855 397 330 1008 1009 1010 557 516 875 750 771 355 512 728 296 472 492
1011 361 1012 880 582 809 166 427 446 803 420 494 1013 1371 1372 1373 1378 1379 1380
1381 167 576 784 1014 474 688 733 759 567 468 829 343 1271 1272 670 636 543 626
867 370 598 327 308 606 380 28 658 1015 774 295 844 665 61 560 188 268 732
821 1035 1036 1037 1038 1039 1040 874 371 360 673 663 196 1016 692 184 597 298 1017
614 415 16 1374 1375 723 1376 1377 55 752 715 244 685 215 1018 199 253 848 362
419 212 52 171 616 613 1019 917 425 390 534 1020 918 889 738 897 565 232 270
481 1021 252 488 535 641 396 838 893 797 578 896 766 1022 249 424 1023 1376 1377
1378 1379 388 834 1024 372 224 633 242 794 276 1025 854 631 1026 336 393 775 1356
1357 1358 1359 1360 691 532 675 197 235 499 194 572 730 1027 776 54 414 654 883
862 573 1028 506 349 909 679 586 56 780 873 493 332 428 231 602 377 632 542
824 423 816 53 749 1029 562 554 717 763 11 696 408 894 210 1030 179 702 645
751 804 680 1031 323 718 448 819 457 175 489 325 851 238 1032 306 814 741 374
233 611 461 1033 313 510 668 376 1034 57 748 724 44 1041 1042 1043 1044 1045 695
1046 1277 1278 1279 421 246 294 740 1058 1059 710 1064 1065 1066 1067 1068 719 1069 1070
1071 1072 1073 1074 1075 1076 1077 1078 1079 706 1080 77 73 76 74 79 117 146 1081
1082 1083 1084 63 1085 1086 1087 1088 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104
1105 1106 1107 1108 712 1109 1110 1111 1112 1113 25 169 881 413 382 1114 1115 708 1116
1117 1118 1119 1120 1121 1122 1123 714 1124 1125 1126 615 770 1131 720 1132 1133 1134 1135
1136 1137 1138 1139 18 1144 1145 1146 365 208 440 429 1147 1148 1149 1150 1151 1152 1153
1154 1155 35 1156 1157 1158 1159 1160 1161 903 537 1162 1163 1164 1165 1166 1167 1168 1169
1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 709 1180 1181 982 983 405 984 845 226
1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 66 1202 1203 1204
1205 713 1206 1207 1208 1209 1210 1211 1212 115 138 118 112 122 1218 1219 1220 1221 1222
1223 17 1224 21 1225 1226 1227 1228 1229 998 999 6 836 817 180 1235 1236 1237 1238
1239 1240 1241 1242 711 703 1243 1244 707 722 1245 1246 1247 1248 1047 1048 367 1249 1250
10 62 697 1251 1252 1253 1254 1255 1256 254 389 176 339 191 1260 716 1261 1262 1263
1264 1265 13 1266 1267 1268 1269 34 182 574 1270 1273 1274 693 1275 1276 1280 1285 1286
1287 725 1288 1289 1290 1291 1292 1296 1297 744 278 219 1298 1299 1300 1301 1302 1303 1304
1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 694 1320 32 1321
1322 158 88 89 121 40 1323 1324 1325 1326 1327 1328 1329 45 1330 1331 1332 1333 1334
1335 1336 1342 1343 67 1344 1345 1346 1347 1348 1349 1350 1351 46 1352 1353 42 1354 1355
544 426 291 647 762 262 963 411 1361 1362 1363 1364 1365 990 302 38 579 479 435

1366 1367 1368 1369 1370 538 859 686 241 908 760 533 165 830 511 266 553 1382 1383
1384 1385 518 677 890 1388 689 652 478

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسيًا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال

(a)

gé¥¥§&◄¥c:¥~x┼Ч‌x;>ش╪♫♫⇑¦;>}&◄◄¦¥"<;g§§q5⇑ ¥¥;}¬;§¦<<<î{xcx;<¥¥§xקקt;:g¥¥;ى";<xx¥:§¥§}§;¦;¥¥x;¥<
³x۞۞⇑۞rgt¦¥xrxg◄t¬g¦x4và¦x¥x>§_⇑¦ƒîx¦;<ƒv*§r¥ىx§{¥x¥xqx;⇑ ↔↔îsxî§⇑;¥§<{Ч<j³:&r'קק¥╪xr:
ى۞q¬xƒ>♫¦¥<:q¸↔~®¥xé§g§۞۞;ش;g&x؟♫§xq;
۞¬{î<¥l§;c§¥xî¥§§x{xî۞0>¦◄хњхњхîàњ¦{¥x_:q§Ч§&r¥x¥étsЧЧ{&>¥x;;¦x╪ى;¥x&4§¥:&³³<"îx§ЧЧЧé§xq{¬<ى&{□¥à&x¦¥⇑⇑&>;<
¥î§§¥§ît¥¥rr§;î&;;>g{x¥x¥¥q;<x§;+ ┼┼┼>g§§{¥x;qs;îrxgxƒ¥⇑&x§¥{_î<³¦cx¥ƒ{x>;x۞x5¥⇑xrg;§x&¥¬xx: ◄x§ƒ³{xj¥¥q{î¥;¥j◄5&{g;;
њњq¥¥x&é;g;/>:;l¶‌x³xg¥î;¦;îxx۞۞q<¥:&&♫;¦p¦:/v¥xxxקp¥&§jx;x¥q◄◄¥>;{↔۞î~_::{;x_§xњ&{ﺔ¬¥gƒ¥éî>¥¥§§tl۞;¦x;x,x¶‌¥¦l:g{
qsq¥¥x<¦¶‌Ч;ºx¦>¦x;&q¥:§; ◄xàtgƒ{¦§¥⇑cq¥ggî³¦xî♫x<t{§>;{¥t¬q>¬╪g<§q:>®&_§;<ىr↔xx>¥¦¥Ч‌g<¥àtt~¥éxà5¥ƒ;_<<¥
♫¬xrîx¥§¥¥g~¥¦j¥{x¥↔↔¥¥g{¥§~§¥§¥¥⇑¥;;¥gƒ!lx;:¬;{gњњњx§╪╪xx&¦g↔↔{¥q۞x¥¥xx¥r{t{<®ggî;□۞!gx_ﺔ{_:x{t¦¶‌Чt>◄◄◄
4g¥x<<¥¥q;à¥⇑qg³x¥;¥ƒs<_và¥r§♫xàx;³ §>xƒƒ◄◄x۞tq¥{>îxt⇑x¥¥éxxs;_xىs;{>
קà'xq{ƒ<<קקx¶‌xrgsrv<xњњ4¥;x<x³sg;§ج¦x>¦t♫§§&g{<{;;¥qקx³gњњxƒ¥;ىxéj¥s۞њsss¥¥¥¥קxq_x;¥>¦<§î¥qstc5;sj³³¥s§>;:_ﺔé~g¥¥<x:4<
۞۞{¥ts§<{s§xxrq¥<¥{;:¥¥>&¥;x¦g♫:s;:®¥gsx;¥<v¦s/¬}>ى¥¸x⇑⇑s~sx;w{◄x:ىº¦ig§jx_¶‌Ч&;g_؟>s{&◄۞¥t¦5;s®{{♫⇑¥ش§¥¥>!x
<۞¥x<x;0>¦<>xt;¥î♫{x¥⇑îjs<¦;↔x¦¥><;&xrקקקs¥¬g&<x⇑:x{î³³x:îЧЧ_'{g§◄r§¥;<x/¥<⇑⇑{;¥{{{:□~¥x۞

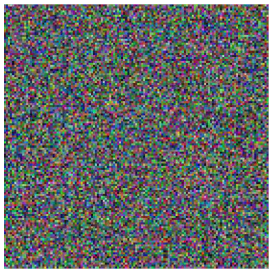
(b)

Figure III.30. **(*a*)** *Donnée originale Texte2,* **(*b*)** *Donnée chiffrée correspondante.*

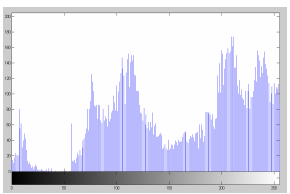
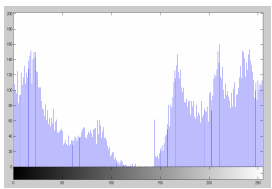
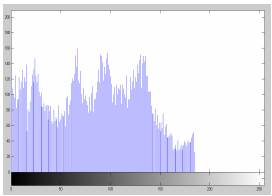
Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

59 1091 1 96 52 8 21 1200 47 54 71 16 43 14 1366 1369 25 64 26
1271 30 32 89 15 65 68 53 28 1303 1305 42 84 81 78 1090 34 24 62
20 58 74 56 39 1117 4 85 99 1324 18 98 1306 29 72 55 66 5 45
49 17 40 23 73 80 97 50 608 102 848 575 286 1220 1223 167 980 869 803
925 1201 134 310 1278 622 866 190 472 727 1118 535 818 1367 114 377 326 835 553
1368 166 488 100 1270 874 206 981 307 232 873 880 676 538 712 578 982 305 860
1325 899 933 842 332 865 315 713 239 445 619 203 939 983 785 192 1302 163 749
872 1327 297 605 513 722 270 940 269 814 198 943 201 570 195 422 746 897 802
402 904 945 250 794 560 691 1276 217 693 514 1304 852 935 804 666 518 1222 133
467 915 156 910 407 1221 170 257 738 542 692 912 984 508 443 245 815 671 372
63 61 67 985 1277 174 128 477 930 639 986 800 987 667 258 988 456 309 989
817 856 688 607 559 520 459 577 188 439 369 325 643 314 172 363 796 592 312
105 193 124 157 148 582 669 822 566 390 779 806 397 1187 1188 573 849 901 917
813 149 317 285 696 180 337 725 824 634 1057 223 990 829 554 953 466 732 242
733 905 647 308 168 101 487 958 1326 793 572 812 454 517 449 991 831 366 540
704 868 356 212 913 847 321 408 1059 280 623 909 484 271 137 662 220 433 324
881 1375 841 1333 896 837 624 992 891 1334 649 724 581 161 502 1275 545 685 924
843 1058 473 585 1374 828 833 275 993 382 450 176 107 1285 375 427 291 929 795
299 362 599 1283 863 178 736 515 531 937 331 292 994 742 888 1332 313 596 126
381 648 359 411 489 431 376 234 120 934 1284 175 714 522 955 534 247 614 598
652 1378 995 143 410 601 996 997 1273 453 883 158 208 719 298 430 689 446 462
357 104 840 928 184 413 303 657 364 399 870 256 396 636 424 159 391 1384 1385
115 1319 702 277 279 419 241 329 1000 734 698 627 469 871 521 690 602 262 140
342 687 1376 1001 464 887 350 116 268 1272 1274 895 720 465 267 197 1002 588 541
468 361 808 132 204 141 1377 287 444 547 438 150 761 300 221 425 921 1320 348
403 737 611 496 304 862 171 586 709 792 820 834 127 406 274 229 145 194 1003
886 546 392 922 288 673 731 448 205 276 561 263 1064 956 625 1004 186 227 492
1005 442 254 593 1006 1007 395 584 260 902 1065 530 968 590 334 389 1008 558 708
877 393 412 475 678 646 526 1203 1204 273 533 832 470 243 1009 579 884 474 451
786 850 908 723 1066 405 501 1010 1011 117 677 1012 498 658 1382 699 765 819 1013
214 1014 1015 233 423 135 1016 650 164 504 615 721 222 457 900 892 911 455 954
1017 642 1018 351 151 838 138 500 1381 505 1019 437 458 226 436 385 580 946 1020
179 131 400 653 173 1101 306 481 213 218 240 686 707 106 626 380 1021 511 358
333 146 1063 620 674 610 111 811 567 1022 185 507 603 878 338 706 1023 283 482
1100 266 379 323 799 926 432 1098 906 893 951 845 1024 388 668 568 718 855 764
110 701 949 224 112 864 756 551 125 920 680 1206 209 142 409 1025 207 612 844
670 633 587 367 645 302 130 591 1312 805 383 621 1026 169 679 103 349 1027 660
655 781 571 544 861 641 730 480 1028 322 1029 628 705 255 682 493 1030 1031 162

890 211 919 1099 478 927 1032 661 355 1205 768 932 1033 853 1067 236 1034 340 681
556 527 801 537 486 182 327 613 189 606 604 703 265 384 1035 617 231 1193 246
165 942 1036 728 640 821 875 200 597 659 261 717 441 415 854 1037 108 1192 569
823 374 281 894 539 923 807 452 202 154 1359 483 440 177 882 485 248 562 319
495 160 387 695 259 697 122 876 401 1190 651 576 414 879 1038 230 966 635 426
519 916 600 950 210 352 118 751 237 918 563 386 963 282 630 656 754 365 797
1182 1183 609 716 295 740 339 476 370 1039 330 378 675 1055 228 885 557 851 113
715 272 594 741 747 826 238 858 672 574 109 497 1191 447 152 353 121 663 494
760 532 499 129 898 404 1040 583 1184 524 565 638 344 294 416 253 644 629 119
938 284 429 1056 320 360 479 555 798 346 825 183 710 947 1041 936 1080 318 595
249 632 827 199 743 1180 336 637 683 435 421 191 729 684 944 810 460 512 509
503 857 1042 418 1094 739 654 1043 1044 525 529 839 1093 809 510 1181 664 771 244
394 311 434 859 1095 931 368 589 316 123 941 957 948 181 139 354 1045 1046 776
784 1047 1048 1049 1050 830 398 264 335 1051 1052 775 1053 1054 420 543 75 76 1060
979 1061 1062 1068 1069 1070 1071 1072 762 1073 1074 196 506 1075 1076 1077 969 1078 1079
11 9 1123 1124 1081 1082 293 373 1083 1084 973 965 1085 1086 296 564 1087 1088 1089
83 36 90 77 6 92 3 91 10 95 33 1092 371 219 1096 1097 726 251 1102
252 903 1103 1104 1105 1106 1107 147 889 552 1108 1109 774 1110 1111 1112 790 1113 773
1114 735 289 1115 1116 37 69 1119 1120 1121 1122 1125 788 975 1126 1127 461 780 1128
960 1129 1130 341 215 1131 1132 1133 971 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143
1144 1145 1146 976 1147 1148 1149 1150 769 1392 1393 1151 1152 1153 962 1154 1155 1156 225
694 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 216 914 153 1347
301 1171 1172 1173 1174 1175 752 1176 1177 1311 290 836 1178 1179 1185 1186 490 471 550
1189 1353 1194 766 1195 782 1196 1313 1197 1198 1199 665 428 1202 907 616 763 1207 977
1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 767 379 321 1224 758 1225 1226
1227 1228 1229 523 278 1230 1234 1235 770 1236 1237 1238 1239 748 1240 1241 1242 22 31
44 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1310 1254 1255 967 1256 1257 1258
1259 1260 753 964 1261 1262 816 1265 1266 618 516 1267 1268 1269 789 1279 1280 13 12
48 41 1281 1282 783 463 1286 1287 1288 1289 1290 959 1291 1292 1293 1294 70 7 1295
1296 1297 548 187 1298 1299 1300 1301 136 536 1307 1308 1309 970 744 1314 1315 757 777
1316 1317 1318 791 1321 1322 1323 27 35 1328 1329 144 345 952 1330 1331 961 1335 1336
1337 51 79 1338 1339 1340 711 235 1341 1342 1343 974 1344 745 1345 1263 1346 94 86
1348 1349 1350 867 155 1351 1352 528 750 1354 998 328 999 1355 1356 60 88 1357 82
46 38 87 57 1358 772 1360 1361 1362 343 549 491 417 1363 1364 1365 1370 1371 767
1372 1373 347 700 1379 1380 778 631 1383 759 846 1264 1386 1387 93 2 1388 1389 755
1390 978 787 972 1391

(a) ***(b)***

(c)

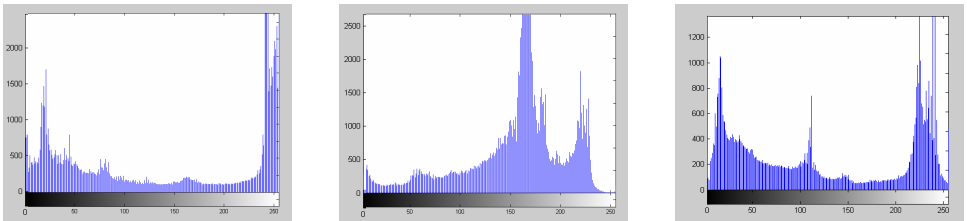
Figure III.31. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 0,9375 K octets) :

369 423 529 171 397 412 474 15 689 64 112 390 267 21 118 304 6 28 574
465 356 338 717 638 339 110 243 458 508 450 361 608 635 115 593 749 151 742
575 716 428 109 486 584 294 722 521 141 87 695 373 225 452 602 679 13 208
85 468 634 709 298 70 297 514 601 439 359 320 251 222 292 319 765 456 614
142 645 625 288 383 217 125 476 718 274 236 729 483 567 114 487 492 303 542

426	628	675	713	296	505	370	730	409	395	764	641	396	669	460	337	264	230	204
623	358	293	418	150	520	134	328	469	762	440	533	651	551	535	286	438	462	103
355	347	63	158	16	561	205	410	643	703	432	519	693	247	620	346	40	736	366
129	652	164	327	371	506	363	154	241	447	434	624	585	163	748	376	380	144	407
59	143	667	252	10	536	548	766	360	335	455	311	497	200	466	249	751	35	119
165	341	36	741	398	448	656	18	65	233	734	17	531	512	595	660	104	586	81
435	299	101	92	309	454	436	710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579	352	739	698	306	253	577	248
735	515	342	285	61	219	138	568	705	658	427	511	532	210	357	307	153	633	711
98	433	619	691	60	20	137	733	485	324	659	419	719	684	269	392	754	68	259
284	467	295	48	300	224	541	600	715	630	613	172	66	425	558	604	38	692	34
557	263	258	384	51	317	375	111	325	146	701	472	388	607	289	94	644	377	745
580	500	598	496	240	443	594	364	554	74	637	187	712	655	747	576	145	47	501
417	490	622	162	491	626	578	699	750	14	128	493	282	226	459	351	589	209	582
545	740	596	147	275	121	362	470	53	193	291	333	290	179	523	527	42	30	402
738	525	113	374	664	646	507	609	685	756	189	700	148	260	654	464	191	350	378
176	257	538	599	603	11	702	372	571	237	12	96	170	192	653	88	416	540	25
180	522	642	166	72	246	194	707	680	33	639	159	215	75	728	227	516	681	344
90	99	559	127	513	265	7	23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555	69	708	345	494	271	238	726
393	534	132	385	587	686	647	368	203	3	481	482	332	199	167	687	560	539	495
743	44	583	91	610	690	336	31	261	403	379	255	725	498	323	386	198	572	220
484	73	322	677	606	340	272	343	58	503	381	26	421	517	761	79	676	334	348
760	49	663	235	597	566	313	627	39	414	618	518	479	55	752	186	694	563	649
175	223	133	549	420	277	510	727	305	391	228	632	404	196	206	84	52	184	755
212	581	763	429	107	697	724	590	77	737	445	71	682	502	509	120	254	413	758
640	149	616	43	195	242	239	211	611	453	757	218	86	19	310	387	2	105	117
424	244	504	116	449	591	8	229	605	399	629	182	57	461	314	588	177	234	190
102	106	714	122	617	612	216	673	312	431	326	621	80	382	721	93	45	156	665
553	767	475	174	329	394	415	547	188	331	543	365	308	46	67	670	330	157	354
302	389	480	367	232	768	27	207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631	41	471	131	201	245	54	202
444	556	544	720	270	32	688	759	442	279	478	136	197	123	273	671	437	262	457
672	268	349	95	50	488	706	546	183	221	178	160	528	696	62	9	214	615	573
489	430	97	744	22	753	524	446	213	746	731	181	231	283	661	570	401	276	78
400	4	562	526	683	301	89	477											

(a) ***(b)***



(c)

Figure III.32. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session (Taille$_{Clé}$ = 0,9375 K octets) :

503	463	515	233	412	71	181	502	153	365	302	449	225	362	500	33	523	198	519
471	404	521	497	145	496	504	494	205	138	505	6	522	214	149	451	422	498	146
520	63	453	320	516	510	350	287	495	39	387	385	20	115	80	361	472	69	513
213	204	105	425	68	311	507	506	208	508	24	41	499	511	237	54	415	509	37
518	517	52	458	384	493	390	8	191	78	501	40	163	514	229	512	99	574	372
579	351	396	36	389	17	370	312	626	436	5	1	284	118	629	334	594	625	174
704	648	28	534	469	657	342	411	479	401	45	533	234	675	379	23	235	231	223
77	324	729	634	548	217	555	142	673	159	346	568	551	85	195	343	333	408	158
10	563	532	373	264	73	242	359	179	280	473	709	605	128	327	82	756	308	638
588	152	31	766	272	285	166	667	714	51	613	157	439	265	561	271	314	277	491
286	32	125	2	332	194	547	552	241	156	443	661	143	455	431	120	215	484	578
434	526	647	344	257	371	744	168	199	584	260	707	486	763	369	244	524	298	540
689	405	764	196	595	538	345	478	636	464	95	216	29	567	46	735	57	331	397
752	426	377	288	133	114	134	62	424	668	720	399	200	336	696	459	141	553	173
702	169	148	171	460	622	210	35	409	201	570	209	444	14	624	558	299	483	66
366	184	224	600	100	671	738	293	677	549	192	193	291	554	230	76	481	639	67
124	545	620	226	93	529	303	136	164	470	603	150	687	419	304	683	414	581	74
565	127	245	144	577	337	760	475	557	705	301	49	418	137	44	110	7	79	306
207	13	635	380	276	758	420	278	248	238	292	630	457	47	437	583	445	375	723
354	492	684	255	679	81	263	119	221	61	682	751	669	652	104	681	659	290	92
121	454	188	172	564	355	754	708	489	251	394	220	12	60	97	126	566	604	403
601	15	395	429	330	715	645	546	490	759	108	154	423	34	441	87	741	575	643
666	101	19	693	686	718	111	631	326	542	617	674	706	84	178	543	381	50	167
733	402	716	654	462	614	239	300	254	725	262	619	232	428	305	593	335	161	427
83	250	341	180	374	632	465	406	719	183	378	623	382	243	147	740	765	296	649
658	736	582	252	165	712	745	89	413	569	160	106	633	329	177	608	340	461	589
537	261	456	38	103	607	640	627	98	4	30	597	416	281	596	599	536	253	734
26	742	477	642	266	438	135	386	573	295	339	721	711	190	487	753	591	539	162
737	610	726	313	140	197	70	16	525	621	637	86	749	182	572	11	474	139	612
730	347	598	43	556	541	319	48	761	187	664	748	275	646	672	258	151	219	710
259	256	236	348	450	609	109	325	186	571	580	678	703	435	430	694	606	376	274
240	328	206	393	175	466	615	315	587	75	383	602	616	527	25	273	762	448	697
113	452	132	155	724	364	739	294	485	728	202	750	476	650	544	116	480	688	42
338	530	352	123	743	247	641	676	368	211	176	107	467	698	322	55	560	56	731
685	270	170	767	227	655	690	279	310	746	644	203	732	680	663	662	417	64	528
651	433	691	96	59	246	585	757	112	318	283	618	628	22	297	665	592	535	94
65	58	222	102	586	699	3	695	358	228	717	212	392	576	531	289	130	88	356
388	316	122	768	660	421	446	185	611	268	670	410	398	468	249	321	755	72	9
18	692	700	117	323	722	447	307	353	360	131	90	407	129	363	562	432	488	590
27	367	400	559	349	317	391	440	656	653	727	218	442	482	282	713	53	747	91
269	309	189	267	21	701	550	357											

III.5. Discussion et évaluation des résultats

Un bon crypto-système doit satisfaire les six points clés de Kerckhoffs [Kerc, 1883] et ceux de Shannon [Shan, 1949]. Plus explicitement, la qualité d'un crypto-système se mesure principalement par sa résistibilité face aux différents types d'attaques. La technique de cryptage proposée à travers nos trois algorithmes proposés PosESecL1, PosESecL2 et OEEA et qui consiste en une perturbation évolutionnaire de la donnée originale, assure une bonne sécurisation contre les attaques les plus destructives. Les critères utilisés pour évaluer leurs sécurités sont les suivants : la vitesse de l'algorithme, l'attaque statistique, l'attaque différentielle, l'attaque exhaustive et l'analyse de l'espace des clefs.

III.5.1 Vitesse des algorithmes

En plus du paramétrage, la vitesse d'un algorithme évolutionnaire dépend étroitement du codage utilisé. Un mauvais codage affecte non seulement la qualité de l'algorithme en termes d'optimalité de solutions mais aussi en termes de temps de convergence. Dans le cas de nos deux premiers algorithmes proposés, PosESecL1 et PosESecL2, le codage dépend de la taille de la donnée à chiffrer ce qui affectera la vitesse des algorithmes puisque le temps de traitement des données de grandes tailles sera beaucoup plus grand que celui de traitement des données de petites tailles, chose due à la taille des chromosomes manipulés durant les générations de l'évolution. Ceci se voit clairement à travers les résultats obtenus par ces deux algorithmes où le temps de convergence de PosESecL1 est meilleurs que celui de PosESecL2

du fait que la taille des chromosomes codant une donnée pour le PosESecL2 est trois fois plus grande que celle des chromosomes codant la même donnée pour le PosESecL1. Toutefois, le codage proposé à travers OEEA unifie la taille des chromosomes codant des données de même nature (textes ou images) et de différentes tailles. Ainsi, le temps de traitement est indépendant de la taille des données traitées. En plus et d'après les résultats expérimentaux présentés précédemment, nous constatons que le temps de convergence de ce dernier algorithme est très raisonnable (de l'ordre de 4 secondes).

Comparons, maintenant, la vitesse de nos trois algorithmes développés avec celle des principaux crypto-systèmes (AES du coté des crypto-systèmes symétrique, RSA du coté des crypto-systèmes asymétrique et SEC qui est un algorithme de chiffrement exploitant les algorithmes génétiques). Le tableau 7 et la figure 33 présentent un résumé des temps de chiffrement et de déchiffrement de chacun des algorithmes cités.

	PosESecL1	**PosESecL2**	**OEEA**	**AES**	**RSA**	**SEC**
Temps de chiffrement (s)	24.5	46.15	3.37	0.5	42.8	1.8
Temps de déchiffrement (s)	0.26	0.75	0.12	0.08	42.5	0.06

Tableau III.7. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

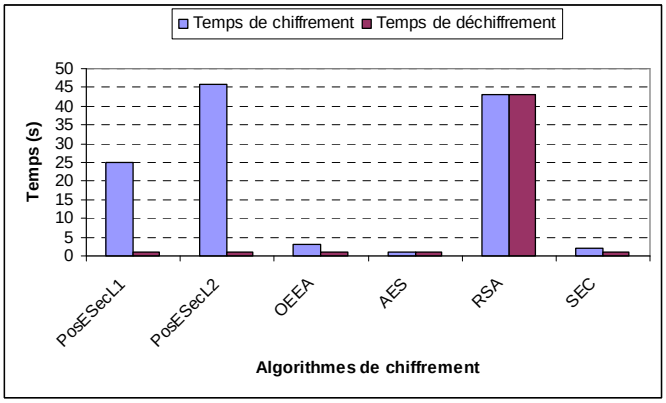


Figure III.33. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

D'après le tableau et la figure, ci-dessus, présentant le temps de chiffrement et celui de déchiffrement de nos algorithmes en comparaison des algorithmes SEC [Omar, 2007], RSA [Mene, 1996] et AES [www4], nous remarquons que :

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement beaucoup plus petit que celui de RSA. De son coté PosESecL1 possède aussi un temps de chiffrement/déchiffrement meilleur que celui de RSA. Toutefois, PosESecL2 a un temps de chiffrement de l'ordre de celui de RSA et un temps de déchiffrement meilleur que celui de RSA.
- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin de celui d'AES. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes PosESecL1 et PosESecL2 sont plus grands que celui d'AES.

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin que celui de SEC. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes sont plus grands que celui de SEC.

De même et d'après les résultats expérimentaux obtenus et présentés précédemment, nous constatons que le temps de calcul de OEEA est indépendant de la taille de la donnée à chiffrer du fait de l'unicité de taille de codage de toutes les données de même nature soumises au chiffrement. Toutefois, ce n'est pas le cas pour PosESecL1 et PosESecL2 où le temps de calcul des deux algorithmes est strictement dépendant de la taille de la donnée à chiffrer. Ainsi, OEEA est l'algorithme ayant un temps moyen de calcul le plus adapté.

III.5.2 Niveau de confusion

La confusion est l'une des notions essentielles énoncées par *Shannon* sur lesquelles se base un algorithme de cryptage. Elle sert à cacher la relation entre la donnée en clair (originale) et la donnée chiffrée correspondante. Donc, elle vise à rendre la donnée aussi peu lisible que possible.

Dans notre cas, les deux modes de chiffrement développés, à savoir le chiffrement à base de positions et le chiffrement à base d'occurrences, sont de niveaux de confusion différents.

Pour le premier, les deux algorithmes proposés dans ce mode sont aussi de niveaux de confusion différents. En effet, le niveau de confusion de PosESecL2 est meilleur que celui de PosESecL1 du fait que PosESecL2 exploite les positions des éléments codant les différents composants de la donnée, considéré chacun comme unité séparée (gène) du codage final de la donnée (chromosome). Ainsi, la donnée chiffrée aura de grande chance de contenir de nouveaux éléments ou composants ne figurant pas dans la donnée originale. Cette chose complique considérablement, voire même, pénalise toute cryptanalyse possible comme nous allons le démontrer dans les sections qui suivent. Toutefois, PosESecL1 est de niveau de confusion strictement dépendant de la taille de la donnée à chiffrer puisqu'il exploite les positions des éléments de la donnée. Donc dans le cas de données de petites ou de moyennes tailles, une attaque exhaustive finira de trouver la bonne combinaison initiale de positions et, ainsi de casser l'algorithme.

Pour le deuxième mode de chiffrement développé, il offre un très bon niveau de confusion grâce à l'utilisation d'un codage bâti autour de la notion de nombres d'occurrences. Ce mode ne présente ni information utile pouvant être exploitée par un cryptanalyse ni possibilité de reproduction de la donnée en se basant uniquement sur cette donnée surtout que le codage adopté assure que les composants de la donnée (caractères ou pixels) changent dans la donnée chiffrée par rapport à la donnée originale. En plus, le niveau de confusion de OEEA, l'algorithme développé dans ce mode, est meilleur que celui de PosESecL2 et ainsi de celui de PosESecL1 du fait que l'espace des nouveaux caractères qui peuvent apparaître dans une donnée chiffrée par rapport à la donnée originale correspondante est beaucoup plus grand dans OEEA que dans PosESecL2.

III.5.3 Attaque statistique

Ce type d'attaque considère le crypto-système comme une boite noire, il analyse statistiquement les entrées et les sorties de ce système. Pour ce faire, nous avons utilisé les mesures suivantes dans le but d'évaluer ou de quantifier la différence entre l'image originale et l'image cryptée correspondante :

Le facteur *NPCR* (*Number of Pixels Change Rate*) donné par l'expression (III.39), l'erreur absolue moyenne (*MAE : Mean Absolute Error*) donnée par l'expression (III.41) et l'erreur

quadratique moyenne (*MSE :Mean Square Error*) donnée par l'expression (III.42). Le tableau 8 résume les valeurs des différentes mesures obtenues après les tests qui ont été effectués sur l'image Lena et l'image chiffrée correspondante obtenue dans le cas de nos différents algorithmes proposés.

	NPCR	**MAE**	**MSE**
PosESecL1	0.6314	0.91	0.58
PosESecL2	0.9073	0.97	0.61
OEEA	0.9955	1.03	0.75

Tableau III.8. *Niveaux de confusion.*

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \tag{III.39}$$

$$D(i,j) = \begin{cases} 0 Si\ \mathrm{Im}_O(i,j) = \mathrm{Im}_C(i,j) \\ 1 Si\ \mathrm{Im}_O(i,j) \neq \mathrm{Im}_C(i,j) \end{cases} \tag{III.40}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \tag{III.41}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \tag{III.42}$$

D'après les résultats d'application des mesures quantifiant la différence entre l'image originale et ses versions chiffrées calculées par nos algorithmes, il est clair que la majorité des pixels formant l'image originale ont changé soit de positions (dans le cas d'application de PosESecL1) ou de contenu (dans le cas d'application de PosESecL2 ou de OEEA) pour, ainsi, calculer l'image chiffrée. En effet cela se voit surtout à travers les valeurs de NPCR qui calcul le niveau de différence entre les deux images où les résultats obtenus sont proches de la valeur optimale qui vaut un (01). Toutefois, ces valeurs ne sont pas les mêmes pour les trois algorithmes, chose qui est tout à fait normale vu que les algorithmes sont de niveau de confusion différent (voir section précédente). Et comme nos algorithmes présentent l'avantage d'être non déterministes, donc chaque nouvelle application d'un même algorithme donnera lieu à des résultats différents ce qui entraîne que l'application des mesures NPCR, MAE et MSE donnera lieu, de son tour, à de nouvelles valeurs. Ainsi, aucune propriété statistique ne peut être déduite d'une donnée chiffrée et l'attaque statistique ne possédera aucune chance de casser nos algorithmes.

III.5.4 Attaque différentielle

Cette attaque essaye de tirer des conclusions sur le fonctionnement d'un algorithme de chiffrement en comparant des versions chiffrées de plusieurs données originales et dans un meilleur cas des versions chiffrées de plusieurs blocs formant une même donnée. Ainsi, dans le cas des algorithmes de chiffrement par blocs (DES [Stin, 1996], [Biha, 1991], [Mats, 1994], 3DES [Stin, 1996], [Gant, 2001], AES [Lepr, 2000], etc), quand la donnée (surtout pour le cas d'images) contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement, ce qui fait que la donnée cryptée contiendra des zones texturées ; mais vu que nos algorithmes traitent la donnée en une seule passe, c'est-à-dire ils opèrent sur la donnée totale et non pas des blocs de la donnée ; alors la cryptanalyse

différentielle sera mise à l'écart. De ce fait aussi, nos algorithmes présenteront une certaine robustesse au bruit par opposition aux algorithmes de chiffrement par blocs, et qui se traduit par le fait qu'une erreur sur un bit chiffré ne va pas propager des erreurs importantes dans tout le bloc courant et par la suite dans toute la donnée lors de la recombinaison des blocs.

Même si le cryptanalyste pense à essayer de tirer des observations à partir des résultats de cryptage de plusieurs données pour essayer d'imiter le fonctionnement des algorithmes, l'approche évolutionnaire pseudo-aléatoire complique considérablement la tâche de ce type d'attaque. Une sensibilité à la donnée originale est aussi un point avantageux des algorithmes proposés vu que la donnée cryptée correspondante à une certaine donnée originale change d'une instanciation du problème à une autre. Autrement dit, nos algorithmes développés sont des algorithmes non déterministes à l'inverse du RSA, par exemple, qui a nécessité au préalable de randomiser les données soumises au chiffrement pour éviter les attaques exploitant les modèles connus des données en clairs [Ster, 2004].

III.5.5 Attaque exhaustive

L'attaque exhaustive ou encore dite attaque à force brute est une attaque qui demeure fatale pour les crypto-systèmes utilisant des clés de petites tailles. À ce stade, nos trois algorithmes proposés présentent des niveaux de sécurité différents. La robustesse de PosESecL1 est strictement dépendante de la taille de la donnée originale. PosESecL2 présente un niveau de sécurité plus élevé que celui de PosESecL1 du fait que sa clé soit de taille quatre fois plus élevée que celle de PosESecL1 dans le cas de manipulation de données textes et de trois fois plus grande dans le cas de manipulation des données images. Ainsi, et vu qu'à l'heure actuelle la taille de la clé assurant une résistibilité face à une attaque exhaustive est de 512 bits, alors le cryptage par PosESecL1 de toute donnée possédant une taille inférieure à 512/le nombre de bits nécessaires pour coder la taille de la donnée (nombre de caractères ou de pixels), sera exposée au risque d'une attaque réussite par force brute. C'est le cas aussi pour les données cryptées par PosESecL2 et qui sont de taille inférieure à : 4*(512/le nombre de bits nécessaires pour coder la taille de la donnée texte) ou de 3*(512/le nombre de bits nécessaires pour coder la taille de la donnée image). Toutefois, OEEA ne peut en aucun cas être cassé par une telle attaque vue que la taille de la clé utilisée est largement sécuritaire. Ainsi, OEEA est l'algorithme le plus sûr parmi nos trois algorithmes développés.

Considérons le texte à chiffrer « **there is only one God and Mohamed his prophet** », nous reportons dans le tableau suivant (tableau 9) les tailles de clés de chiffrement de nos trois algorithmes développés en comparaison de DES, 3DES, AES et SEC, ainsi que le nombre d'opérations nécessaires pour générer toutes les combinaisons de bits formant les clés pour chaque algorithme. Ceci donnera une idée sur la complexité de l'attaque exhaustive appliquée contre chacun des algorithmes où, il se voit clairement que l'attaque exhaustive menée contre nos algorithmes est la plus complexe.

	Taille de clé (bits)	**Nombre d'opérations**
DES	56	2^{56}
3DES	128	2^{128}
AES	256	2^{256}
SEC	240	2^{240}
PosESecL1	352	2^{352}
PosESecL2	1408	2^{1408}
OEEA	1393	2^{1393}

Tableau III.9. *Complexité de l'attaque exhaustive.*

III.5.6 Analyse de l'espace des clés

Ici, nous testons la sensibilité du processus cryptographique proposé aux clés utilisées. Dans notre cas, la clé générée est une clé calculée à partir de la donnée originale et de la donnée chiffrée correspondante, donc elle change d'une instanciation du problème à une autre. Ainsi, le cryptage successif d'une même donnée donne lieu à un ensemble de données chiffrées différentes ce qui entraînera, à chaque fois, la génération d'une clé de session différente pour le chiffrement d'une même donnée originale. Cela dotera notre méthode de cryptage d'une très grande sensibilité aux clés puisque toute clé interceptée d'une manière illégale ne servira que pour le déchiffrement d'une seule version chiffrée d'une même donnée donc elle ne sera plus utile par la suite.

Le problème rencontré dans le cas des crypto-systèmes symétriques, et par conséquent dans le cas de nos algorithmes développés qui sont des algorithmes symétriques, est la communication de la clé secrète au destinataire en vue de l'utiliser dans l'extraction de l'information originale (le déchiffrement). Pour ce faire, certains concepteurs ont proposé de combiner les fonctionnalités de la cryptographie symétrique et asymétrique. C'est le cas du PGP par exemple. Il crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ; puis, il crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire. Ainsi, le même principe peut être adopté pour sécuriser le transfert de nos clés en leurs chiffrant par un crypto-système asymétrique. Dans ce cas un schéma hybride de transmission sécurisée peut être envisagé comme le montre la figure suivante :

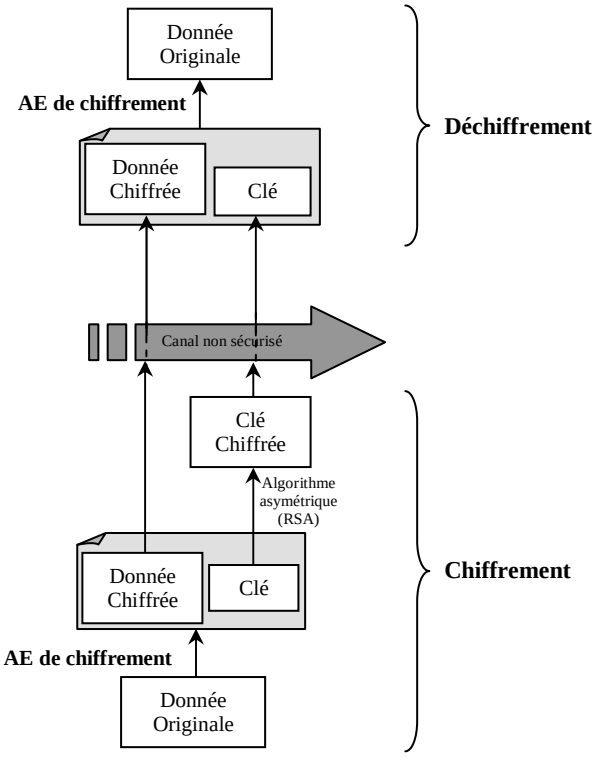


Figure III.34. *Schéma hybride de transmission sécurisée.*

Or, pour nos algorithmes PosESecL1 et PosESecL2, il est clair que la taille de la clé dépend strictement de la taille de la donnée. Le problème qui se pose lors du chiffrement d'une donnée de grande taille, c'est que la taille des clés générées elle aussi devient grande ce qui consommera un grand temps de chiffrement de clés du fait que les crypto-systèmes publiques sont déjà lents. Le problème s'aggrave de plus en plus lorsque la taille de la clé

devient égale ou dépasse la taille du codage de la donnée en termes de bits. Dans ce cas, l'utilisation de ces deux algorithmes devient inutile et il sera plus intéressant d'utiliser un crypto-système publique pour chiffrer la donnée au lieu de la clé générée malgré que la sécurité de tels crypto-systèmes soit grandement affectée par les choix paramétriques (p et q dans le cas de RSA [Zimm, 2005]). Pour remédier à cette anomalie, nous proposons de coder (ou bien comprimer) la clé générée par un système de codage sans perte, tel que le codage de Huffman par exemple [Huff, 1952], en vue de réduire sa taille avant de la chiffrée par un crypto-système asymétrique, puis d'envoyer au destinataire le package englobant la donnée chiffrée et la clé codée chiffrée. Lors de la réception, on procède inversement. On déchiffre la clé codée en utilisant la clé publique du destinataire, puis on la décode pour obtenir la clé proprement générée par PosESecL1 ou PosESecL2 qui va permettre de reproduire la donnée originale. Ainsi, le schéma du processus de sécurisation englobant cette dernière fonctionnalité sera le suivant :

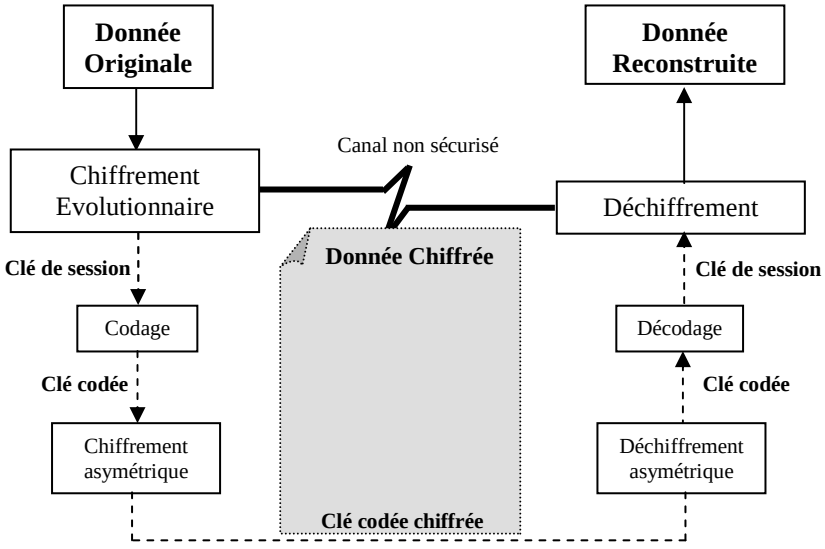


Figure III.35. *Schéma général du processus de chiffrement et de transmission sécurisée de la clé générée par chiffrement publique.*

Une deuxième solution de transmission sécurisée de la clé de session générée peut être envisagée dans le cas de manipulation de données images. Il s'agit de la technique de tatouage. Après avoir codé la clé générée par notre algorithme de la même manière proposée dans le cas de la première solution de transmission sécurisée de la clé (en utilisant un codage de Huffman par exemple), nous proposons, ici, de marquer l'image chiffrée par la clé de session codée. Ainsi, le schéma du processus de sécurisation englobant cette deuxième solution de transmission de la clé peut être résumée par le schéma de la figure III.36.

Cette deuxième solution de transmission est conditionnée par la taille de la clé codée pour satisfaire le compromis Robustesse-Capacité-Invisibilité lors du tatouage [Katz, 2000], [Barn, 2004], [Koba, 1990].

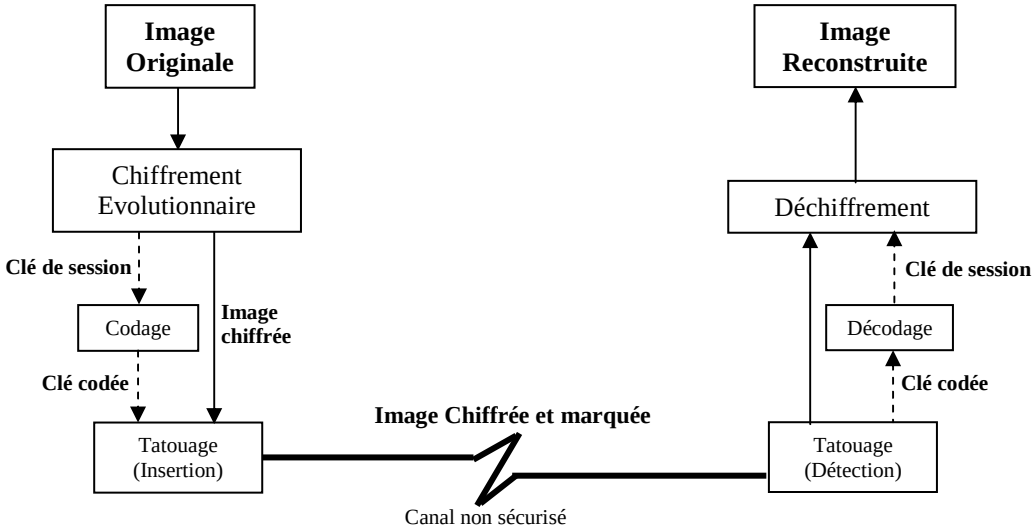


Figure III.36. *Schéma général du processus de chiffrement d'images et de transmission sécurisée de la clé générée par tatouage.*

III.6. Conclusion

À travers le présent travail, nous avons démontré que les algorithmes évolutionnaires, inspirés fondamentalement de la sélection naturelle des espèces, peuvent être également appliqués au cryptage de données. En effet, avec ces algorithmes, nous avons ramené le problème de chiffrement en un problème d'optimisation.

L'application de ces algorithmes requiert le réglage d'un certain nombre de paramètres pour la phase de chiffrement. La taille de la population initiale, le taux d'exploration et le nombre de générations affectent le temps de convergence des algorithmes, quant aux probabilités *Pc* , *Pm*, elles possèdent, théoriquement, une influence directe sur la qualité du résultat. Or, dans notre problème, les opérateurs génétiques (croisement et mutation) n'agissent pas sur l'information traitée (caractères du texte ou pixels de l'image) mais sur leurs coordonnées ; ils sont utilisés comme un outil pour explorer le reste de la population. De ce fait, les paramètres P_c , P_m affectent également la convergence des algorithmes.

Dans la pratique, les paramètres des algorithmes évolutionnaires sont réglés approximativement par tâtonnement [Gold, 1989]. Suite à une série de tests sur diverses données et en utilisant les fonctions fitness proposées, nous avons retenu certaines valeurs qui permettent d'avoir un résultat exploitable (donnée chiffrée) en un temps de calcul variant d'un algorithme à l'autre.

Les résultats obtenus montrent que les schémas proposés présentent des aptitudes dans la confusion et dans la sensibilité à la donnée originale qui les rend loin des attaques différentielles. De même, et pour pénaliser ou plus ou moins compliquer la tâche de l'attaque exhaustive, une deuxième alternative (PosESecL2) a été proposée en augmentant de trois ou de quatre fois la taille de la clé générée par la première alternative (PosESecL1). Une troisième alternative de cryptage robuste et sûre a été également définie. Il s'agit d'OEEA opérant suivant un mode de chiffrement (chiffrement à base d'occurrences) inexploitable par aucune attaque avancée.

Et pour résumer l'efficacité des deux modes de cryptage exploités, le cryptage à base de positions et le cryptage à base d'occurrences, le plus correcte est de dire que le deuxième

mode est plus important que le premier. En effet, ce mode est bâti autour de la notion d'occurrences, chose qui ne peut en aucun cas guider ou être exploitée par un cryptanalyste puisqu'elle ne présente aucune information utile pour aucune attaque possible. Ce qui n'est pas le cas pour le premier mode de cryptage à base de positions où une recherche exhaustive finira, un jour, par retrouver la bonne combinaison de positions d'éléments d'une donnée et ainsi de remettre en cause l'algorithme de cryptage utilisé.

Ainsi c'est les qualités du mode de chiffrement à base d'occurrences qui nous a motivées à le tester à travers d'autres métaheuristiques qu'elles soient à population ou à une seule solution. D'ailleurs le prochain chapitre présente une métaheuristique de colonie de fourmis traitant le problème de cryptage en opérant suivant ce mode de cryptage proposé.

Finalement, il sera utile de rappeler que l'objectif en cryptage est d'assurer la sécurité des données pendant leur durée de validité, donc, toute attaque tardive n'aura aucune importance.

CHAPITRE IV

CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

IV.1. Introduction

L'auto-organisation est un phénomène décrit dans plusieurs disciplines, notamment en biologie dans la branche qu'est l'éthologie (étude des comportements des espèces animales dans leurs environnements). Une définition a été proposée par Deneubourg [Dene, 1977] **:** *«l'auto-organisation est un processus dans lequel, un modèle de niveau global émerge uniquement, d'un grand nombre d'interactions entre les composants de bas niveau du système. De plus, les règles spécifiant les interactions entre ces composants sont suivies en utilisant uniquement des informations locales, sans références au modèle global ».*

Autrement dit, l'auto-organisation explique l'émergence d'un comportement collectif macroscopique par des interactions simples au niveau microscopique. L'auto-organisation n'exclut pas la complexité au niveau individuel. Elle suppose simplement qu'à un certain niveau, les individus se comportent comme des entités simples [Monm, 2000].

Depuis longtemps déjà, des comportements auto-organisés ont été découverts dans la nature. Par exemple, dans une colonie de fourmis, on peut observer différentes castes spécialisées dans un certain nombre de tâches : élevage de couvain, recherche de nourriture, construction de nid, etc. Aussi, le nid est construit sans que les insectes soient dirigés, ils répondent à un certain nombre de stimuli provenant de leur environnement.

D'importantes recherches ont eu pour intérêt principal l'étude de ces comportements intelligents afin de savoir comment ces populations interagissaient, accomplissaient des tâches et évoluaient. Ces recherches ont abouti à des modélisations inspirées du principe qu'un groupe d'entités plutôt simples et obéissant à des règles locales de coordination, sont capables d'engendrer des comportements globaux beaucoup plus complexes. Ces modélisations caractérisent ce que l'on dénomme « l'intelligence collective » [Bona, 1994], [Monm, 2000].

Dans ce chapitre, nous nous intéressons à l'algorithme de colonies de fourmis. Nous l'avons étudié, adapté et appliqué au problème de cryptage de données texte ou images. Le vocabulaire employé par cet algorithme est directement calqué sur celui de l'éthologie, nous parlerons donc, de fourmis, de connaissance collective, de phéromone, d'évaporation, etc.

Dans la première section de ce chapitre, nous présentons notre approche utilisée pour la conception du système de chiffrement dont l'objectif principal est l'obtention d'un système de

chiffrement symétrique non déterministe dont la clé secrète est générée pendant l'application de l'algorithme ; donc elle n'est pas connue à l'avance ce qui rend difficile le travail du cryptanalyste. Pour ce faire et étant donné que dans le précédent chapitre le problème de cryptage a été ramené en un problème d'optimisation, il a fallu, tout d'abord, coder le problème d'une manière adéquate permettant, ainsi, d'effectuer les opérations propres aux fourmis tel que : le dépôt, l'incrémentation et l'évaporation de phéromone, etc. Ainsi, le codage adopté est celui validé par la première métaheuristique d'AEs testée qui est le codage à base d'occurrences. Ce dernier permet la concrétisation des déplacements de fourmis par exploitation de la notion de permutation.

Ensuite, nous avons bâti notre algorithme en définissant soigneusement : la fonction d'évaluation cherchant à maximiser la différence entre l'information initiale et celle codée, le mécanisme de sélection adopté pour le déplacement des fourmis, le dépôt et l'incrémentation de phéromone, l'ensemble de la connaissance collective, etc.

IV.2. Motivation

Les algorithmes de colonies de fourmis forment une classe des métaheuristiques récemment proposée pour les problèmes d'optimisation difficile. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. Une colonie de fourmis communiquent indirectement via des modifications dynamiques de leur environnement (les pistes de phéromone) et construisent ainsi une solution à un problème, en s'appuyant sur leur expérience collective. C'est d'ailleurs cette dernière notion que nous cherchons à exploiter à travers l'utilisation de la métaheuristique de colonies de fourmis pour la résolution du problème de cryptage.

Ainsi, les objectifs que nous nous sommes fixés consistent à comprendre et isoler les mécanismes intéressants de cette métaheuristique, utiliser l'approche de chiffrement à base d'occurrences validée dans le précédent chapitre, suivant cette optique et appliquer l'algorithme ainsi conçu au problème de chiffrement de données texte ou images.

D'une manière générale, notre objectif est de développer un algorithme de chiffrement à base de colonies de fourmis par du constat simple que les colonies de fourmis résolvent des problèmes complexes, bien que l'intelligence d'une fourmi soit limitée ce qui est équivalent au fait de dire que, l'intelligence du système entier est plus grande que celle de la simple somme de ses parties.

IV.3. Algorithme proposé

Afin de bien comprendre notre algorithme de chiffrement proposé, AntCrypt, une description complète des différentes étapes de l'algorithme profondément inspirées de la métaheuristique *ACO (Ant Colony Optimisation)*, sera présentée. Des retouches ont été apportées sur l'algorithme de base pour l'adapter au problème étudié qui est celui de cryptage. Ainsi, le schéma de l'algorithme adopté est le suivant :

Algorithme IV.1 *AntCrypt*

Entrées
Codage de la donnée originale
Ensemble de paramètres (*m* : taille de la population, ρ taux d'évaporation)
DEBUT
1) Initialiser la connaissance collective // connaissance collective $\leftarrow \Phi$;
2) Créer la population initiale comportant *m* solutions initiales (*m* : nombre de fourmis) ;
Répéter
3) Construire les solutions;
4) Enrichir la connaissance collective ;
5) Evaluation des solutions de la connaissance collective ;
6) Sélection ;
7) Mise à jour de phéromone ;
Jusqu'à Satisfaction du critère d'arrêt
Retourner la solution trouvée ;
FIN

IV.3.1. Codage adopté

Pour prouver encore une fois l'efficacité de l'approche proposée et précédemment présentée exploitant un chiffrement bâti autour de la notion de nombres d'occurrences d'éléments de la donnée à chiffrer, le codage adopté par AntCrypt sera le même que celui d'OEEA.

La figure IV.1 rappelle le codage de données à utiliser.

$O(e_1)$	$O(e_2)$	.....	$O(e_i)$	.....	$O(e_{l-1})$	$O(e_l)$

Figure IV.1. *Codage d'une solution sous AntCrypt.*

Où : $O(e_i)$ est le nombre d'occurrences de l'élément e_i dans la donnée, l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Ainsi, notre algorithme AntCrypt cherche à changer itérativement la répartition des nombres d'occurrences $O(e_i)$ sur les différents éléments d'une certaine donnée avec génération aléatoire de positions dans la solution afin de créer le maximum de désordre dans leurs positions. Cela donne, aussi, l'avantage de produire des éléments (caractères/pixels) inexistants dans la donnée initiale.

IV.3.2. Création de la solution initiale

Initialement la connaissance collective (le savoir partagé) est vide. Dans cette étape on crée la population initiale comportant *m* solutions (*m* : le nombre de fourmis).

Nous désignons par *DonneeInitiale* le codage de la donnée originale. Donc, il est représenté par un vecteur dont les éléments contiennent les nombres d'occurrences des 1393 caractères (respectivement des 256 valeurs possibles de chacune des composantes R, V et B codant les pixels) dans le texte (respectivement dans l'image). Ces éléments sont notés :

$$O(e_1), O(e_2), ..., O(e_n) / n = \begin{cases} 1393 : Texte \\ 768 : \text{Im}age \end{cases}$$

Nous générons ensuite *m* fourmis susceptibles de créer, chacune, une nouvelle solution. Pour ce faire, chacune des fourmis est positionnée aléatoirement sur un élément de *DonneeInitiale*. Nous avons choisi de noter les différentes positions courantes par *PosCourante*. Ensuite, chaque fourmi va se déplacer un certain nombre de fois vers d'autres éléments (d'autres nœuds) notés chacun *PosSuivante* par application de la roulette aléatoire, et elle permute lors de chaque déplacement le nombre d'occurrence de l'élément « *DonneeInitiale[PosCourante]* » avec celui de l'élément « *DonneeInitiale[PosSuivante]* ». Une fois une fourmi s'arrête de se déplacer, nous obtenons une solution qui sera ajoutée à l'ensemble de connaissance collective. Alors à la fin de cette étape, la connaissance collective sera augmentée de *m* nouvelles solutions (chaque solution est obtenue par une fourmi).

IV.3.3. Construction de solutions

Après l'étape d'initialisation et dans le but de construire de nouvelles solutions sur une itération donnée de l'algorithme, les fourmis vont choisir à partir de la connaissance collective cette fois-ci leurs points de départ pour construire de nouvelles solutions. Ces points représentent les solutions ayant les plus grandes quantités de phéromone, c.-à-d. celles qui perturbent le plus la donnée initiale par rapport à une version cryptée représentée par la solution choisie. Donc le mécanisme permettant la construction d'une solution est le suivant :

1) Choisir les *m* meilleures solutions de la connaissance collective ;
Pour i = 1 : *m* ***faire***
    2) Choisir une solution *Sol* parmi les *m* solutions choisies en (1) ;
    Pour j = 1 : NbrMaxDép ***faire***
        3) Positionner aléatoirement la fourmi sur un élément *Sol[PosCourante]*;
        4) Déplacer la fourmi suivant la méthode de roulette aléatoire vers une autre position *PosSuivante* ;
        5) Permuter les deux éléments *Sol[PosCourante]* et *Sol[PosSuivante]* ;
    Fin pour
    6) Ajouter *Sol*, la solution qui vient d'être construite, à la connaissance collective.
Fin pour

IV.3.4. Evaluation

La fonction d'évaluation (ou d'adaptation) quantifie la qualité des solutions calculées. Celle que nous avons définie pour associer des valeurs d'adaptation à nos solutions calculées est la suivante :

$$F(Sol_j) = \sum_{i=1}^{n} \left| O_j(e_i) - O_0(e_i) \right| \quad \text{(IV.1)}$$

Tels que :

Sol_j : Solution j,

n : Nombre des éléments du vecteur *DonneeInitiale*,

$O_j(e_i)$: Nombre d'occurrences de l'élément (caractère / pixel) i dans la solution j,

$O_0(e_i)$: Nombre de répétitions de l'élément (caractère / pixel) i dans la donnée initiale.

IV.3.5. Sélection

Le rôle de la sélection est de distinguer les solutions sur la base de leur qualité, en particulier, pour permettre aux meilleures solutions d'une génération donnée d'être copiées dans la génération suivante.

Dans notre cas, lors du passage d'une génération g à la génération qui suit (g+1), nous favorisons les m meilleures solutions parmi celles de la connaissance collective globale y compris celles construites durant la génération g.

IV.3.6. Manipulation de phéromone

IV.3.6.1. Incrémentation de phéromone

Suivant la fonction F donnée en (IV.1), nous devons choisir la quantité de phéromone déposée pour chaque solution. Le mode de dépôt sélectionné peut fortement modifier le mode de convergence de l'algorithme. On pourrait choisir de déposer la même quantité à chaque solution. Cependant, on peut aussi décider de donner une puissance plus forte aux phéromones déposées sur les solutions améliorant l'altitude. C'est cette seconde solution qui a été appliquée dans notre algorithme.

Pour chaque solution *Sol*, nous calculons leur efficacité selon la formule (IV.1). Si cette solution existe déjà dans la connaissance collective on incrémente leur quantité de phéromone comme suit sans la réinsérer :

$$Phéromone(Sol_i) = Phéromone(Sol_i) + \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.2)}$$

Tel que EffMax est l'efficacité de l'une des meilleures solutions calculées de manière déterministe en permutant dans un ordre choisi le $i^{ème}$ élément avec le meilleur des l-i éléments restants (en maximisant la différence entre les deux éléments en question).

Sinon, si la solution n'existe pas, elle sera ajoutée à la connaissance collective en lui attribuant une quantité de phéromone selon la formule suivante :

$$Phéromone(Sol_i) = \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.3)}$$

IV.3.6.2. Évaporation de phéromone

Comme dans la nature, les phéromones sont des substances chimiques qui s'évaporent au fil du temps. Donc, il sera nécessaire que l'algorithme imitant ce coté de la nature représente ce phénomène afin d'éviter aux fourmis de converger vers un maximum local.

$$Phéromone(Sol_i) = Phéromone(Sol_i) \times (1 - \rho) \qquad \text{(IV.4)}$$

Où ρ est le taux d'évaporation.

L'évaporation de phéromone de toutes les solutions après une période de temps permettra de redonner de l'intérêt à d'autres solutions qui peuvent nous amener à la meilleure solution. Le taux d'évaporation ρ doit être bien choisi, puisque s'il est très faible, les phéromones s'accumulent et atteignent la borne max et nous risquons alors de s'enfermer dans un maximum local, et s'il est trop grand, les phéromones atteignent rapidement la borne min et beaucoup de solutions n'auront pas la chance d'être choisies par les fourmis.

Dans notre application, ce taux doit aussi dépendre des valeurs des éléments du codage. Si la différence entre deux éléments est grande, c.à.d. la fonction F va prendre de grandes valeurs correspondantes à de grandes quantités de phéromones, donc le taux d'évaporation peut prendre une valeur importante. De même, il dépend du nombre de fourmis travaillant sur une même génération : beaucoup de fourmis dans une génération augmentent la possibilité d'avoir deux solutions identiques, c.-à-d. la phéromone sera incrémentée plusieurs fois de suite, alors le taux d'évaporation sera important.

Ainsi et après un certain nombre de générations fixé expérimentalement, nous avons varié un pourcentage d'évaporation de 0% jusqu'à 0.5% en s'inspirant, au début, de l'application de l'algorithme ACO pour la résolution du problème de TSP [Barg, 2005]. Ce choix a été, ensuite, validé par expérimentation.

IV.3.7. Critère d'arrêt

Après un certain nombre de générations fixé expérimentalement, l'algorithme converge vers la meilleure solution trouvée. Reste de générer ou de reconstituer la donnée cryptée à partir du codage de la solution proposée. Ceci correspond à l'opération inverse de l'opération de codage. Il s'agit d'une opération de décodage qui procède comme suit : à partir du codage de la solution proposée qui est un vecteur de nombres d'occurrences des éléments de la donnée chiffrée, pour chaque élément *i* du vecteur *Sol*, on génère *Sol[i]* positions aléatoires du caractère/pixel correspondant à l'élément *i* dans la donnée cryptée.

IV.3.8. Déchiffrement

Nous arrivons maintenant au processus inverse qui permet de rendre la donnée à nouveau intelligible sans aucune perte d'information ; il s'agit du processus de déchiffrement.

Notre algorithme proposé est un algorithme symétrique, donc la clé générée doit être maintenue secrète. Cette clé s'obtient au fur et à mesure du calcul de l'information chiffrée au fil des générations. Sa valeur finale correspond aux permutations des positions des nombres d'occurrences des éléments de la donnée chiffrée pour obtenir celles des nombres d'occurrences des éléments de la donnée en clair. Le processus de déchiffrement consiste donc à replacer les éléments dans leurs positions initiales suite à l'introduction de la bonne clé.

IV.3.9. Réglage des paramètres et résultats

Cette section est consacrée d'une part, à l'optimisation ou la fixation du jeu paramétrique et d'autre part, à l'application de l'algorithme proposé *AntCrypt* sur les mêmes données de test utilisées pour l'évaluation des AEs proposés et présentés dans le précédent chapitre. Ils s'agissent des données faisant l'objet de la figure III.1. Dans un premier temps, nous allons exposer notre stratégie pour établir et fixer le choix paramétrique de notre algorithme. Puis nous allons présenter les résultats d'application d'*AntCrypt* sur les données modèles (textes et images), à savoir la donnée chiffrée, la clé générée, la quantité de phéromone, l'efficacité, le temps de chiffrement et de déchiffrement.

IV.3.9.1. Réglage des paramètres

Les figures IV.2 et IV.3 présentent les résultats des différents tests effectués en termes d'efficacité et de temps de chiffrement. Ces valeurs représentent la moyenne de dix (10) exécutions successives appliquées sur l'image de test Lena pour différentes valeurs de nombre de générations et de fourmis.

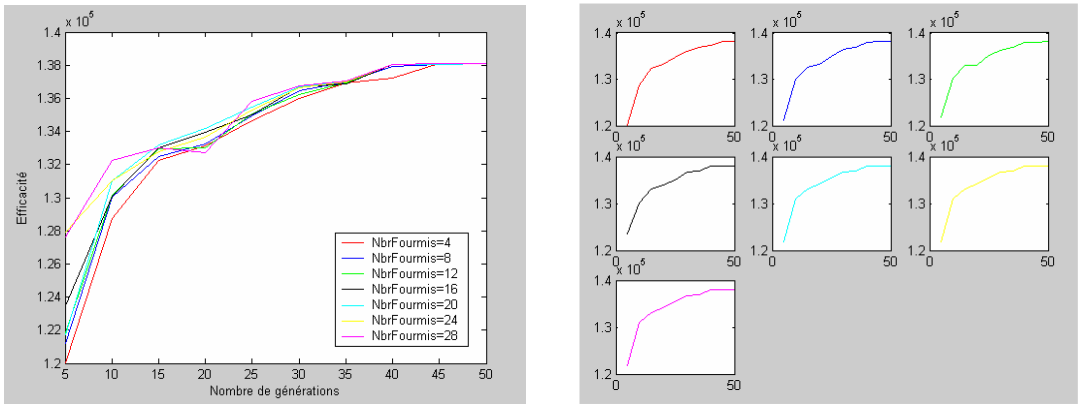


Figure IV.2. *Influence du nombre de générations et du nombre de fourmis sur l'efficacité.*

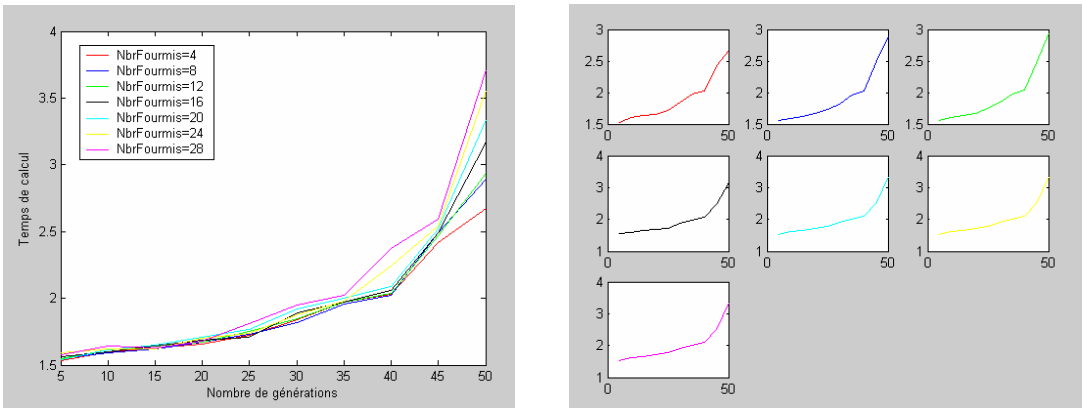


Figure IV.3. *Influence du nombre de générations et du nombre de fourmis sur le temps de calcul.*

D'après les résultats résumés à travers les figures IV.2 et IV.3, nous constatons que la majorité des résultats des tests ont des valeurs d'efficacité assez proches. La meilleure valeur d'efficacité est obtenue au bout de 50 générations à 20 fourmis (Efficacité = 138077) mais avec un temps de chiffrement assez important par rapport à l'ensemble de test (3,17 s). Cependant, le meilleur temps de chiffrement est obtenu pour le test qui a été déroulé sur 5 générations et 4 fourmis (1,53 s) mais malheureusement avec la plus faible valeur d'efficacité (Efficacité = 120009).

Les tests déroulés sur 45 générations avec 4, 20, 24 ou 28 fourmis, et 50 générations avec 4, 8, 12, 20, 24 ou 28 fourmis ont donné de très bonnes valeurs d'efficacité (138073, 138070, 138073 et 138074, 138075, 138074, 138076, 138073, 138072, 138075 respectivement) mais après de longs temps de calcul (2.42 s, 2.52 s, 2.54 s, 2.59 s et 2.67 s, 2.94 s, 3.34 s, 3.56 s et 3.71 s).

D'autres tests déroulés sur 5 et 10 générations ont montré, cette fois-ci, de bons temps de calcul (compris entre 1.53 s et 1.64 s) mais de mauvaises valeurs d'efficacité (comprises entre 120009 et 132263).

Donc, notre objectif sera de réaliser un compromis entre la valeur d'efficacité et le temps de calcul. En effet, le test déroulé sur 40 générations avec 12 fourmis semble le plus satisfaisant. Il permet d'atteindre une bonne valeur d'efficacité (Efficacité = 138063) en un temps de calcul très raisonnable (2.04 s). Ca sera le paramétrage à adopter pour *AntCrypt*.

IV.3.9.2. Résultats

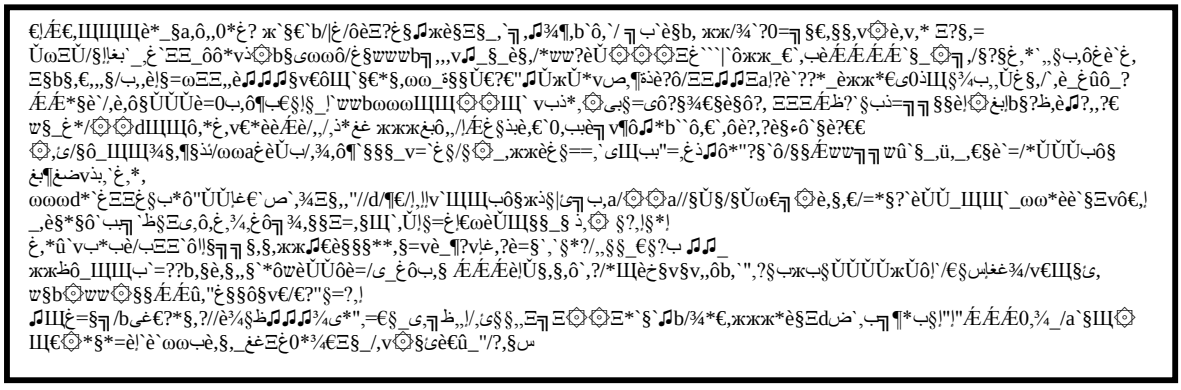
Ci-dessous, nous présentons les résultats d'application d'*AntCrypt* sur les mêmes données modèles utilisées pour tester nos trois algorithmes présentés dans le précédent chapitre : Texte1, Texte2, image Lena et image Logo suivant le paramétrage fixé dans la section précédente (résumé à travers le tableau IV.1). Toutefois, et pour démontrer l'aspect non déterministe de notre approche de chiffrement par exploitation de métaheuristiques, dont l'ACO fera l'objet ici, nous présentons, cette fois-ci, pour chaque donnée originale deux versions chiffrées en deux instances différentes accompagnées des clés de chiffrement générées dans chaque cas.

	Valeurs des paramètres d'*AntCrypt*
Nombre de fourmis	12
Nombre de générations	40

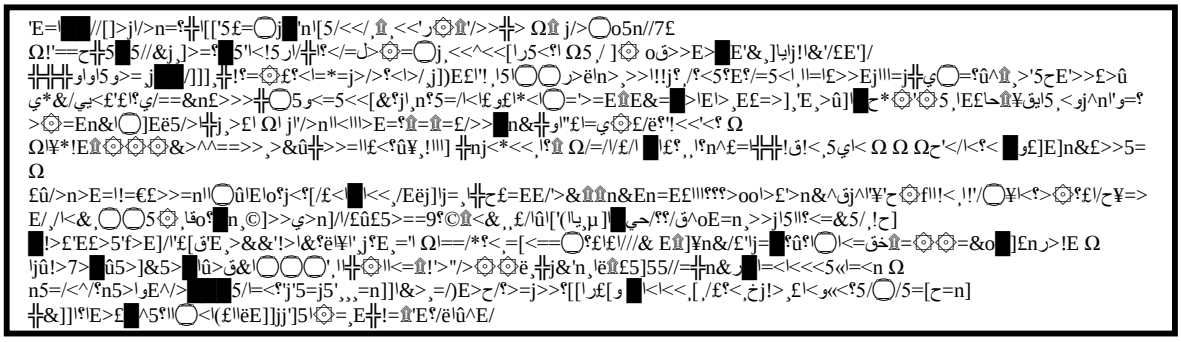
Tableau IV.1. *Valeurs adoptées pour les paramètres de AntCrypt.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure IV.4. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets) :

124	398	407	770	921	266	896	19	794	90	918	245
843	914	614	547	809	904	560	80	375	908	879	886
901	816	830	543	573	900	156	196	98	254	906	314
230	443	917	497	903	418	288	907	289	680	147	861
820	642	902	259	916	840	30	898	905	102	822	635
317	161	920	465	292	458	546	91	911	246	910	833
111	912	155	193	640	919	923	643	357	681	121	725
915	892	581	899	922	216	909	913	789	549	179	578
28	552	3	650	1070	379	850	996	934	933	1104	285
459	997	1124	1242	1026	897	1171	1075	8	521	64	281
1017	955	985	872	262	485	774	16	85	709	279	97
628	120	374	87	286	1112	277	862	518	47	255	1097
1203	45	163	60	599	12	461	5	848	1240	92	1032

331	174	888	83	1073	20	676	342	494	280	422	487
607	1020	958	61	1154	601	95	1100	112	386	1238	1196
1185	1119	393	669	1	756	1172	1012	1102	274	394	363
237	755	267	6	219	389	893	187	1225	437	175	683
834	659	1215	1085	1138	819	447	688	436	764	1158	117
206	1024	675	695	264	319	345	76	127	1183	484	1006
1191	540	1205	210	384	689	655	991	71	928	1130	416
1129	324	761	1134	1089	1029	449	1200	704	231	1013	466
474	1166	813	302	506	1050	490	1204	1092	382	194	744
1066	682	164	72	630	972	15	1213	738	309	1199	531
1149	1084	1202	971	535	395	1031	731	988	737	41	1188
1142	584	336	839	113	818	1245	306	723	657	620	403
252	858	994	423	1036	563	1217	1137	135	108	1133	1088
1160	810	829	946	1058	1068	952	944	751	673	1110	79
1232	835	733	557	1193	982	1179	298	1052	1101	198	1144
225	232	949	400	229	981	545	846	954	432	778	411
634	1161	1042	457	18	250	68	1123	866	462	1243	1051
1120	768	960	990	1189	1182	491	827	185	1145	1148	714
1115	428	574	995	1067	653	1091	978	27	415	132	940
145	409	665	1234	956	622	727	1227	786	402	1009	859
1236	1224	870	1039	387	239	1028	565	1043	594	176	856
787	424	323	588	1033	715	945	1195	359	1162	1165	793
791	930	684	299	716	1076	605	101	1241	149	826	412
1181	480	364	1187	1004	14	1226	579	438	1222	542	613
125	664	1176	197	202	1208	593	520	183	1207	248	604
434	1139	1127	1014	211	891	169	747	968	55	1190	339
1178	625	758	698	330	1038	446	425	611	668	646	1170
1174	144	166	283	1244	857	1231	1131	366	48	31	34
1107	52	814	315	24	414	983	1135	860	11	452	937
539	337	812	935	639	507	863	378	637	301	1055	1219
1077	1198	440	943	451	658	478	361	392	82	103	586
693	881	1210	652	1235	37	1001	970	1156	1175	808	184
278	662	799	341	1041	244	736	1030	109	730	984	195
1019	678	728	420	294	1212	796	1218	303	1192	627	645
1079	1109	1046	481	272	488	1116	1096	222	238	21	349
1034	2	1128	1011	130	157	445	167	705	1087	1201	33
470	489	10	598	453	128	39	1061	817	1209	1216	741
926	1184	1008	942	524	969	372	308	118	228	473	815
312	623	1122	516	1078	962	1143	929	1233	989	316	602
941	1125	1223	648	1072	1136	58	1062	224	931	571	1173
987	649	158	1040	612	998	233	864	241	140	9	621
177	88	1047	321	57	1000	1037	1117	340	1151	961	1221
595	973	138	152	959	483	213	479	207	477	191	287
740	522	977	654	284	227	332	141	234	105	842	305
1080	957	572	355	310	1140	660	258	78	322	831	1054
201	1229	467	1045	273	353	629	441	618	63	1074	849
527	924	43	173	710	295	948	708	1057	876	606	36
967	1025	1081	1147	1132	868	979	1082	966	844	1237	1010
1239	845	772	51	391	1168	855	865	823	65	62	874
275	276	159	885	182	1105	1118	1002	603	592	999	759
1035	762	1069	1083	1007	1005	932	1094	824	745	110	828
154	677	638	703	851	204	1180	806	208	760	884	771
610	1230	25	1211	439	925	1022	765	1155	670	672	519
469	356	548	679	1056	1015	236	1167	964	803	1106	218
1194	591	663	587	260	50	69	1141	953	1071	975	701
212	396	550	307	1063	203	811	743	510	84	333	1153
739	566	807	1159	1018	162	1157	385	1023	35	1114	22
644	1108	261	1093	711	963	296	328	706	122	358	597
938	965	504	1197	249	788	1098	790	325	1214	1228	401
1027	525	1086	951	624	529	1099	992	1150	136	667	380
1126	1111	199	1048	1206	580	404	697	251	719	475	40
585	1044	1095	1053	686	350	575	1163	1003	1060	1064	223
950	146	986	1177	752	304	726	123	853	742	168	492
454	766	804	1021	533	247	46	939	42	1121	221	555
226	980	327	17	180	974	798	927	209	1152	692	554
370	126	151	129	596	377	800	1169	1146	408	463	417
569	1103	615	498	326	59	456	1186	1049	137	139	291
427	632	685	1113	433	1220	153	1164	722	354	189	976
523	100	763	509	1016	444	399	369	1065	936	825	220
360	265	947	1059	749	568	1090	838	852	631	365	993

329	268	651	1330	390	500	1331	1332	460	320	468	729
1333	1274	583	617	753	564	56	782	1306	661	1334	263
165	1335	1318	795	713	371	367	1336	134	1305	271	748
1337	81	532	1299	343	486	1338	1339	472	1264	1302	413
1340	362	769	801	633	1273	517	534	867	94	335	732
1247	1290	1316	1260	89	781	718	1341	450	1323	551	1342
1343	1344	894	1292	687	515	1345	608	1346	1347	1246	351
368	767	1248	570	430	1322	735	235	1348	1349	1284	1350
1288	1278	802	847	257	499	746	805	1267	114	1295	514
1351	188	785	558	1317	1352	784	70	537	832	1353	448
181	1354	297	99	44	442	217	609	501	410	1355	837
1356	66	1314	313	38	1357	777	541	559	1358	1320	426
170	1312	1319	1279	186	32	882	171	429	776	1359	1360
880	720	671	1275	406	215	431	1253	347	792	717	1361
200	1362	1313	1363	54	754	1251	476	511	1364	435	1325
1365	1291	338	1257	1315	699	1282	641	712	381	96	1327
482	883	1280	421	1366	513	1367	26	1368	1250	779	1276
1326	242	143	1307	300	1249	1369	508	23	721	690	1298
773	895	887	1270	875	53	576	67	589	455	890	636
626	1328	405	536	647	582	119	656	696	1370	282	290
334	619	1371	1372	512	503	1373	1374	854	544	877	1375
205	493	106	1308	783	77	577	293	1376	1254	344	1252
148	526	256	1377	464	616	214	878	1294	1303	1321	1378
1297	1379	1272	1311	1300	1266	240	590	1380	530	243	1287
567	1258	1293	160	495	707	150	1381	666	702	873	505
528	674	1382	562	694	86	889	397	1277	93	1296	1285
1256	104	131	270	1383	311	49	190	1281	841	388	734
1283	1265	346	775	115	600	1263	1255	1384	352	1310	172
1259	1309	561	869	13	691	142	133	871	373	821	836
269	502	75	1385	253	1386	116	318	1301	376	1387	556
348	74	178	1268	724	107	1324	553	7	1388	1286	192
1389	29	73	780	1329	750	538	797	471	1262	1390	419
757	1391	1261	1392	1304	1289	496	4	700	1269	383	1271
1393											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets) :

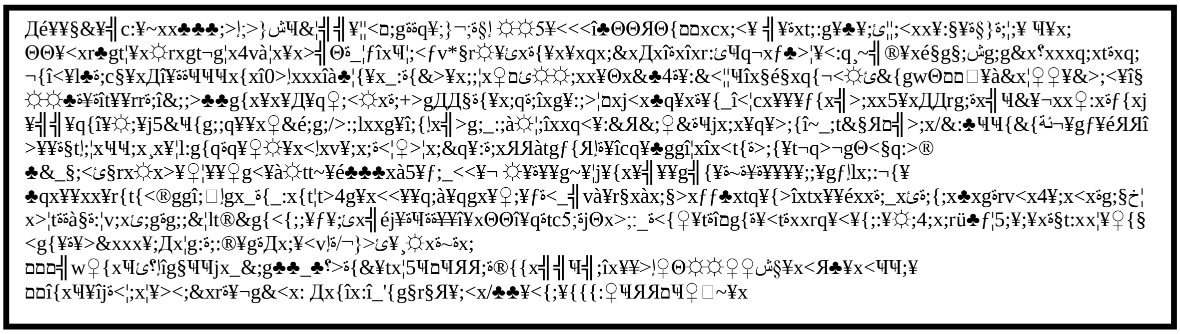
80	34	87	94	51	64	91	46	18	32	55	81
44	2	63	86	54	72	45	16	88	84	20	75
82	97	39	13	6	36	67	95	12	53	17	92
24	98	89	14	74	60	25	41	40	76	10	57
66	93	5	11	9	70	3	31	73	61	42	43
23	27	59	28	78	37	52	69	7	71	77	8
99	50	85	96	30	21	29	22	58	19	90	68
26	47	79	15	35	48	38	878	603	723	164	621
401	997	320	274	4	126	364	628	908	150	286	287
600	815	170	482	424	159	307	220	139	213	596	967
217	133	405	644	511	861	909	334	802	964	597	691
449	942	485	721	538	945	862	870	495	513	505	957
623	883	229	506	955	419	767	751	917	56	143	237
831	972	799	62	324	224	479	750	864	255	771	330
378	316	474	564	161	403	172	748	461	223	978	778
166	318	904	540	985	167	741	202	467	569	49	434
793	283	765	568	248	102	250	653	816	359	297	602
232	249	207	196	365	190	754	947	455	490	338	931
788	454	698	649	774	308	588	702	708	631	33	444
557	463	503	913	886	1	715	116	550	516	189	849
244	685	458	584	285	764	975	418	589	877	921	487
309	632	144	701	850	317	932	563	239	643	610	192
348	574	533	732	993	212	809	299	417	790	501	464
922	567	867	331	650	231	466	722	895	666	168	891
529	541	558	762	279	730	635	824	486	177	313	520
581	560	607	140	977	719	859	339	638	599	380	491
662	777	532	890	758	169	745	382	83	580	900	965
534	987	968	395	104	892	107	826	225	670	222	200
893	387	992	292	545	499	961	353	590	919	869	291
772	575	191	160	605	305	180	811	142	410	832	236
450	389	868	314	659	699	163	247	507	335	481	293
110	982	673	214	634	363	675	842	65	995	343	576
814	546	175	488	425	280	910	677	680	265	718	438
497	716	875	414	366	692	710	976	344	548	970	376

851	421	781	837	927	792	388	874	860	836	798	570
627	743	912	937	204	310	803	203	509	686	306	665
131	776	728	808	682	938	657	197	843	735	779	549
739	593	179	640	846	185	717	370	753	379	178	360
857	465	579	565	899	756	158	452	606	737	132	326
629	729	498	784	295	429	221	648	925	375	500	385
408	182	235	616	431	654	768	171	478	346	694	300
609	994	797	125	357	926	766	863	103	257	145	789
924	420	696	321	528	329	646	476	981	148	402	123
227	818	374	396	954	833	749	121	233	198	935	795
462	720	105	153	897	724	705	416	604	230	536	655
539	866	228	697	246	773	998	526	951	426	352	974
591	988	852	647	573	806	642	127	427	100	619	404
713	459	782	328	783	827	433	496	551	208	349	880
807	369	146	371	118	523	181	521	928	613	277	439
939	661	645	946	510	393	484	633	578	601	106	906
787	887	819	278	907	902	553	276	151	770	272	134
394	681	165	358	432	668	810	556	399	129	820	350
136	117	205	812	622	262	817	652	984	109	260	397
916	448	618	559	898	530	742	780	973	689	885	264
847	245	514	687	630	747	524	986	493	303	259	477
270	267	195	442	958	991	215	547	347	269	914	162
667	114	571	537	733	663	124	865	377	586	583	587
761	184	936	135	744	940	368	296	684	941	712	592
137	113	206	706	412	933	241	594	271	688	822	543
561	980	216	332	844	660	535	304	996	949	894	731
423	341	755	734	284	356	918	183	201	504	830	392
390	585	956	457	446	440	492	775	174	651	582	841
456	615	199	149	554	522	238	362	266	813	796	111
959	989	614	876	953	608	119	409	282	544	428	834
639	351	354	664	637	855	301	873	517	519	625	896
963	725	400	736	355	209	595	290	671	508	835	658
381	256	430	979	345	871	786	901	176	714	839	288
437	120	319	243	840	598	983	251	157	226	752	342
700	669	821	435	856	470	273	407	315	726	234	888
406	727	252	704	340	473	443	929	542	823	740	337
785	641	384	693	460	872	218	445	469	155	193	853
386	828	138	298	999	854	881	943	483	101	759	769
147	930	611	678	112	672	624	709	311	141	577	518
791	422	626	480	336	612	950	905	948	617	327	920
471	128	829	562	695	884	240	858	281	122	254	323
367	707	683	453	794	512	711	889	944	475	263	302
372	186	656	294	911	531	845	969	253	738	801	805
966	489	472	383	289	990	210	258	441	879	391	451
525	322	188	763	173	312	413	261	154	962	515	156
800	971	411	882	219	436	398	527	415	555	333	760
674	757	447	620	572	268	952	242	115	502	494	187
194	152	552	804	703	468	108	903	838	636	130	275
960	325	848	676	923	825	679	690	361	211	746	566
934	915	373	1097	1243	1324	1351	1271	1352	1040	1080	1241
1277	1136	1353	1089	1051	1354	1355	1039	1077	1342	1258	1320
1037	1064	1202	1242	1312	1281	1123	1054	1170	1267	1285	1222
1340	1349	1129	1111	1296	1174	1252	1345	1356	1166	1357	1115
1275	1328	1250	1128	1263	1171	1098	1284	1033	1231	1276	1005
1058	1313	1311	1059	1124	1232	1157	1013	1100	1132	1019	1099
1103	1023	1112	1225	1240	1304	1194	1215	1358	1247	1198	1334
1335	1122	1018	1210	1164	1341	1156	1127	1106	1006	1300	1323
1114	1359	1245	1113	1066	1154	1150	1036	1344	1289	1360	1238
1298	1034	1035	1093	1048	1008	1318	1253	1309	1028	1331	1361
1016	1195	1286	1362	1363	1050	1090	1038	1137	1119	1287	1079
1163	1182	1151	1125	1043	1303	1025	1325	1133	1042	1364	1002
1234	1075	1347	1088	1365	1212	1117	1230	1301	1294	1071	1214
1200	1307	1049	1264	1190	1060	1147	1299	1332	1160	1165	1366
1109	1180	1343	1305	1315	1065	1187	1055	1052	1191	1367	1110
1196	1368	1074	1205	1257	1262	1168	1336	1108	1308	1045	1369
1330	1278	1239	1134	1221	1149	1235	1104	1209	1370	1053	1265
1282	1107	1094	1233	1069	1153	1213	1244	1327	1266	1146	1229
1091	1141	1105	1346	1226	1041	1070	1186	1142	1056	1302	1063
1199	1321	1162	1193	1086	1101	1270	1083	1310	1017	1192	1177
1179	1095	1248	1184	1219	1121	1022	1273	1172	1371	1046	1372

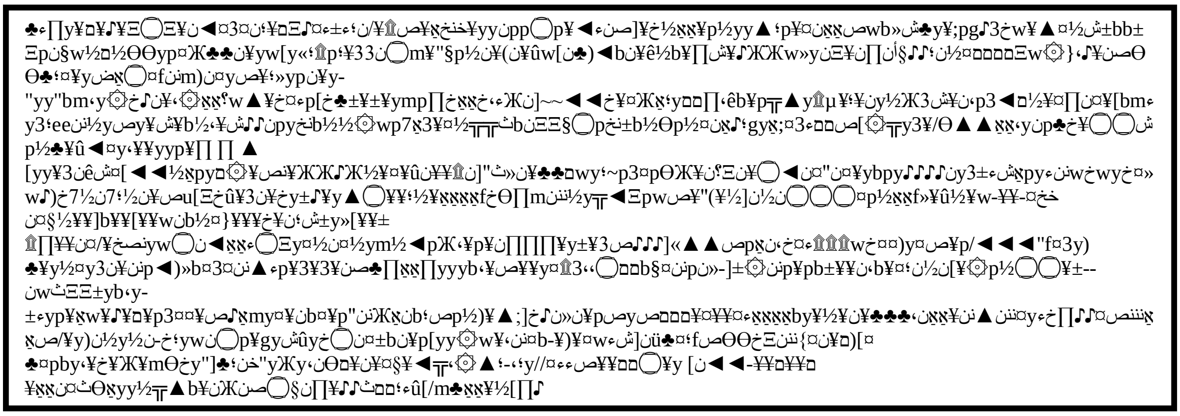
1044	1178	1096	1208	1224	1183	1003	1084	1012	1131	1130	1348
1062	1316	1159	1188	1021	1373	1009	1135	1374	1375	1203	1211
1228	1010	1087	1268	1376	1333	1081	1377	1322	1220	1145	1072
1057	1288	1061	1047	1249	1255	1120	1030	1306	1261	1189	1246
1314	1144	1378	1254	1379	1126	1295	1138	1176	1319	1152	1337
1216	1201	1236	1380	1293	1076	1185	1291	1158	1143	1381	1029
1297	1067	1259	1382	1140	1155	1068	1269	1031	1383	1384	1118
1339	1260	1290	1102	1385	1386	1082	1280	1167	1000	1078	1085
1073	1387	1197	1272	1007	1181	1227	1011	1001	1161	1206	1251
1116	1204	1388	1173	1032	1350	1024	1389	1207	1292	1217	1326
1237	1390	1223	1218	1148	1391	1392	1279	1274	1026	1027	1256
1092	1014	1015	1393	1317	1139	1338	1329	1169	1004	1175	1283
1020											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)



(b)



(c)

Figure IV.5. (a) *Donnée originale Texte2,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

700	340	911	855	661	6	451	910	17	3	151	267
937	143	936	914	475	880	745	874	939	897	155	935
281	730	209	938	173	638	777	191	329	930	384	421
673	131	485	462	512	259	736	644	457	728	932	593
933	65	920	924	926	918	349	196	232	309	553	782
917	369	919	927	455	878	255	921	913	929	156	931
912	759	400	922	616	915	594	691	243	934	592	916
819	361	128	923	486	24	925	159	748	301	928	140
411	379	1093	16	599	1169	1108	1079	1040	985	1021	141
799	1006	1067	1107	82	572	1014	325	977	498	9	943
974	670	394	805	270	49	178	658	808	604	392	1229
642	456	1058	80	1184	78	669	472	1045	1214	445	1011
195	492	242	952	389	1069	814	839	1060	591	983	895
285	248	67	947	1194	460	380	38	46	584	482	430
631	1236	478	1012	1004	297	570	1240	228	332	390	1127
1153	40	992	991	1151	625	44	1106	47	1189	99	823
986	706	395	85	559	205	28	641	887	197	319	979
1094	554	882	1066	822	1166	531	564	949	305	398	654
694	544	200	160	801	1017	189	1031	1083	1143	476	1196
1002	198	894	348	1052	619	69	214	1192	1099	120	1134
318	300	534	73	862	104	425	328	1092	1141	677	296
692	1212	466	612	1190	144	960	686	1188	53	404	668
683	381	106	672	523	1145	688	42	1053	342	102	779
292	757	1055	1183	852	514	34	890	702	525	1233	863
870	79	444	858	1241	851	904	1250	948	346	224	31
807	754	528	942	794	1140	511	1101	347	295	1198	289
436	408	587	308	269	403	247	1074	25	1148	27	264
194	489	1160	163	1056	1225	298	322	965	1202	563	237
1035	1179	941	56	1051	1057	1064	434	978	1015	532	413
602	1049	71	772	162	1123	483	1142	231	1152	844	1078
1235	1159	356	418	1047	1082	737	273	1137	676	1144	263
372	1039	530	1098	440	284	1191	1081	94	1205	29	783
1216	603	630	859	877	598	473	4	274	944	674	681
709	1154	565	1008	837	662	549	671	1003	215	710	1228
1243	758	1016	962	1238	175	1217	1158	518	57	1172	628
764	876	1223	606	7	253	657	968	1245	766	999	542
324	775	399	738	989	1097	1009	515	238	406	770	245
993	499	1167	30	217	230	1077	299	787	74	537	1029
784	802	645	76	1147	1197	415	452	1126	45	517	629
1227	1000	940	569	632	618	164	172	101	18	88	216
1135	950	397	377	879	809	1226	1176	527	227	1203	431
833	276	970	87	1027	256	480	843	409	637	286	1013
1248	142	725	605	743	1114	567	643	791	640	576	1090
1244	150	689	1018	650	589	311	516	995	1178	519	1175
611	828	621	857	1122	1020	1065	294	474	1230	959	376
869	501	898	370	545	1117	816	1071	1110	43	526	623
115	836	655	680	26	494	1195	447	1023	385	513	548
969	161	250	1185	1034	110	111	860	15	561	793	1170
961	820	133	432	1116	97	114	279	93	193	495	422
1246	1059	997	112	212	1210	1130	33	219	60	469	698
1231	174	812	479	763	1207	1218	277	50	450	416	41
954	582	116	488	127	1128	552	697	1209	1080	306	490
1085	1129	539	981	953	337	449	211	726	1061	1124	510
715	1091	861	83	1007	827	477	330	800	849	546	8
747	419	10	186	454	500	66	583	899	357	401	1001
971	659	856	226	1234	240	233	91	761	343	1046	1146
595	107	1186	1237	1103	1139	1076	1048	647	579	722	353
608	386	660	1115	596	1042	610	636	405	718	290	529
1138	234	433	1112	1164	964	945	742	967	773	20	77
1070	1221	1072	804	1224	663	1161	721	1120	1118	458	797
1193	707	1119	326	453	35	505	1030	533	126	1041	64
769	753	14	63	1024	1100	387	149	744	239	1200	703
1037	504	522	32	956	875	536	723	998	345	1171	435
1068	1155	366	832	1213	491	1028	1173	829	429	23	374
1181	976	11	973	393	1187	988	1232	135	1102	169	493
278	664	134	90	821	885	1163	762	741	627	108	892
1199	996	367	958	982	1165	417	1174	95	1150	557	1033
740	333	891	382	129	972	1247	1109	1121	266	204	1206

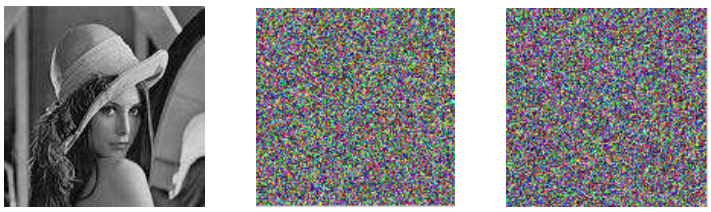
1010	388	187	966	1204	955	987	70	790	1239	503	352
957	262	1084	272	59	866	414	1168	1219	1201	1222	907
1111	317	963	1113	1026	497	487	818	810	109	442	439
190	780	423	896	130	363	1086	798	617	1050	1087	994
1105	208	1211	580	796	89	291	68	1025	251	1182	1005
682	1104	378	980	538	646	118	323	699	1157	359	1136
236	1156	1131	1075	1043	1149	853	96	229	351	179	1220
180	138	145	424	177	1019	1032	560	975	136	1132	1125
261	774	685	568	886	675	315	202	158	1036	168	1133
713	601	1215	600	176	1054	1208	817	865	678	739	864
789	1095	1162	509	1022	614	327	1249	946	607	806	760
185	1062	221	667	902	566	316	635	665	39	813	717
786	951	834	624	438	1088	58	1177	984	1044	62	719
1251	900	1096	100	1073	166	990	695	1180	463	2	334
124	1242	1038	123	148	889	615	121	1089	651	407	481
749	1063	1271	470	86	597	465	1332	1307	260	1342	1275
881	368	371	48	1343	1301	1270	1344	838	1313	1345	666
1346	132	265	871	302	840	84	502	842	1347	222	765
573	252	1348	1322	732	246	1349	835	1334	13	1319	1339
464	1350	778	771	1351	355	268	590	845	1311	210	223
1289	577	184	1282	1352	735	1269	1293	1353	1278	459	1263
241	207	1274	1354	652	1325	1298	446	12	588	1268	873
884	1355	622	1252	125	344	551	213	304	1356	1265	1357
1296	609	1330	908	693	98	5	868	555	1320	225	428
687	1331	383	541	1308	81	188	756	1267	558	154	1272
1358	1359	341	1258	52	847	467	1360	776	724	1361	22
1279	103	280	36	220	1362	795	634	578	461	571	484
181	303	321	690	1363	734	905	1285	574	19	733	412
396	313	402	1364	1365	331	139	550	51	1366	1281	365
335	746	1367	586	257	336	613	287	1253	803	883	792
653	850	72	585	310	831	1329	360	729	192	1368	218
426	21	496	727	1273	815	750	320	1326	1318	1369	1327
1370	1371	639	684	1262	358	1372	903	1373	1257	581	825
312	906	117	1374	1341	649	1375	767	1376	1264	556	1377
373	235	1378	826	1379	1336	731	872	1316	203	468	1304
1380	848	708	167	543	1291	1261	626	1256	888	712	206
1340	307	1294	854	535	338	1337	1315	1259	909	410	37
244	620	1277	443	1338	362	75	1287	471	867	153	768
575	1381	811	824	1314	171	1323	105	1254	1290	1328	271
1303	1266	846	183	441	1300	648	521	1324	1255	146	893
375	1382	1383	508	354	1283	696	54	254	282	283	656
785	788	448	841	1310	506	137	1317	1295	165	830	1333
339	1286	701	714	679	751	1299	540	704	249	147	182
1260	1384	364	1302	157	293	1306	1385	633	92	420	1309
427	288	716	1335	391	1321	55	258	1280	1288	201	61
1386	152	1387	1388	901	705	350	1389	752	1390	1	1391
720	711	755	113	562	1284	1292	1392	1297	314	437	547
1393	507	1305	520	1312	199	119	524	781	170	122	275
1276											

Clé de chiffrement de la deuxième version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

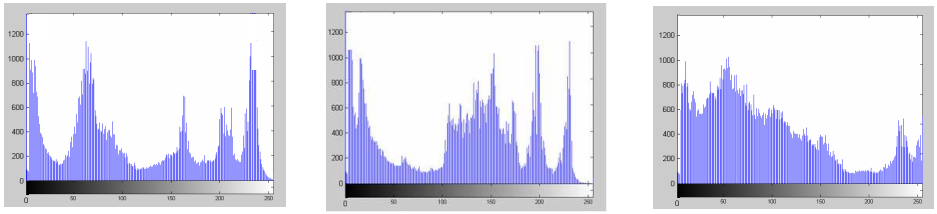
6	162	2	1	279	439	507	352	360	804	758	567
472	465	78	992	500	327	830	580	948	101	535	173
756	991	805	522	928	10	147	416	461	606	769	803
827	692	341	982	129	12	244	944	197	477	831	139
684	658	255	593	356	49	36	172	504	108	272	375
248	575	302	299	81	358	496	773	607	476	852	562
382	726	134	846	667	641	98	73	735	230	548	540
902	484	64	727	643	542	640	380	556	632	723	
835	436	748	109	721	654	619	212	277	894	211	475
705	335	257	844	856	781	938	462	961	11	623	62
651	254	74	4	746	265	525	130	943	185	52	473
457	160	251	860	59	939	447	121	946	511	757	934
258	136	974	350	35	50	483	491	965	32	941	453
339	344	514	929	137	470	27	502	724	131	184	512
362	677	213	871	802	981	39	5	570	345	848	76
328	884	412	238	68	391	376	488	325	292	882	71
142	970	810	196	220	796	730	132	296	622	550	490
590	574	3	560	87	826	158	8	518	817	973	635

819	409	689	498	206	353	526	890	41	617	440	647
396	915	935	797	96	242	975	297	285	608	177	264
782	563	99	203	659	7	544	104	761	629	719	271
333	329	877	581	332	838	778	611	704	762	674	156
972	899	610	930	366	785	927	976	956	31	263	117
312	315	664	775	752	337	413	996	878	853	963	46
637	866	111	538	310	105	30	418	445	179	814	933
151	867	597	274	86	467	609	984	273	152	834	43
408	303	594	44	505	696	419	516	631	732	794	712
284	588	682	435	874	521	858	589	969	906	33	138
398	722	698	529	372	524	93	378	122	486	554	494
448	655	103	833	547	808	9	148	924	855	728	713
386	691	751	226	468	239	311	688	482	628	28	737
753	566	373	602	543	168	118	318	326	813	14	307
309	275	485	818	942	870	653	481	868	950	993	861
690	499	766	648	443	349	679	280	493	673	824	56
424	199	321	889	621	962	166	218	977	714	487	444
336	582	281	402	624	553	387	747	441	390	546	261
743	306	492	603	58	616	267	235	89	528	703	630
811	364	497	381	715	800	250	665	893	94	379	361
913	627	532	262	530	585	980	295	388	21	697	241
903	828	971	851	995	686	672	469	888	615	519	905
595	908	501	832	169	433	897	718	63	508	228	80
816	385	922	115	862	91	423	428	95	559	427	873
786	779	676	909	15	604	979	710	354	680	370	578
601	739	150	245	157	907	342	128	953	417	454	404
881	223	240	82	17	270	646	792	125	153	940	896
266	694	571	552	135	127	898	986	901	669	772	229
842	837	916	346	371	523	47	661	464	771	783	839
790	18	308	460	577	702	154	289	770	990	57	463
114	69	24	642	400	120	671	232	650	97	48	249
774	895	605	178	919	314	863	442	926	53	558	634
921	989	458	29	587	568	741	406	434	145	657	561
791	920	369	340	246	645	821	749	429	614	403	539
330	509	144	945	225	918	541	394	112	537	706	850
405	60	879	900	845	222	374	789	474	625	549	425
209	584	276	317	421	652	426	67	843	599	198	355
947	338	557	305	37	431	319	678	290	876	155	638
596	806	451	331	124	744	809	54	411	51	182	883
912	825	170	807	16	983	857	904	666	836	955	660
988	61	175	420	159	395	742	951	515	133	26	869
815	268	410	205	219	176	191	446	729	347	252	864
534	958	849	207	466	234	65	531	90	269	75	478
384	221	181	795	183	551	22	968	459	911	479	399
414	286	734	670	192	685	925	959	545	777	300	287
174	415	357	106	88	840	960	987	42	859	636	780
471	745	294	701	592	164	600	750	733	377	764	320
949	208	422	716	681	291	140	301	449	711	83	393
165	668	343	952	84	917	700	892	126	113	639	195
954	759	966	573	85	644	102	231	224	92	885	914
180	143	613	40	141	994	887	348	875	200	20	367
188	316	456	110	720	489	503	569	450	260	626	822
119	38	383	793	767	708	186	171	167	579	736	812
572	72	784	163	77	365	555	392	243	765	452	146
283	187	322	282	107	760	683	886	407	693	662	216
841	23	707	363	564	210	359	55	999	964	401	738
217	324	847	506	598	116	201	13	586	253	576	695
823	798	865	591	699	323	256	190	389	872	788	620
288	829	533	763	247	717	313	298	19	985	854	149
278	520	931	998	455	100	687	820	34	351	768	193
923	967	740	755	480	432	25	227	725	161	799	932
189	675	937	663	204	66	957	237	565	334	891	583
618	787	233	304	259	397	997	536	656	612	437	527
495	202	368	936	510	438	633	649	801	754	910	776
880	123	194	214	236	513	978	293	215	430	517	709
731	1165	1126	1054	1145	1227	1122	1168	1335	1190	1338	1157
1224	1003	1310	1038	45	1301	1153	1295	1316	1032	1318	1329
1167	1115	70	1004	1339	1340	1341	1081	1150	1294	1342	1343
1319	1086	1125	1018	1269	1118	1012	1214	1344	1129	79	1332
1061	1213	1111	1205	1305	1130	1252	1093	1315	1057	1037	1299

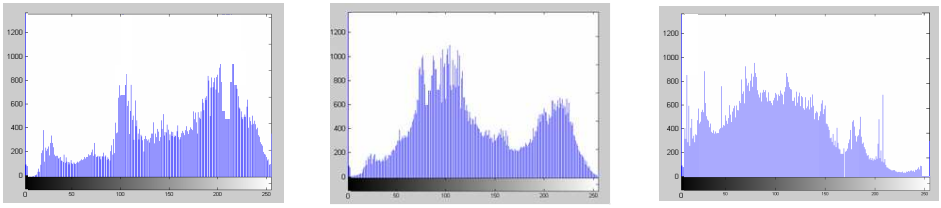
1083	1277	1062	1219	1345	1238	1346	1096	1221	1022	1103	1195
1257	1142	1050	1347	1274	1077	1088	1006	1058	1220	1046	1020
1171	1268	1116	1348	1278	1013	1188	1291	1237	1314	1296	1349
1350	1351	1109	1192	1352	1353	1189	1267	1229	1005	1155	1107
1243	1060	1008	1091	1128	1064	1023	1041	1293	1245	1248	1102
1113	1354	1263	1131	1355	1163	1356	1015	1039	1357	1247	1000
1053	1311	1358	1283	1049	1124	1065	1359	1360	1361	1281	1173
1254	1181	1114	1276	1191	1336	1104	1223	1184	1362	1079	1123
1226	1259	1137	1273	1112	1070	1363	1287	1035	1197	1364	1365
1140	1366	1367	1368	1127	1288	1369	1199	1244	1010	1133	1105
1024	1370	1371	1048	1180	1312	1317	1045	1139	1330	1177	1019
1040	1029	1055	1230	1225	1285	1275	1334	1372	1085	1036	1076
1206	1258	1208	1333	1028	1002	1154	1320	1186	1166	1161	1030
1135	1261	1095	1101	1067	1280	1136	1194	1097	1302	1260	1099
1034	1373	1172	1063	1374	1132	1094	1080	1337	1234	1265	1175
1066	1169	1047	1087	1246	1121	1375	1193	1068	1292	1376	1377
1138	1378	1325	1075	1326	1232	1203	1249	1092	1073	1379	1297
1152	1001	1071	1306	1026	1256	1242	1304	1380	1148	1156	1321
1201	1303	1021	1239	1210	1222	1381	1290	1251	1204	1025	1264
1382	1383	1084	1187	1228	1271	1146	1185	1078	1324	1384	1385
1215	1216	1176	1170	1100	1098	1212	1016	1031	1218	1209	1159
1196	1217	1178	1233	1241	1106	1162	1266	1141	1143	1090	1240
1250	1298	1328	1108	1307	1386	1262	1089	1387	1074	1027	1313
1388	1389	1056	1179	1009	1151	1119	1044	1255	1331	1134	1286
1272	1117	1033	1147	1231	1052	1149	1207	1309	1200	1164	1282
1390	1007	1160	1198	1082	1072	1308	1284	1253	1174	1069	1391
1202	1392	1182	1017	1270	1235	1236	1042	1043	1211	1322	1051
1183	1011	1120	1323	1158	1289	1279	1327	1110	1393	1014	1300
1059	1144										



(a) ***(b)*** ***(c)***



(d)



(e)

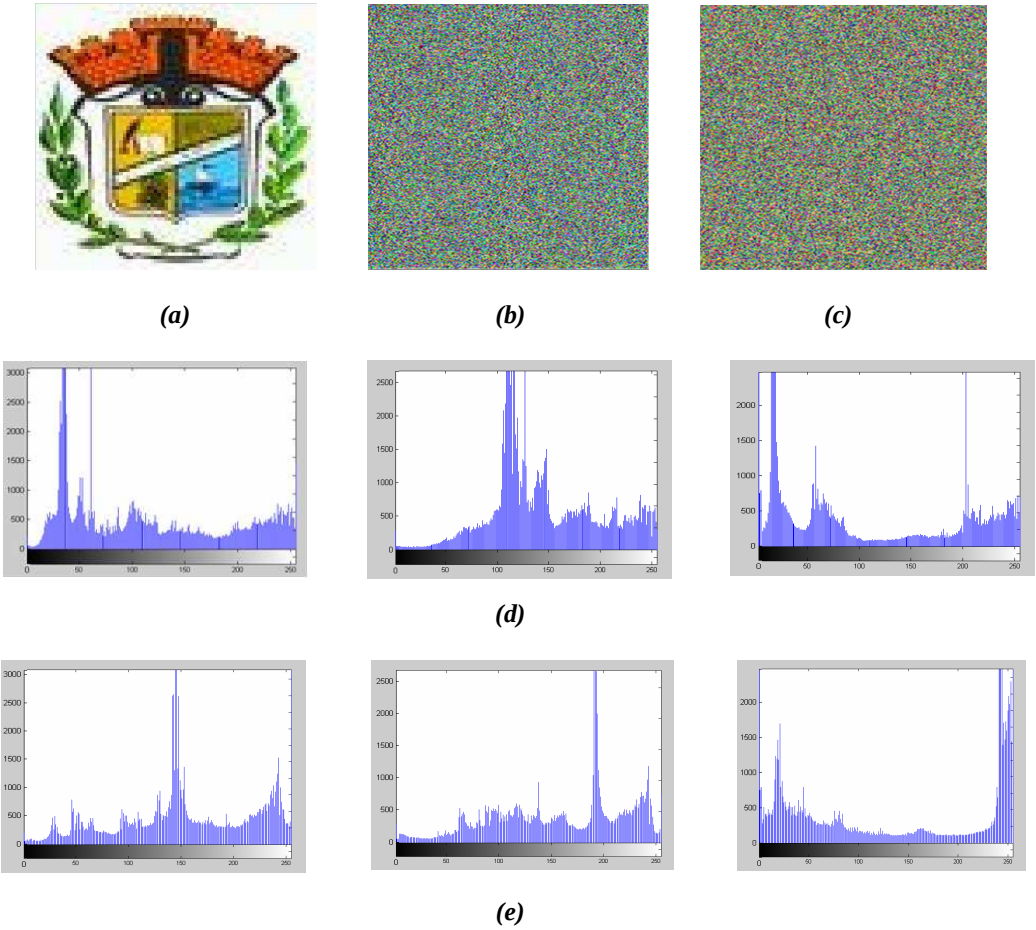
Figure IV.6. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

5	623	186	3	733	275	321	55	398	496	437	199
363	266	63	526	685	316	438	200	324	525	595	106
194	114	109	386	198	126	674	374	751	755	176	273
541	467	369	279	516	661	197	609	447	4	86	679
716	196	117	422	736	430	129	474	261	762	527	630
759	737	615	362	694	367	687	583	581	646	19	676
234	645	203	574	284	580	332	389	517	613	743	637
732	272	209	431	453	408	495	557	651	394	30	242
337	735	551	566	473	477	391	104	50	79	401	11
535	514	500	269	352	143	372	216	664	267	341	499
184	713	543	553	458	300	22	51	718	325	712	68
532	704	512	501	1	7	547	690	763	256	522	536
130	381	215	511	224	371	647	94	485	70	78	494
761	220	436	622	21	577	149	725	582	443	322	334
191	576	424	697	513	10	296	96	550	345	492	88
154	585	506	239	549	456	519	590	702	399	426	505
212	339	120	727	631	692	166	159	201	67	684	742
102	440	537	225	29	655	709	562	567	308	171	644
539	452	264	502	592	416	510	274	111	753	621	561
235	208	226	45	468	370	175	471	410	747	349	42
304	387	677	230	439	157	534	121	170	657	8	552
617	213	691	455	150	43	223	678	484	95	301	36
442	503	598	579	449	610	731	156	635	563	90	249
103	247	432	524	206	633	6	533	584	160	564	721
754	750	91	538	162	722	508	285	250	353	254	744
726	717	597	283	185	559	625	393	570	669	69	504
174	421	457	329	355	87	268	616	136	195	639	336
100	720	668	518	54	116	214	605	319	303	190	58
23	405	342	548	428	376	231	470	373	34	320	670
49	475	132	181	588	271	418	314	167	703	243	404
364	123	419	297	227	48	291	624	461	81	528	738
192	105	766	619	62	25	187	330	115	636	441	450
152	545	760	498	365	221	71	383	153	530	134	660
662	189	757	396	348	368	602	596	238	714	205	671
749	614	435	493	317	395	672	307	182	481	17	47
415	338	27	57	682	640	648	765	60	219	356	689
155	173	673	629	683	529	600	658	448	32	292	384
125	172	327	681	113	740	351	603	24	312	675	148
40	643	15	309	402	294	93	217	232	122	434	135
326	183	281	315	593	480	118	53	288	358	89	142
128	39	699	131	241	41	521	66	343	282	741	734
218	277	420	202	540	13	642	486	589	204	723	478
730	715	340	710	764	80	653	472	211	445	483	411
459	464	433	18	392	138	618	406	377	571	193	591
360	366	158	680	252	469	382	145	52	35	462	72
407	278	133	73	101	412	26	165	745	344	575	12
599	400	739	695	748	310	568	139	293	140	608	403
375	708	587	768	298	666	76	554	251	606	414	572
245	97	222	74	270	507	388	44	601	696	546	594
229	107	706	168	311	350	487	112	667	255	33	444
361	177	454	169	164	479	509	82	686	463	515	228
663	244	357	649	711	259	628	265	397	707	728	423
665	2	632	626	460	654	210	290	124	767	163	333
280	429	295	489	451	698	565	127	37	641	491	688
161	531	302	719	179	75	729	724	604	482	693	347
240	413	61	490	746	378	586	258	84	627	427	248
263	335	16	542	523	233	556	31	144	257	38	64
331	92	544	701	650	466	56	573	188	659	306	578
253	287	756	236	425	299	119	611	65	9	555	323
141	417	110	318	359	20	146	77	656	758	652	207
289	151	237	98	385	620	700	85	137	634	147	260
354	180	276	560	59	380	465	313	569	346	409	638
558	108	328	612	476	488	46	446	262	286	246	178
607	83	705	752	390	497	99	305	28	379	14	520

Clé de chiffrement de la deuxième version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

81	506	289	165	107	184	553	751	570	17	729	724
457	204	65	760	510	256	116	708	314	296	315	381
538	6	734	302	480	33	139	345	54	144	113	57
393	424	399	163	470	71	599	733	51	653	151	30
175	743	321	556	187	232	112	323	363	707	588	183
235	260	128	37	477	453	462	267	88	304	351	56
308	359	762	464	121	374	294	382	715	392	478	609
647	205	431	253	89	594	666	262	622	592	224	322
311	164	277	29	341	291	154	646	699	522	495	74
625	47	320	395	644	77	43	100	494	723	402	706
702	105	152	244	283	737	404	340	750	629	396	764
125	412	49	387	22	364	484	643	337	242	131	389
342	690	120	572	567	435	745	176	466	13	209	403
517	508	279	265	4	252	117	132	475	587	391	310
635	331	492	502	360	372	380	456	710	104	96	669
565	416	257	531	3	326	612	171	568	1	563	444
434	507	41	259	505	577	84	192	147	447	597	747
347	711	335	744	383	421	540	367	394	660	143	281
657	178	339	418	562	436	301	766	27	94	516	66
703	161	91	313	58	63	136	610	203	45	162	292
195	483	317	705	228	62	196	476	18	539	64	368
141	451	503	39	169	307	229	297	672	350	670	626
596	309	9	19	740	353	519	160	673	134	614	400
541	75	649	605	748	223	82	137	388	525	667	655
535	640	90	585	659	448	243	295	170	48	299	237
754	533	726	452	548	528	231	230	590	700	24	586
182	678	732	101	573	290	156	272	207	631	607	701
177	445	155	365	545	73	38	583	284	80	560	138
591	469	559	97	423	398	119	208	603	527	512	514
767	78	12	173	490	95	324	126	619	349	69	488
665	662	460	693	459	180	420	604	509	720	598	595
529	174	5	50	87	271	683	561	696	422	188	59
245	620	439	482	240	580	361	385	685	593	755	142
358	274	415	497	571	698	677	370	536	679	31	663
641	219	468	736	375	222	189	757	432	158	216	639
471	413	633	455	23	85	642	407	168	651	675	238
206	450	123	146	465	518	276	739	546	319	199	145
611	106	689	636	688	714	461	768	348	632	214	133
442	333	46	129	118	344	654	10	68	722	627	379
581	575	758	730	390	661	411	682	172	249	334	14
652	695	11	336	576	487	278	725	268	384	61	630
191	520	211	521	239	589	756	26	286	409	417	127
227	501	55	298	426	713	735	35	148	159	437	401
186	330	99	638	618	550	410	226	15	44	40	430
742	103	140	458	111	369	449	373	537	202	615	83
270	674	472	600	564	608	397	645	215	234	721	130
261	153	269	515	579	316	405	741	621	727	628	656
254	354	352	122	72	763	76	287	34	752	181	446
544	21	236	566	441	524	617	300	513	79	248	467
115	731	474	408	704	491	305	547	557	377	20	427
719	697	102	357	52	346	526	709	582	185	718	149
542	481	7	716	318	761	606	312	288	201	429	109
549	493	338	285	378	454	86	438	613	419	601	329
303	293	712	110	728	247	433	157	443	282	212	218
197	634	694	746	489	135	558	623	717	680	246	648
264	664	366	574	676	124	362	36	53	8	213	200
530	668	114	555	543	485	650	498	233	42	150	325
166	98	511	343	749	738	414	552	753	251	250	255
356	280	386	266	551	28	273	328	93	499	275	92
486	108	198	534	32	532	194	70	569	217	691	658
406	578	16	263	327	2	221	496	765	463	332	681
616	500	25	428	554	479	306	67	637	584	371	190
692	473	258	60	759	220	425	440	671	523	167	225
210	355	684	602	179	687	376	193	686	624	504	241



(a) ***(b)*** ***(c)***

(d)

(e)

Figure IV.7. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée*
(d) *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

92	39	9	38	13	16	45	76	98	47	24	8
83	40	3	78	80	90	61	26	68	89	53	52
19	31	50	18	12	14	54	64	2	46	58	65
35	71	5	28	74	36	44	73	79	99	96	62
63	94	29	15	37	11	75	10	30	43	7	51
91	55	33	66	57	23	81	41	70	56	77	88
4	95	25	1	69	86	22	82	97	67	42	59
20	87	32	49	72	48	27	425	652	573	441	543
468	129	748	420	738	513	353	453	569	700	185	525
707	143	512	264	237	198	755	658	487	221	741	557
484	647	227	257	708	162	179	579	197	724	60	326
629	757	548	220	413	115	266	258	120	443	218	639
253	669	673	260	712	202	345	430	225	561	284	449
481	109	294	661	248	552	710	354	433	644	372	630
161	684	648	21	476	432	110	489	331	503	403	111
507	363	637	680	17	734	556	588	438	692	666	367
753	723	613	287	34	604	505	168	273	267	421	564
239	542	742	382	642	471	508	278	614	333	359	234
136	764	285	765	545	672	259	114	214	572	170	520
147	458	502	352	244	659	423	688	537	175	386	706
609	678	328	149	272	462	393	567	379	6	144	330
523	145	562	417	119	761	767	226	478	355	726	518
737	722	583	127	440	238	397	429	339	247	173	450
296	635	254	378	341	213	488	134	409	292	731	427

399	494	193	280	314	93	704	123	469	148	634	657
303	344	501	705	404	733	171	627	662	551	311	470
743	510	401	157	554	346	491	340	698	256	240	347
125	611	727	589	358	568	374	325	631	307	361	745
308	756	746	164	369	131	306	448	640	245	565	118
586	668	167	580	313	553	758	189	84	124	350	262
645	703	676	574	540	255	348	575	515	286	128	664
398	366	531	113	217	456	300	681	224	223	593	699
691	422	585	506	402	434	641	270	528	446	426	696
301	656	690	302	529	151	682	204	735	571	416	663
165	108	514	435	444	689	190	290	158	602	140	335
299	582	174	677	516	616	203	736	104	454	536	625
279	577	396	495	312	750	310	349	559	205	315	141
728	269	192	250	208	695	718	709	740	618	687	587
550	283	322	142	209	282	199	187	276	336	653	229
321	176	730	626	714	342	327	219	599	139	608	112
752	566	603	581	390	210	466	373	650	576	152	447
233	163	384	196	242	362	768	180	188	570	461	601
126	216	395	365	497	309	751	201	230	739	595	177
178	249	133	605	766	116	207	394	667	318	281	194
530	600	117	376	235	636	623	646	85	732	598	555
437	241	693	334	222	100	293	612	406	184	492	563
560	547	534	679	407	215	535	351	674	332	475	670
271	212	275	320	558	749	721	418	181	186	633	371
701	591	385	517	533	467	638	686	490	590	364	375
154	606	683	388	762	759	200	474	729	649	405	360
500	166	655	389	343	538	628	493	524	596	483	594
527	172	243	549	754	304	473	415	477	103	597	615
464	295	725	368	412	195	277	231	713	632	297	480
211	442	459	620	532	660	651	319	410	544	485	419
445	744	101	252	191	121	298	486	265	439	182	463
150	452	392	457	400	146	106	747	526	357	370	665
317	482	763	377	720	451	578	621	455	356	324	521
130	288	228	156	323	122	504	643	246	522	496	619
436	697	694	675	431	316	584	232	607	107	381	671
719	261	428	411	408	498	105	159	717	610	539	715
153	268	546	617	387	102	160	414	685	716	465	460
338	511	391	622	289	291	479	138	702	305	711	424
654	155	499	169	329	137	183	251	383	132	592	472
206	263	541	135	760	509	274	624	337	380	519	236

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

528	192	38	202	98	315	156	142	97	339	304	457
569	700	485	538	200	292	180	544	443	684	130	287
352	689	571	468	536	101	169	244	639	353	215	24
735	501	545	266	628	575	359	745	653	1	694	535
411	37	9	342	491	497	146	278	620	88	533	80
151	380	298	137	521	508	409	61	123	191	728	662
558	115	398	514	454	50	323	541	429	410	686	488
228	232	247	559	378	234	181	261	233	345	243	643
758	387	384	33	8	761	412	346	371	695	750	40
308	477	128	179	487	493	294	604	11	504	129	710
68	39	196	739	725	753	43	63	187	73	26	749
401	578	162	176	404	395	699	302	140	270	157	272
563	341	249	81	381	92	652	58	673	289	67	712
547	768	94	322	172	218	241	69	721	438	618	645
242	723	611	556	635	688	297	193	641	474	553	62
654	25	502	310	301	148	269	428	377	70	106	592
672	617	258	439	708	658	34	737	274	407	113	112
279	12	209	119	326	531	207	677	584	42	282	330
199	237	268	648	251	731	447	338	744	685	126	557
216	610	621	483	328	467	205	674	184	259	44	210
649	532	75	600	114	625	473	290	334	83	596	752
256	434	470	82	650	425	141	482	763	534	116	201
354	385	96	549	601	726	576	717	283	173	730	212
707	525	522	373	452	281	348	117	227	318	589	565
208	663	214	505	711	609	276	93	526	31	767	23
636	687	667	372	570	303	581	369	252	510	343	546
540	219	177	418	182	722	760	495	171	656	255	78

329	691	608	293	580	690	229	27	262	513	554	223
296	22	152	432	57	333	159	211	347	516	665	675
727	676	124	450	713	299	564	86	679	4	757	226
311	335	393	402	161	217	702	102	741	54	260	397
637	213	118	548	230	32	364	622	382	111	254	379
568	107	174	165	366	195	417	153	759	512	697	764
367	696	742	76	597	394	435	158	480	361	103	149
120	448	706	755	716	178	166	520	602	537	422	740
682	577	327	357	475	168	560	585	598	436	35	413
110	189	414	250	127	552	358	332	403	190	668	41
587	424	122	588	567	391	612	17	376	416	631	291
355	175	307	222	456	154	340	66	427	586	331	238
542	84	370	484	295	566	183	471	437	236	529	583
593	52	459	319	614	271	449	613	131	698	55	6
313	19	280	60	550	325	349	59	46	714	396	671
453	45	605	245	337	105	85	350	469	167	664	400
669	351	623	660	197	720	591	594	224	16	246	51
421	562	692	2	724	478	388	442	666	53	3	320
419	659	132	91	383	423	235	121	71	462	461	481
543	20	147	160	515	415	316	87	527	203	300	754
286	138	681	288	15	284	579	511	143	704	524	65
509	632	198	574	64	163	29	455	231	139	624	733
638	539	765	275	309	680	472	263	108	321	463	607
640	693	523	135	630	619	701	389	683	627	732	170
709	99	603	420	426	312	441	374	440	392	747	77
719	136	661	317	756	368	406	48	476	489	305	748
109	715	465	615	285	496	375	155	21	446	519	408
561	18	498	220	734	336	5	530	188	762	492	399
464	13	518	555	499	766	433	431	204	458	273	751
248	74	89	479	186	100	705	506	551	500	405	572
28	267	14	743	444	164	150	240	430	362	655	257
616	265	657	185	494	703	644	642	56	134	646	390
445	104	678	647	729	718	363	466	306	599	629	517
360	10	507	746	225	736	595	47	253	651	194	95
582	633	490	486	451	344	606	738	7	634	277	386
503	144	324	30	670	36	365	125	145	356	79	264
314	590	90	626	460	573	206	133	49	239	72	221

Le tableau suivant résume les résultats relatifs au temps de calcul (temps de chiffrement), la taille de clé, la quantité de phéromone et l'efficacité dans le cas de chiffrement des différentes données.

			Taille donnée (éléments)	**Taille clé (bits)**	**Efficacité**	**Temps calcul (s)**	**Phéromone**
Données Texte	Texte1	Version-Chiff1	1084	15323	11784.0	5.63	9,06
		Version-Chiff2			12030.0	6.02	9,13
	Texte2	Version-Chiff1	1151	15323	14498.0	6.7	8,67
		Version-Chiff2			14712.0	6.58	8,65
Données Images	Lena	Version-Chiff1	131 X 131	7680	135362.0	2.16	2.75
		Version-Chiff2			130074.0	2.31	1.17
	Logo	Version-Chiff1	420 X 395	7680	244926.0	2.87	23.11
		Version-Chiff2			226188.0	2.84	21.81

Tableau IV.2. *Résultats obtenus par AntCrypt.*

IV.4. Discussion et évaluation des résultats

La première chose à remarquer, d'après les résultats résumés à travers le tableau IV.2, c'est que le temps de chiffrement des données texte est beaucoup plus grand que celui des données images malgré que la taille des données images est généralement plus grande en comparaison à celle des données texte. Cela revient au fait que la taille du vecteur codant les données texte (1393 éléments) est beaucoup plus grande que celle du vecteur codant les données images (768 éléments), ce qui nécessitera beaucoup plus de temps de calcul lors de l'application du processus de résolution à base de fourmis. Toutefois, le temps de chiffrement est raisonnable pour les deux types de données texte ou images.

Nous remarquons, aussi, que les temps de calcul des différentes versions d'une même donnée sont proches, car ils sont indépendants de la taille de la donnée à chiffrer (une différence apparaît quand la dissemblance entre les tailles de deux données de même type sera importante et qui représente un temps de lecture ou de codage). En effet, la taille du vecteur codant toutes les données de même type et soumises au chiffrement est la même (1393 éléments pour le cas de texte / 768 éléments pour le cas d'images), donc sa manipulation sur l'ensemble des générations prend le même temps. La toute petite différence correspond au temps de codage des données avant le lancement du processus de chiffrement. Toutefois, la légère différence entre le temps de chiffrement des deux versions d'une même donnée revient au caractère aléatoire exploité dans certaines phases de l'algorithme.

Pour le cas de chiffrement des deux données images Lena et Logo, nous constatons que le temps de chiffrement de l'image Logo est plus grand que celui de l'image Lena. La différence est due au fait que le temps de chiffrement est la somme des temps de lecture puis codage de l'image qui varie suivant la taille de l'image, plus le temps de manipulation du vecteur codant les solutions qui n'a aucune relation avec la taille de l'image. Donc, la différence en temps de chiffrement des deux images est justifiée.

La valeur de la phéromone dépend de la valeur d'efficacité maximale et de l'efficacité de la solution calculée. Il est bien clair que pour de grandes valeurs d'efficacité, une amélioration de la quantité de phéromone sera marquée. Prenons l'exemple du premier message où pour la première version chiffrée nous avons obtenu 2166 / 9,06 pour efficacité / phéromone. Cependant, pour la deuxième version chiffrée, nous avons eu 2184 / 9,13. Donc, l'amélioration de l'efficacité est certainement accompagnée d'une amélioration de la quantité de phéromone. Il sera utile de rappeler, ici, que l'évaporation est la cause de diminution de la quantité de phéromone.

En mesurant le taux de différence entre les deux versions chiffrées de l'image Lena et l'image originale (l'image Lena), le tableau suivant récapitule les résultats obtenus en termes de NPCR, MAE et MSE. Il est clair que les deux versions chiffrées sont presque totalement différentes de l'image originale. Autrement dit, les pixels formant les versions chiffrées sont presque tous changés, soit de positions, soit de nombres d'occurrences et de positions ; ce qui donnera lieu à un très bon niveau de confusion.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.9207	1.08	0.73
Lena-version chiffrée 2	0.9082	1.03	0.69

TableauIV.3. *Niveaux de confusion de AntCrypt.*

Maintenant comparant AntCrypt avec le plus rapide de nos trois algorithmes développés et présentés dans le précédent chapitre, bien sûre sur le plan de temps de calcul. Le tableau suivant rappelle les temps de convergence des deux algorithmes OEEA et AntCrypt. Ce dernier a montré un temps de calcul meilleur que celui de OEEA que ça soit pour chiffrer des données textes ou images malgré qu'aucune explication exacte ne peut être donnée du fait que les deux algorithmes utilisent un même mode de chiffrement (chiffrement à base d'occurrences), donc un même codage qui est exploité par les composants de la métaheuristique impliquée (opérateurs génétiques et mécanismes de la sélection naturelle pour OEEA et fourmis et mécanismes de manipulation de phéromone pour AntCrypt). Et vu que le temps de convergence de notre algorithme de chiffrement basé sur les mécanismes évolutionnaires (OEEA) est légèrement plus grand que celui basé sur les fourmis (AntCrypt), donc possible que les premiers mécanismes consomment plus de temps de calcul que les deuxièmes, mais aucune preuve n'est connue !

	Temps calcul (s)	
	Texte 1	Lena
AntCrypt	5.825	2.235
OEEA	6.54	3.37

Tableau IV.4. *Temps de calcul de AntCrypt en comparaison avec le temps de calcul de OEEA.*

Remarque :

Dans le tableau IV.4, les temps de calcul de l'algorithme AntCrypt pour les deux cas de données (Texte 1 et Lena), représentent la moyenne des temps de calcul des deux versions chiffrées (Voir tableau IV.2).

De même et vu que la méthode de chiffrement présentée dans ce chapitre (Cryptage à base de fourmis) et celle présentée dans le précédent chapitre (Cryptage évolutionnaire) appartiennent à une même catégorie mère qui est celle de métaheuristiques, donc toute les deux vont bénéficier de quelques caractéristiques cryptanalytiques en commun. Il s'agit de l'aspect non déterministe démontré à travers les résultats présentés ci-dessus où deux versions chiffrées différemment correspondent toutes les deux à une même donnée originale (voir figures IV.4, IV.5, IV.6 et IV.7), et d'une robustesse contre les types d'attaques les plus avancées (attaque exhaustive, attaque statistique, attaque fréquentielle et attaque différentielle) comme c'était démontré dans la section III.5 du chapitre précédent.

IV.5. Conclusion

ACO est une méthode stochastique qui requiert de plus en plus l'attention de la communauté scientifique. Cette métaheuristique a prouvé sa performance pour plusieurs problèmes d'optimisation combinatoire. En effet, l'utilisation des traces de phéromone permet d'exploiter l'expérience de recherche acquise par les fourmis et ainsi renforcer l'apprentissage pour la construction de solutions dans les itérations futures de l'algorithme. En même temps, l'information heuristique peut guider les fourmis vers les zones prometteuses de l'espace de recherche.

Ainsi et dans ce chapitre, nous avons présenté, interprété et discuté les résultats expérimentaux obtenus par l'application de notre algorithme de chiffrement proposé *AntCrypt* sur des données modèles texte et images. Nous sommes arrivés à fixer nos paramètres par expériences, en choisissant 40 générations et 12 fourmis en essayant de réaliser un compromis entre l'efficacité et le temps de chiffrement.

L'algorithme, ainsi développé et qui a été nommé *AntCrypt*, a été comparé avec nos algorithmes développés à base d'algorithmes évolutionnaires où il a montré de bonnes caractéristiques telles qu'un bon niveau confusionnel, un aspect non déterministe, une taille sécurisée de clés, lui offrant une bonne résistibilité contre les attaques les plus avancées. En plus le temps de convergence de cet algorithme est meilleur que celui du plus rapide de nos trois algorithmes proposés dans le précédent chapitre.

CHAPITRE V

CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

V.1. Introduction

Nous nous sommes intéressés, dans les deux chapitres précédents, à la résolution du problème de cryptage de données texte et images par application de métaheuristiques à populations. Les approches que nous avons proposé sont caractérisées par leur qualité de confusion, leur temps d'exécution réduits pour certains d'eux (OEEA et AntCrypt) et leur robustesse face aux attaques avancées.

Dans ce chapitre, nous nous intéressons à l'exploitation d'une métaheuristique appartenant à une autre catégorie, les métaheuristiques à trajectoire. Il s'agit de la recherche tabou. Cette dernière s'appuie sur une recherche locale combinée à un mécanisme de prévention des cycles, grâce à un système de mémoire des mouvements précédemment appliqués ou des configurations visitées (la liste tabou).

De manière générale, une recherche locale démarre d'une solution initiale possible et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant. Un voisinage d'une certaine solution correspond à des éléments adjacents à cette solution dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration dans la solution courante ne puisse être faite.

Nous présentons, après la description de l'approche proposée, les résultats numériques obtenus afin de les comparer aux résultats obtenus par nos autres méthodes proposées (cryptage évolutionnaire et cryptage par colonies de fourmis) et certaines autres méthodes de la littérature.

V.2. Motivation

La recherche Tabou est une métaheuristique basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

L'originalité de la méthode de recherche tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objective f et l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum

quelconque par acceptation de la dégradation de la fonction objective, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste d'une certaine longueur, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

V.3. Algorithme proposé

À partir d'une solution initiale, le principe général de la recherche tabou est le suivant (voir Algorithme V.1). À chaque itération de la recherche, une partie ou l'ensemble des voisins de la solution est exploré, et un des mouvements parmi ceux de coût minimal est sélectionné. Ce mouvement est appliqué quel que soit son coût, à la condition de ne pas créer un cycle dans le processus de recherche locale à court terme (en menant à une configuration dont les caractéristiques sont stockées dans la liste tabou). Une exception est faite (critère d'aspiration) lorsque le mouvement permet d'atteindre une solution de meilleure qualité que la meilleure solution enregistrée à ce stade. La liste tabou est mise à jour à chaque itération de la recherche en fonction des mouvements choisis. Ce mécanisme permet de sortir des minima locaux en acceptant des mouvements détériorants et en empêchant parallèlement le retour immédiat à la solution de minimum local qui vient d'être quittée.

Algorithme V.1 *Schéma général d'un algorithme tabou*

```
Engendrer une configuration initiale s
s* ← s
T ← ∅ liste tabou
tant que condition d'arrêt non satisfaite faire
    m ← meilleur mouvement (i.e. minimisant f) parmi ceux non tabou
    ou ceux vérifiant un critère d'aspiration
    Modifier s en effectuant le mouvement m
    Mettre T à jour
    si f(s) < f(s*) alors
        s* ← s
    fin
fin
retourner s*
```

La succession des étapes suivantes montre plus de détails du schéma du mécanisme utilisé pour la construction d'une solution au problème étudié :

1) Détermination d'une solution initiale $S_{(0)}$ codant l'image originale.
2) Création puis la mise à jour d'une liste tabou qui mémorisera les déplacements effectués.
3) Choix aléatoire d'un nombre d'emplacements ou d'éléments de la solution courante ($i^{ème}$ solution) à déplacer.
4) Génération aléatoire d'un ensemble de voisins pour chacun des éléments choisis dans l'étape précédente.
5) Choix du meilleur voisin de l'ensemble généré.
6) Calcul tabou de la solution suivante $S_{(i+1)}$.
7) Création et la mise à jour d'une table de hachage qui comportera les dernières meilleures solutions obtenues.
8) Évaluation de la solution calculée $S_{(i+1)}$ obtenue à chaque itération.
9) Validation de la solution calculée $S_{(i+1)}$.
10) Vérification du critère d'arrêt de l'algorithme.

En ce qui suit, nous décrivons les principales étapes de l'algorithme de chiffrement tabou proposé et que nous l'avons appelé *TabuCrypt*.

V.3.1. Création de la solution initiale

La solution initiale exploitée par *TabuCrypt* s'obtient en codant les données originales suivant le même mode de codage adopté par *OEEA* ou *AntCrypt*. C'est d'ailleurs le même codage utilisé pour modéliser toutes les autres solutions.

V.3.2. Choix des éléments à déplacer

Les solutions sont représentées soit, par un vecteur englobant 1393 éléments pour le cas de manipulation de données texte soit, par un autre regroupant 768 éléments pour le cas de manipulation de données images. *TabuCrypt* cherche à réarranger aléatoirement le vecteur correspondant à la solution initiale codant la donnée originale en permutant les éléments entre eux. Toutefois, la taille du vecteur est assez grande (1393 éléments ou même 768 éléments), c'est pourquoi la manipulation un par un de ses éléments sur l'ensemble des itérations de l'algorithme nécessitera un grand temps de calcul. Ainsi, sauf un sous ensemble du vecteur globale sera manipuler pour déplacement. Cela permet, d'un coté, d'optimiser le temps de calcul et, d'un autre coté, de converger petit à petit vers la solution finale en évitant, ainsi, le problème de convergence prématurée.

V.3.3. Génération de voisinage

Le voisinage est le responsable sur le fait de pousser l'algorithme à l'amélioration de la qualité des résultats. Dans notre cas, il est généré de la manière suivante :

Pour chacun des éléments du sous-ensemble construit dans l'étape précédente, nous générons aléatoirement un ensemble de voisins candidats au déplacement avec l'élément en question. Lors de l'élection d'un voisin, les conditions suivantes doivent être vérifiées :

- ✓ Un voisin ne doit pas être élu plus qu'une fois, c'est-à-dire, tous les voisins de l'ensemble de voisins sont différents les uns des autres.
- ✓ Les voisins élus n'appartiennent pas dans la liste tabou.

Une fois l'ensemble des voisins construit, nous choisissons celui qui diffère le plus de l'élément à déplacer. Formellement, soit E_V l'ensemble de n voisins V, et E_i l'élément candidat au déplacement. La sélection du voisin v avec lequel l'élément E_i sera permuté se fait comme suit :

$$v = V_j \,/\, j = \overline{1,n} : Max\left(\left|E_i - V_j\right|\right) \qquad \text{(V.1)}$$

V.3.4. Mise à jour de la liste tabou

Dans notre cas et à une itération donnée, la liste tabou regroupe les voisins qui ont participé à des déplacements au cours des itérations précédentes avec l'un des éléments du sous-ensemble construit dans l'étape 2 décrite dans la section V.3.2. Ainsi, la mise à jour de cette liste (liste tabou) se fait à chaque fois qu'un élément sera permuté avec un voisin choisi suivant l'étape 3 décrite dans la section V.3.3 en ajoutant ce dernier à la liste des éléments tabous. A la fin de chaque itération, le contenu de la liste tabou sera effacé.

La politique utilisée pour rendre tabous certains éléments évite la modification des nombres d'occurrences originaux, puisque le processus de calcul de la solution ne doit pas modifier les éléments du vecteur initial mais plutôt il cherche à changer leur dispersion.

V.3.5. Calcul de solutions

Au cours d'une itération donnée, le calcul de la solution proposée à cette itération se base sur la solution calculée à l'itération précédente. En effet, des éléments du vecteur présentant la solution à l'itération précédente échangeront leurs positions avec celles des voisins désignés pour chacun d'eux. Cette nouvelle dispersion des éléments forme le nouveau vecteur solution. Il est à noter que les éléments non élus pour un déplacement garderont leurs positions sans modification.

V.3.6. Mise à jour de la table de hachage et évaluation des solutions

La table de hachage sert à mémoriser les différents points visités de l'espace de recherche en gardant les différentes solutions calculées sur l'ensemble des itérations. Le but sera d'interdire la recherche à revenir vers des points déjà visités à travers les précédentes itérations. Ainsi, une solution générée à la fin d'une certaine itération sera comparée au contenu de la table de hachage avant qu'elle soit soumise à une validation finale en vu de l'accepter comme intéressante solution à partir de laquelle la recherche peut continuer sur l'ensemble des itérations restantes. Si cette solution représente un nouveau point de l'espace non encore visité, elle sera ajoutée à la table de hachage pour quelle fera l'objet des futurs tests. Cette politique sert à échapper aux minima locaux et à favoriser l'exploration de nouveaux points de l'espace de solutions possibles.

La validation définitive d'une solution S_i calculée à une itération i donnée, consiste tout d'abord, à l'évaluer. Dans notre cas, l'évaluation se fait suivant la fonction d'évaluation illustrée par la formule suivante, notant que S_0 soit la solution initiale (voir étape 1).

$$F(S_i) = \sum_{j=1}^{l} \left| S_{i_j} - S_{0_j} \right| \qquad \text{(V.2)}$$

Où l : représente le nombre de valeurs possibles que peut prendre un élément d'une donnée (égale à 1393 pour le cas de données texte et 768 pour le cas de données images).

Une fois l'évaluation faite, la qualité de la solution en question sera comparée à celle de la solution calculée à l'itération précédente. Si elle est meilleure que la précédente, la recherche à travers la prochaine itération va se continuer à partir du point représenté par cette solution, sinon elle sera rejetée et la recherche continuera à partir de la solution calculée à l'itération précédente.

Ces points peuvent être résumés comme suit : à chaque itération la solution calculée est ajoutée à la table de hachage puis elle sera examinée selon les trois critères suivants :

1) Si la solution S_i obtenue n'est pas une nouvelle solution alors elle sera rejetée et les étapes de 2 à 6 sont à refaire.

2) Si la solution S_i obtenue est une nouvelle solution mais plus mauvaise que celle de l'itération *i-1* (S_{i-1}), de même elle est rejetée, mais sauvegardée dans la table de hachage, et les étapes de 2 à 6 sont à refaire.

3) Si la solution S_i obtenue est une nouvelle solution mais, cette fois ci, est meilleure que celle de l'itération (i-1), alors elle sera retenue pour continuer la recherche à partir de la prochaine itération et, ainsi, elle servira à calculer la solution suivante S_{i+1}. En même temps, elle sera mémorisée dans la table de hachage.

Remarque :
Dans notre algorithme, on considère comme *mouvement global* le passage d'une solution courante S_i vers une solution suivante S_{i+1}, en respectant les conditions citées précédemment. Toutefois, les *mouvements élémentaires* sont les déplacements des éléments et leurs voisins vers leurs positions mutuelles. Ainsi, un mouvement global émerge comme résultat des mouvements élémentaires.

V.3.7. Critère d'arrêt

Pour toute recherche itérative un critère d'arrêt est obligatoire pour éviter le problème de boucles infinies. Ce dernier qui doit être fixé à l'avance permet de déterminer le nombre de fois que la tâche à exécuter sera répétée.

Pour notre algorithme, nous avons choisi de fixer, expérimentalement, le nombre d'itérations. Ce choix influe directement sur longueur du temps de calcul et de la qualité de la solution construite.

V.3.8. Mécanismes avancés en recherche tabou

V.3.8.1. Intensification / Exploitation

L'intensification consiste à approfondir la recherche dans certaines régions de l'espace, identifiées comme susceptibles de contenir un optimum global.
Pour notre cas, les mouvements globaux en plus des mouvements élémentaires serviront à mieux intensifier la recherche. En effet, les mouvements élémentaires traduisent les déplacements des éléments d'un vecteur solution vers les positions de leurs voisins et vise versa. Ces voisins sont les éléments optimaux (les meilleurs voisins) de l'ensemble des voisins candidats au déplacement (voir étape 3). Donc, nous cherchons à exploiter les qualités des éléments. De même, les mouvements globaux cherchent à exploiter les qualités des solutions (les vecteurs d'éléments) en n'acceptant de poursuivre la recherche qu'à partir des solutions améliorantes lors du passage d'une itération à l'itération qui suit.

V.3.8.2. Diversification / Exploration

La diversification vise à utiliser des mouvements encore jamais réalisés afin d'explorer de nouvelles régions de l'espace de recherche et des régions éloignées du voisinage actuel. Dans notre cas, cette caractéristique est accomplie par la sélection aléatoire d'un nouveau voisinage non tabou pour chaque élément.

V.3.8.3. Critère d'aspiration

Le critère d'aspiration autorise un mouvement tabou sous certaines conditions. Il consiste à tester si la solution produite de statut tabou présente un coût inférieur ou de qualité meilleure que ceux de la meilleure solution trouvée jusqu'à présent. Si c'est le cas, le statut tabou de la solution est levé.

Dans notre cas, la notion de tabou est exploitée lors de la construction d'une certaine solution en interdisant les mouvements élémentaires vers les éléments voisins avec lesquels les éléments du sous-ensemble construit comme l'indique l'étape 2 ont échangé de positions. Ainsi, le mécanisme d'aspiration n'a pas de place dans notre méthode du fait que toute libération d'un élément de la liste tabou entraîne une modification des éléments des vecteurs et, par conséquent la modification des caractéristiques servira à calculer ultérieurement la donnée déchiffrée, tandis que notre but est, plutôt, modifier la répartition des éléments des vecteurs.

V.3.9. Déchiffrement

L'opération inverse au chiffrement est le déchiffrement qui permet de regénérer la donnée originale précédemment chiffrée par notre algorithme de chiffrement tabou *TabuCrypt*. Pour ce faire, ce processus exploite une information secrète calculée lors du chiffrement à coté de l'image chiffrée. Il s'agit d'une clé de session secrète. Elle représente les permutations des positions des l éléments (1393 éléments ou 768 éléments) du vecteur codant une certaine donnée à travers les différentes itérations. Une fois elle parviendra sans modification au destinataire approprié, elle servira à calculer la donnée originale.

V.3.10. Réglage des paramètres et résultats

Le schéma général de l'algorithme finalement implémentant les étapes de l'approche de cryptage tabou proposé est le suivant :

Algorithme V.2 *TabuCrypt*

Étape 1 : Générer une solution initiale de gain total G_0.
Étape 2 : Initialiser
- le nombre d'itération iter (à 1),
- la table de Hachage H (ajouter la solution initiale à H),
- la liste tabou T (vide).

Répéter les étapes de 3 à 11 jusqu'à iter = iterMax.
Étape 3 : Initialiser le compteur d'éléments (composants d'une solution)
Répéter les étapes de 4 à 6 jusqu'à
Étape 4 : Sélectionner aléatoirement un ensemble de voisins des éléments non tabous, puis choisir le meilleur de l'ensemble.
Étape 5 : Déclarer comme tabou le voisin choisi (voisin ajouté à T).
Étape 6 : Déplacer l'élément courant et le voisin choisi vers leurs positions mutuelles.
Étape 7 : Evaluer la nouvelle solution calculée pour mesurer son gain G_{iter}.
Étape 8 : Si : la solution calculée est déjà contenue dans H *alors :* Solution ignorée, *Sinon :* Ajouter la solution à H
Étape 9 : Si $G_{iter} > G_{iter-1}$ *alors* continuer le calcul à partir de $Solution_i$ *Sinon* Continuer le calcul à partir de la $solution_{iter-1}$
Étape 10 : Vider T.
Étape 11 : iter + +.

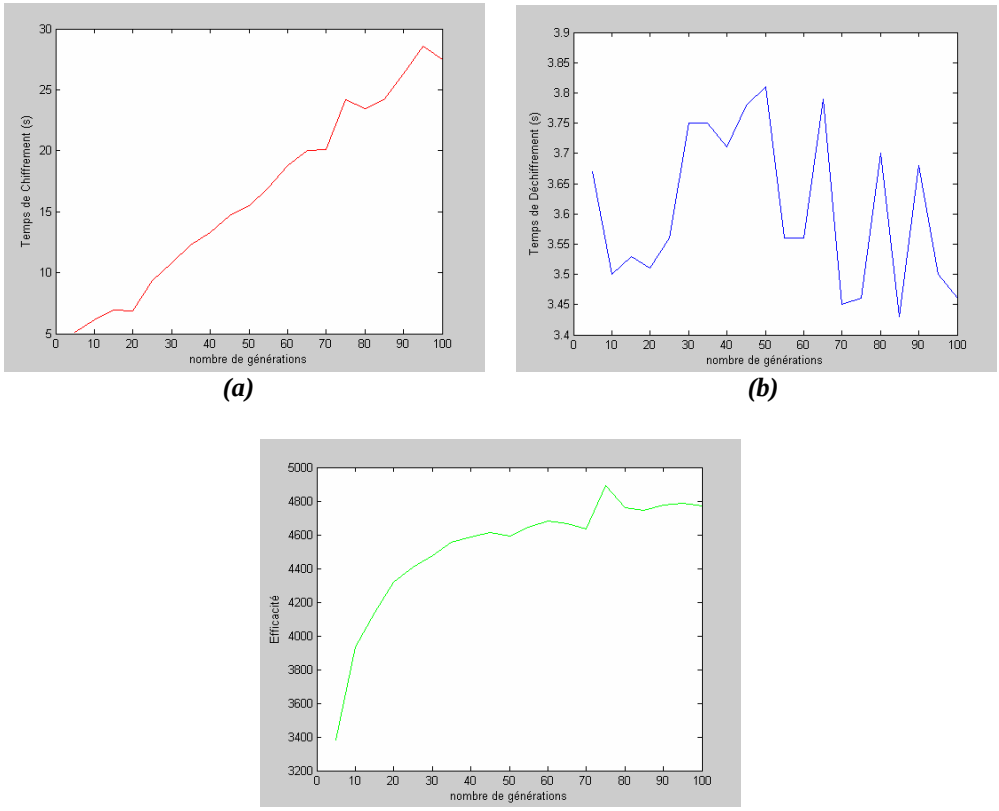
Nous précisons, dans cette section, les valeurs des paramètres de *TabuCrypt* ainsi que les résultats de son application sur les données modèles précédemment présentées (Texte1, Texte2, Image Lena et image Logo).

V.3.10.1. Réglage des paramètres

Les principaux paramètres de *TabuCrypt* concernent le nombre de générations ou d'itérations de l'algorithme et le nombre de voisins seront présentés.

Après *IterMax* itérations, fixée expérimentalement, la procédure de recherche tabou s'arrête et fournit la meilleure solution trouvée. La figure V.1 récapitule la moyenne des différentes valeurs de test et leurs impacts sur le temps de convergence et la qualité de la solution trouvée représentant une version chiffrée de l'image de test Lena.

D'après les résultats résumés dans la figure, ci-dessous, nous constatons que la meilleure efficacité a été obtenue pour un nombre de générations égale à 75 après un temps de chiffrement de 24.20 s et un temps de déchiffrement de 3.46 s. De même, les nombres de générations 80, 85, 90, 95 et 100 ont donné de bonnes valeurs d'efficacité (4760, 4744, 4777, 4789 et 4771, respectivement), mais leurs temps de chiffrement sont assez longs (23.46 s, 24.18 s, 26.29 s, 28.54 s et 27.48 s, respectivement).



(a) ***(b)*** ***(c)***

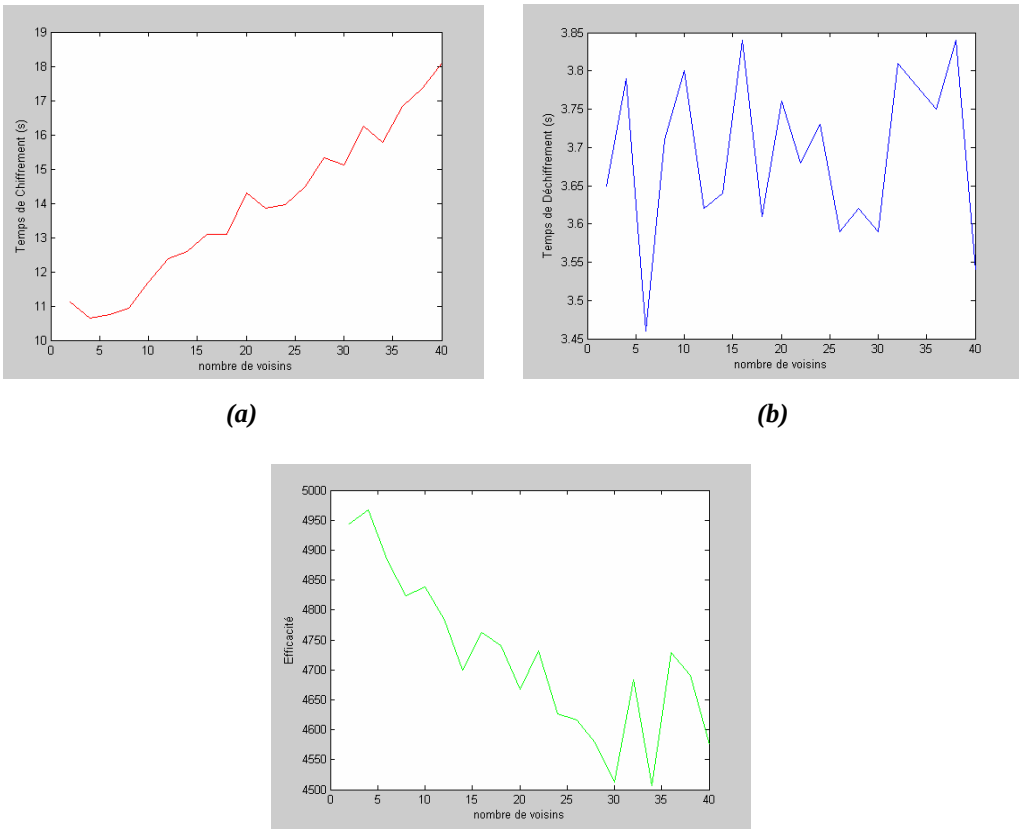
Figure V.1. *Résultats obtenus suivant les différentes valeurs de test de nombre d'itérations (générations) :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Toutefois, les résultats obtenus, pour le test avec 45 générations, sont acceptables que se soit sur le plan efficacité (4612) ou temps de chiffrement (14.70 s). D'autres valeurs de nombre de générations (55, 60, 65 et 70) ont donné des résultats d'efficacité de même ordre que celle du nombre de générations 45, mais après des temps de chiffrement relativement plus longs (29.93, 28.76, 20.04 et 20.12, respectivement). Ainsi, la valeur de *IterMax* a été fixée à 45 générations.

L'autre paramètre à régler est le nombre de voisin. Pour ce faire, nous avons utilisé le même mécanisme que celui utilisé pour régler le paramètre relatif au nombre d'itérations. Les résultats des différents tests sont résumés à travers la figure V.2 en donnant l'influence des différents nombres de voisins testés sur le temps de convergence et l'efficacité de la solution trouvée.

D'après le contenu des figures V.1 et V.2, la meilleure efficacité (4967) a été obtenue avec un nombre de voisins qui vaut 4 après un temps de chiffrement assez raisonnable et représentant le plus petit temps de calcul sur l'ensemble des tests effectués (10.64s).

Ainsi, nous avons opté à régler le paramètre relatif au nombre de voisins en lui attribuant la valeur 4.



(a) ***(b)***

(c)

Figure V.2. *Résultats obtenus suivant les différentes valeurs de test de nombre de voisins :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Le tableau suivant (tableau V.1) résumé les valeurs de paramètres adoptés par *TabuCrypt.*

	Valeurs des paramètres de*TabuCrypt*
Nombre de voisins	4
Nombre de générations	45

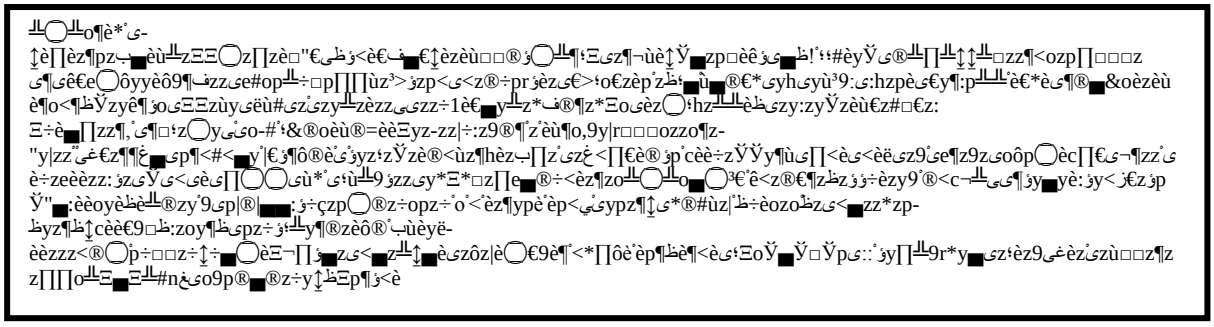
Tableau V.1. *Valeurs adoptées pour les paramètres de TabuCrypt.*

V.3.10.2. Résultats

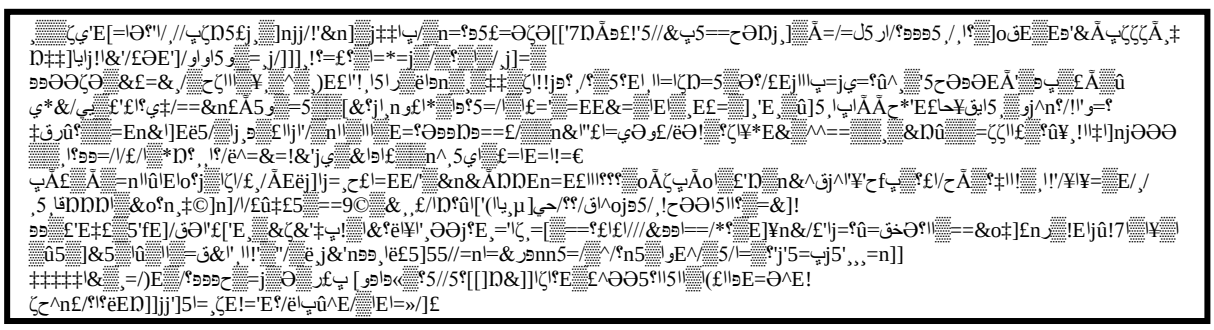
A ce niveau nous présentons les résultats de chiffrement par *TabuCrypt* des données tests. Deux versions chiffrées de chacune des données tests sont présentées accompagnées des clés générées dans chaque cas. Le tableau V.2 résume les temps de chiffrement et de déchiffrement ainsi que l'efficacité de chacun des exemples.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure V.3. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets)

728	898	353	668	901	875	906	437	425	40	739	524
797	890	877	786	357	654	889	258	566	884	883	899
399	721	896	133	905	483	410	895	735	117	900	876
878	303	479	907	706	879	100	813	91	453	152	634
874	186	597	902	480	341	99	42	736	438	456	893
794	344	881	267	887	474	617	191	885	56	892	886
897	377	601	904	34	98	672	880	903	673	496	760
373	882	894	888	347	463	750	891	520	248	465	571
624	116	573	744	866	646	926	68	1228	131	311	532
299	518	139	1134	1120	734	714	290	22	97	541	1047
1234	961	1110	819	52	153	255	1006	560	960	1077	942
1109	154	836	30	412	1119	1202	1037	1003	383	1238	335
392	1035	1241	119	448	409	1115	266	1036	857	971	194
713	912	422	378	500	782	274	1048	1102	651	1129	1091
810	696	1040	994	321	362	46	563	1148	149	339	858
503	328	110	770	1053	773	272	598	77	1087	1187	1121
551	418	225	801	1185	1082	955	557	737	506	464	1070
1240	168	187	843	796	82	867	450	200	764	190	982
1076	967	873	260	695	224	1018	207	1049	277	3	652
792	1014	1213	936	466	621	829	2	400	1085	38	23
1159	570	509	816	973	164	205	1201	112	1078	870	333
364	1212	350	250	102	536	172	629	705	178	381	722
283	1100	823	426	84	122	134	1128	591	361	798	24

1224	1111	179	1143	526	1219	1166	176	834	10	1221	1093
1062	1232	167	587	647	342	490	1041	1114	785	525	1020
473	1233	856	562	423	579	527	427	1194	215	653	242
1237	767	795	1153	862	952	998	644	1081	461	182	232
29	539	1025	1223	434	411	320	932	1204	499	11	546
1112	558	958	1160	486	467	1039	436	1245	1151	421	920
126	692	174	802	581	1066	852	355	1023	180	125	821
1104	129	265	1094	158	1101	824	871	555	1156	106	681
1145	645	803	1208	114	1068	1080	1222	269	996	1074	263
1064	1075	1184	213	699	123	312	1056	1130	175	468	754
380	1063	616	234	1200	768	865	1154	108	1069	631	698
360	1207	397	356	711	540	484	1180	1215	941	1220	596
604	1244	774	916	1050	749	420	169	1029	1242	424	376
1107	1178	820	352	372	118	443	818	701	1136	690	950
1196	93	724	543	1155	476	1034	963	196	978	1086	472
1230	781	610	273	1125	1137	531	120	554	264	969	763
165	407	390	1209	583	49	648	183	331	104	985	130
1126	1152	977	43	917	508	614	992	428	1067	582	983
987	454	497	230	530	285	374	1097	758	308	354	495
664	128	825	441	990	288	951	1131	1008	1004	632	121
752	249	105	682	1195	460	1162	981	746	297	228	700
1116	1095	296	83	1164	1198	528	502	27	762	769	1139
655	863	514	1028	276	811	1177	663	432	12	1002	716
826	96	59	458	1225	1181	136	416	830	1072	937	209
94	986	784	940	146	658	204	976	945	918	185	1012
643	611	1169	1142	1079	1173	793	457	1191	393	212	635
239	984	261	231	1010	1186	1203	576	1124	1140	404	738
991	33	1163	358	586	1098	324	968	1083	642	972	76
1013	1182	1092	751	16	88	817	957	569	135	1117	535
622	670	1009	291	832	329	28	979	72	236	1144	462
298	79	626	772	221	238	487	559	1236	556	946	633
702	69	235	1005	109	140	319	452	1054	155	1019	304
485	756	512	74	398	216	1210	925	1171	911	1243	842
997	478	1239	293	921	429	160	933	1088	431	145	1229
661	295	322	35	1055	1043	1016	201	1150	637	620	482
369	790	253	640	719	491	1061	1042	237	113	914	1122
64	348	171	727	233	254	522	206	1174	81	115	511
1193	928	1218	346	931	405	1	613	1214	394	709	592
359	307	553	708	850	163	999	166	39	1011	370	949
710	1113	1157	935	1022	860	147	78	948	198	1211	177
330	19	771	211	684	993	677	980	1206	927	723	804
1065	970	1044	1099	729	523	442	908	1217	1165	159	9
828	995	150	244	1231	15	325	137	1189	1205	844	691
922	964	564	89	281	243	671	218	657	1000	1172	939
1059	1032	683	649	783	989	757	606	385	965	470	954
389	195	959	806	1060	615	676	630	219	1170	1052	294
257	864	869	371	715	301	840	318	1118	1158	268	1027
627	435	1103	47	1015	1175	975	1192	313	1084	1132	529
67	787	953	1167	1135	924	919	1073	1031	1058	853	271
384	471	812	947	18	1161	765	567	934	217	66	759
545	475	8	1147	505	1001	1045	599	1141	1235	1176	584
284	1216	929	704	1089	943	501	717	445	966	693	605
259	1168	733	743	747	1138	550	336	343	53	814	1133
144	913	439	1051	872	1127	909	1024	459	915	17	504
944	1096	1046	73	726	1105	1026	956	1188	718	549	1146
48	910	494	189	222	1017	827	334	612	585	745	678
208	580	974	938	1123	396	1190	1030	367	55	337	366
962	246	21	4	777	1179	1149	1227	220	930	1226	1021
800	659	1183	731	202	1197	403	1090	988	533	379	1057
1106	1071	515	282	451	1038	574	1033	1199	226	1108	923
1007	280	111	544	406	61	748	1274	402	753	181	1374
143	306	868	1264	6	193	489	618	1351	667	1346	1375
1267	32	732	687	286	688	507	363	1308	859	95	575
855	1376	345	1358	65	1361	1257	565	44	1371	214	703
447	36	561	1329	799	203	625	720	278	1287	845	210
444	1280	1327	1303	414	1268	1273	846	477	446	1253	578
251	1377	822	229	1328	1295	847	419	1320	837	1378	831
1373	1362	538	1379	662	1254	1334	778	1331	58	45	602
87	287	854	839	1292	780	1322	309	619	401	279	588
849	1350	332	791	351	156	1247	1355	639	157	132	1380
1282	1270	1381	641	375	730	1300	498	388	594	519	1382
689	1313	1258	542	595	162	1383	493	275	252	5	1367
1348	368	776	227	1319	675	413	1338	20	1318	188	338
516	548	568	590	1294	415	1343	609	417	1356	54	1315
223	1323	1316	1357	685	1246	1384	433	41	815	1360	1296
755	1345	316	808	650	310	449	1284	623	1347	1302	101

1364	481	1293	142	170	517	680	1314	1324	1281	1385	317
1321	572	833	1261	141	62	697	305	1330	51	173	1312
669	148	1256	656	92	391	161	382	1251	600	665	805
686	1342	589	1369	534	349	1249	838	1332	1336	521	1272
387	1262	1340	1301	70	552	192	1386	1311	603	1277	766
440	789	638	292	37	1279	1291	1372	725	1387	247	851
1299	86	1388	674	848	1310	1359	57	138	1286	124	1285
1266	1248	809	1252	340	289	1389	775	488	71	694	607
31	1349	628	1263	1390	1255	365	1341	300	1275	262	1309
1250	761	25	50	85	408	256	577	835	841	742	1370
707	712	666	1283	1354	455	1317	386	430	1269	492	1339
14	660	1288	314	1368	1306	151	1353	807	90	1265	788
608	510	1391	1335	327	1304	741	7	1365	197	245	861
1297	1259	80	679	1333	75	1326	1352	779	26	199	270
547	537	636	1289	740	323	326	395	1298	60	1344	302
1392	1337	1271	593	127	63	184	13	513	103	1307	469
1290	1393	1260	1366	1305	1325	1278	1276	107	241	315	240
1363											

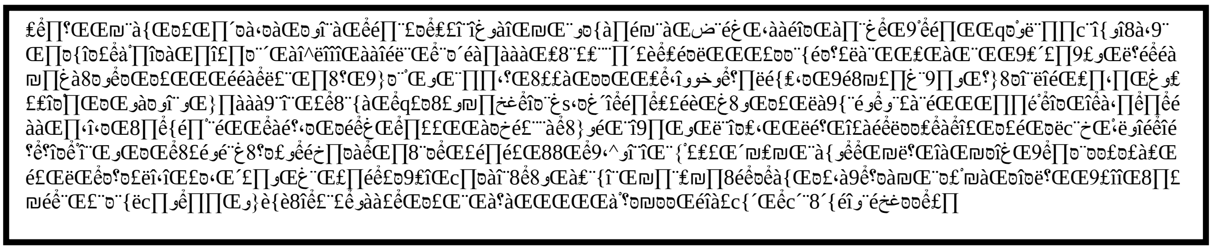
Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets)

7	36	33	79	39	24	81	16	5	73	3	1
20	88	30	44	56	41	11	62	65	47	98	26
814	806	645	389	313	988	737	838	346	946	243	890
300	123	778	333	207	229	663	571	127	816	821	355
1220	1344	1025	1270	1147	1345	1020	1218	1338	1037	1085	1041
1016	1226	1114	1073	1346	1190	1195	1253	1227	90	40	70
15	95	6	53	87	34	91	25	82	76	45	13
93	14	55	28	37	17	72	66	85	61	22	64
21	51	8	77	35	78	86	38	10	42	18	99
31	60	2	27	54	92	94	80	19	71	69	29
67	46	84	89	12	9	96	4	57	58	63	49
59	75	50	23	83	48	526	275	178	752	965	810
453	272	436	898	463	245	321	586	113	506	208	419
708	183	640	451	376	195	372	414	942	895	238	302
443	631	474	712	434	917	277	392	187	467	607	557
845	647	688	188	52	97	630	680	865	449	583	164
262	105	634	317	639	119	928	464	714	802	797	932
823	822	953	684	529	694	270	991	482	154	956	978
440	135	197	670	280	351	987	248	881	779	933	327
606	214	662	695	292	456	909	608	158	285	491	877
311	771	395	891	938	803	563	612	561	963	715	501
447	153	498	134	792	220	215	811	874	427	283	418
538	68	32	43	74	687	345	661	924	284	486	359
839	495	115	893	732	103	307	442	305	675	600	136
157	603	106	417	122	918	441	785	930	982	181	466
255	758	765	794	817	271	854	597	494	900	219	198
484	315	194	218	923	382	667	309	901	408	120	558
204	772	336	761	470	266	519	996	240	582	795	969
168	142	541	559	540	853	564	578	936	916	788	882
274	861	527	365	651	367	707	665	162	435	469	929
589	979	939	290	753	755	776	584	656	269	391	416
860	700	786	514	344	566	492	319	993	139	384	790
256	475	252	437	244	145	276	544	374	598	373	828
258	652	870	685	784	318	801	927	353	109	180	324
689	858	910	108	940	508	525	137	749	697	975	296
489	650	503	329	101	570	326	850	716	581	246	627
403	782	375	556	985	949	126	907	422	179	282	221
161	944	981	604	517	349	212	783	138	906	835	873
594	545	880	655	462	967	539	457	203	528	717	952
341	316	366	813	591	836	299	769	926	951	547	904
848	844	840	800	744	576	819	995	412	413	147	588
812	424	356	404	554	869	619	186	390	150	692	825
740	852	343	601	831	431	312	989	535	481	377	702
764	286	855	682	976	261	515	768	750	254	696	883
654	169	304	618	889	727	710	805	362	799	360	621
543	530	288	428	379	118	587	531	423	833	426	210
342	241	905	787	572	439	560	954	748	653	402	773
394	980	448	649	223	488	760	415	368	959	611	913
287	493	569	729	253	767	132	322	536	479	553	476
648	410	815	964	562	458	323	217	385	830	144	234
173	617	643	143	660	628	461	551	473	193	337	257
644	698	673	293	104	872	757	334	843	826	510	452
289	107	573	213	251	459	516	512	110	998	438	242
974	971	314	471	669	550	620	465	148	580	593	777

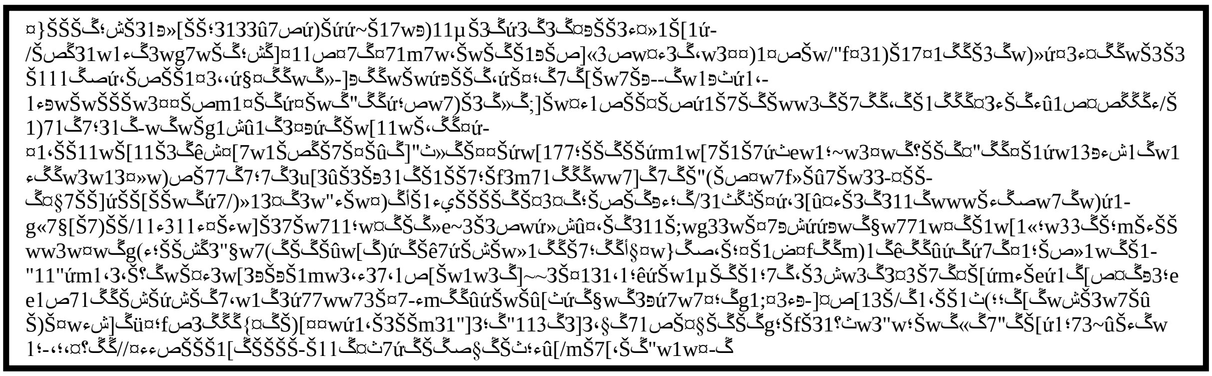
192	990	155	230	614	504	638	983	915	775	842	149
236	363	759	999	140	745	851	885	736	766	941	294
871	867	994	152	914	407	396	555	690	886	585	896
369	613	731	966	249	774	460	170	505	331	970	721
846	807	892	295	577	298	804	961	191	796	948	159
590	791	674	117	742	720	738	490	405	664	599	184
432	166	509	762	672	986	615	542	306	165	130	622
455	728	683	445	820	658	957	605	177	279	176	335
859	129	364	706	202	879	809	502	348	383	657	477
281	947	832	671	857	868	546	595	112	268	233	724
741	228	713	878	908	167	370	887	925	100	325	704
200	211	499	681	626	574	677	763	433	701	624	446
637	354	111	960	841	977	125	483	444	478	972	711
131	808	430	133	387	380	163	468	911	310	897	864
160	513	227	141	945	518	352	718	984	524	487	425
116	781	411	278	636	876	128	320	532	330	629	725
834	454	250	114	291	668	520	534	849	770	185	358
623	973	709	409	303	146	579	259	955	429	175	747
397	950	659	679	239	856	642	610	609	568	301	958
496	297	533	121	937	480	201	730	552	818	500	225
400	182	521	328	894	922	935	522	686	888	189	931
199	511	754	406	124	232	102	332	920	361	398	693
224	386	399	592	350	739	567	703	265	226	919	263
899	222	472	746	156	635	378	507	206	171	273	827
401	866	565	548	338	789	421	992	209	151	676	641
485	884	699	260	393	523	902	388	726	997	968	190
381	751	633	231	264	756	824	357	420	625	450	743
196	216	602	371	237	962	174	235	735	875	837	172
863	829	666	247	719	912	793	339	847	943	646	734
934	347	798	340	205	575	733	267	537	903	678	308
616	862	596	632	549	691	705	723	497	921	780	722
1229	1311	1126	1074	1105	1179	1155	1223	1023	1161	1175	1129
1184	1134	1305	1319	1088	1082	1347	1079	1281	1152	1144	1149
1288	1176	1186	1289	1254	1283	1348	1017	1159	1189	1349	1217
1131	1039	1350	1272	1064	1192	1188	1228	1115	1095	1264	1310
1058	1201	1198	1022	1014	1043	1108	1071	1351	1352	1027	1165
1151	1059	1028	1353	1141	1200	1143	1354	1247	1100	1029	1112
1005	1355	1257	1083	1194	1046	1356	1035	1024	1307	1101	1091
1278	1287	1092	1139	1120	1250	1007	1246	1225	1116	1051	1357
1244	1086	1209	1096	1090	1284	1358	1106	1127	1318	1309	1290
1193	1216	1076	1052	1259	1359	1117	1360	1361	1239	1047	1034
1003	1057	1277	1111	1093	1332	1301	1009	1183	1328	1178	1061
1099	1265	1215	1240	1362	1213	1269	1298	1049	1312	1363	1171
1263	1252	1364	1197	1077	1243	1365	1366	1145	1214	1142	1018
1315	1123	1146	1107	1060	1367	1113	1006	1160	1337	1249	1326
1011	1279	1368	1208	1196	1275	1056	1170	1110	1234	1335	1261
1294	1069	1031	1203	1063	1267	1258	1300	1202	1157	1241	1030
1150	1280	1206	1369	1181	1180	1021	1370	1158	1121	1097	1205
1128	1012	1293	1089	1341	1237	1187	1317	1102	1010	1068	1304
1371	1291	1372	1148	1153	1065	1256	1303	1306	1124	1295	1053
1013	1162	1268	1044	1373	1078	1032	1166	1177	1137	1026	1255
1374	1375	1133	1236	1191	1172	1048	1245	1282	1119	1296	1062
1130	1308	1376	1207	1299	1036	1377	1333	1136	1199	1378	1379
1292	1164	1045	1248	1343	1380	1314	1235	1321	1066	1232	1182
1271	1336	1320	1260	1251	1381	1382	1168	1286	1383	1384	1316
1322	1385	1042	1262	1173	1212	1038	1386	1040	1274	1154	1327
1273	1185	1387	1313	1210	1388	1019	1132	1033	1001	1156	1389
1122	1070	1098	1211	1055	1329	1323	1285	1072	1054	1325	1004
1302	1118	1231	1219	1204	1087	1222	1221	1125	1080	1342	1224
1015	1340	1390	1050	1330	1233	1094	1081	1067	1104	1174	1297
1391	1331	1075	1392	1339	1238	1230	1135	1324	1276	1163	1334
1103	1084	1002	1138	1167	1109	1242	1140	1008	1266	1169	1000
1393											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبعة الحال

(a)



(b)



(c)

Figure V.4. (a) *Donnée originale Texte2*, ***(b)*** *Première version chiffrée*, ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 ($Taille_{Clé}$ = 1,8704 K octets)

8	2	4	7	6	9	1	5	811	910	899	896
878	885	622	892	767	914	357	901	895	297	771	211
70	851	189	890	897	694	880	913	662	78	908	900
906	210	170	889	879	904	833	407	208	300	498	306
259	891	871	858	469	698	543	87	903	368	52	884
266	835	642	886	92	909	905	882	883	405	527	371
915	658	12	893	113	172	457	898	888	633	887	242
877	881	414	376	911	556	912	907	902	163	648	281
894	876	949	701	174	1155	643	192	649	213	466	1103
514	1076	1022	660	624	3	632	430	623	944	925	770
785	796	67	1194	230	707	799	1177	1218	245	508	1007
179	484	1209	1223	549	1118	1037	1167	740	73	1160	248
193	1088	287	1200	304	131	175	544	511	428	1173	176
1176	931	983	815	951	370	32	955	826	53	945	754
168	863	1009	1219	692	532	459	273	1018	356	730	255
636	604	539	11	984	79	930	817	383	581	829	1148
1133	764	961	1068	1127	1140	1225	630	956	403	1143	1235
822	601	755	1217	535	1073	960	1137	203	325	545	1036
745	1112	971	1097	1168	1232	223	437	805	1062	1204	321
272	374	1080	260	748	86	1045	434	429	1157	675	994
145	1142	587	585	1229	1069	1044	267	386	471	1163	147
1102	1182	747	593	798	1054	1019	82	981	978	348	1066
928	669	1192	1065	1038	311	1196	943	1141	597	610	63
741	1085	1075	234	1063	962	850	682	1156	401	843	670
81	1058	84	1000	151	525	60	1110	225	501	969	552
341	446	486	985	1093	580	372	1187	998	517	65	778

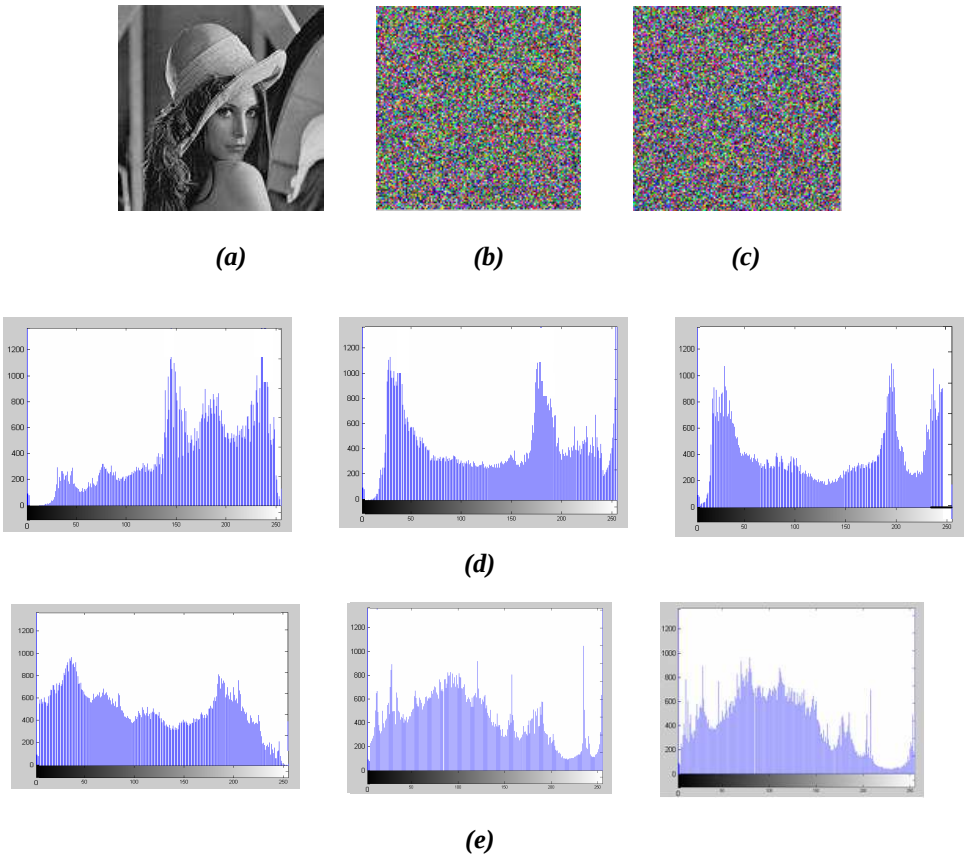
606	137	97	938	180	952	1230	90	35	1067	1084	1083
1014	463	237	814	650	319	935	379	315	309	46	676
1139	1060	1238	240	1152	645	689	846	221	1003	251	116
635	1215	182	28	467	460	171	495	377	674	80	1228
1027	392	452	54	502	957	1064	1079	228	733	932	359
609	997	647	834	1149	1004	751	166	923	830	1166	976
980	1198	1116	975	578	631	75	253	235	1095	1224	860
965	449	1109	43	979	150	763	202	271	252	394	59
516	1205	239	838	874	247	353	685	704	169	83	177
1012	389	114	39	1074	659	924	982	104	565	783	839
705	167	1222	987	656	292	57	93	194	1056	1165	282
861	500	526	564	34	411	922	1189	566	445	506	99
1119	793	257	637	398	1087	1122	1185	105	582	51	494
490	1024	199	207	696	782	596	521	412	138	50	1233
1030	993	1191	258	653	1023	1105	312	757	492	712	419
173	1055	406	948	21	713	119	378	402	929	1186	574
19	654	286	1098	327	246	852	1017	1039	183	592	936
1082	954	332	714	781	328	666	85	518	620	787	1033
20	920	347	750	546	756	1034	1181	1031	62	1035	497
22	522	519	346	1051	591	803	373	1239	1015	927	37
382	966	586	218	140	1124	410	1183	74	595	1180	942
841	1170	872	1125	742	1243	845	316	1104	106	554	159
802	967	1099	710	344	158	196	939	1195	1091	146	614
569	1002	562	1226	641	279	567	416	1178	100	1132	236
823	49	809	296	360	254	575	1013	1048	313	441	801
206	992	1193	157	48	723	1234	280	184	768	615	415
691	824	1213	762	743	753	187	233	404	262	479	1016
422	365	481	1171	661	837	323	963	686	772	605	1227
599	572	777	1072	443	717	1244	1135	1131	735	448	1047
109	657	941	1078	1071	531	865	794	1032	25	107	1100
238	621	120	334	71	795	1077	1001	1159	626	1161	451
342	23	699	1040	919	1145	510	336	715	274	739	468
1231	1172	1237	831	1150	1144	423	14	832	1188	31	1162
68	918	385	338	766	867	1050	528	216	305	577	524
972	709	1081	222	639	399	72	1212	953	1216	124	112
432	1203	1201	126	570	1146	1211	101	426	125	1136	178
209	706	1005	1174	1086	921	926	33	917	358	1011	161
977	450	42	1057	688	728	1111	847	219	456	322	970
197	999	1199	720	366	1175	201	627	1130	320	536	290
1236	995	229	868	478	1214	330	589	1008	625	355	690
968	310	973	337	488	1006	672	77	148	746	808	603
608	1070	343	667	553	354	752	989	946	520	568	628
1061	725	693	1089	110	1117	1151	326	629	1108	1092	294
855	759	16	792	1208	651	391	776	58	1164	256	800
263	916	1120	1179	424	579	547	1115	369	160	974	538
15	89	541	934	947	18	480	1049	1029	465	612	191
958	1028	986	1134	774	548	1138	1210	1197	937	1221	806
1096	542	613	1123	433	678	804	726	76	933	1090	571
1158	1043	959	1094	1041	10	1052	1106	1020	1169	724	1241
862	990	988	269	461	644	711	1153	421	602	677	684
1107	594	1206	1242	164	853	1021	475	277	1113	214	1025
1053	440	491	573	1207	1059	683	134	139	1046	318	864
285	1121	560	1026	1154	397	996	190	509	1190	350	702
1128	417	1042	438	950	673	1010	1114	515	991	665	810
91	869	964	493	188	395	108	844	152	769	335	827
1101	1129	88	435	64	472	1147	807	551	56	1126	719
1202	791	857	136	779	217	1240	444	1184	499	409	1220
668	224	940	299	1343	697	1344	616	442	95	1336	1334
1286	204	646	836	220	489	301	700	1313	288	820	561
773	1297	1345	156	607	473	476	1259	721	133	98	840
1346	351	464	559	1347	384	1348	1304	375	1349	563	1268
436	873	652	439	142	731	1264	314	111	681	393	611
784	1302	558	1270	1350	205	1351	165	13	303	361	1269
1352	734	618	381	732	363	1353	1354	664	557	380	1294
513	94	530	727	1291	244	1355	1338	367	425	454	758
333	1356	1300	1357	1248	339	1342	1341	588	1282	250	317
760	122	1308	512	1358	69	1299	129	102	1251	1359	534
331	788	1298	396	1360	268	264	103	427	482	1325	1307
523	1361	329	1324	1362	1363	1290	1364	859	455	155	663
1340	1333	819	261	1365	1366	1329	1328	786	293	284	1367
154	1331	270	1265	849	132	27	135	729	130	749	505
1368	231	638	1369	390	1370	345	540	1371	1278	1293	276
1372	1250	308	737	17	1373	241	298	362	1262	198	127
447	153	474	1272	1252	825	349	121	1315	1303	215	47
1337	1305	249	1260	186	775	671	1320	387	1374	291	870
289	1375	400	41	227	1289	144	30	55	1339	744	1316

1296	128	1319	504	1376	1245	1377	529	1378	848	431	485
1255	418	1254	708	352	1267	212	232	640	1246	1314	458
1379	722	1327	1380	634	181	118	797	123	590	1280	679
1381	1326	1258	195	1271	507	38	1382	66	1279	453	1256
1309	1383	738	483	340	655	736	533	1312	1384	420	364
842	149	780	1263	1332	1275	1274	283	816	1284	1330	1266
866	1385	117	598	818	703	200	716	408	1285	854	576
295	1386	1292	115	695	1249	1318	40	185	302	1306	1283
1288	45	487	1323	875	143	600	477	583	1281	1253	550
1387	278	619	828	324	1257	813	812	24	790	503	496
36	1388	1389	243	96	1301	1321	718	555	789	1247	388
26	1261	821	1276	162	761	1322	687	275	765	1390	1287
617	1310	141	265	44	1277	226	1391	1392	1311	29	1393
1317	856	584	1295	680	537	462	307	413	1273	61	1335
470											

Clé de chiffrement de la deuxième version chiffrée de Texte2 ($Taille_{Clé}$ = 1,8704 K octets)

681	257	54	272	127	416	212	193	126	445	404	584
108	735	917	617	694	270	386	241	702	562	898	175
381	459	904	737	596	689	130	228	326	836	460	289
37	958	638	703	356	817	741	466	971	858	1	909
688	524	53	13	448	624	634	199	369	807	116	686
102	121	204	490	397	187	670	650	522	163	256	950
868	720	153	510	660	580	68	427	698	546	523	900
621	306	312	331	721	488	314	243	350	313	452	325
844	986	82	499	494	48	12	991	526	453	479	910
978	56	409	607	172	240	620	627	391	779	15	641
174	931	71	90	55	265	963	947	981	60	84	250
95	39	977	513	748	220	235	516	507	916	401	190
360	213	362	725	447	333	105	491	857	78	881	383
89	934	705	999	123	426	231	293	323	91	943	557
803	846	324	945	791	718	826	903	396	258	839	602
713	83	859	38	639	411	400	201	359	545	487	92
138	766	879	799	346	558	928	864	49	961	365	519
149	146	371	18	281	158	430	684	277	885	754	58
374	435	269	318	358	850	335	953	567	443	970	899
169	719	291	789	809	613	432	595	275	882	247	347
62	282	851	685	97	775	151	814	601	384	439	771
980	343	552	598	106	853	541	192	612	993	687	154
271	461	496	125	708	776	948	744	939	375	232	952
286	927	675	671	483	575	373	455	155	305	421	761
731	279	869	288	642	933	788	367	122	676	46	997
35	31	830	902	873	480	736	402	751	477	336	654
450	704	696	295	236	534	244	944	988	629	230	861
341	100	433	906	786	390	750	905	307	42	351	659
714	301	395	34	205	549	77	438	215	285	454	662
871	883	949	884	165	573	935	398	728	112	888	5
985	304	412	440	505	514	219	292	920	131	965	72
349	509	833	287	157	707	308	47	472	810	492	144
339	489	734	139	233	223	474	261	533	209	987	656
912	994	475	911	968	98	772	506	553	214	610	469
132	202	159	568	925	983	938	237	224	669	777	693
538	964	891	746	431	464	604	227	722	756	773	554
50	527	142	252	529	334	171	712	465	437	515	255
874	57	758	540	161	760	733	503	792	27	486	532
822	385	462	234	407	299	583	210	446	88	544	757
436	319	700	109	478	614	393	732	246	599	556	316
682	753	767	70	586	423	796	361	570	793	177	913
73	8	414	30	372	81	709	429	456	79	64	936
508	878	578	63	780	329	442	135	110	457	597	226
870	512	876	458	812	866	266	942	765	768	302	26
330	69	76	537	724	907	3	946	608	500	561	872
4	424	23	535	865	181	120	493	539	315	160	93
589	588	611	701	200	216	661	531	419	115	677	273
399	982	380	188	890	382	24	377	749	655	194	922
673	87	653	823	267	740	85	221	44	582	310	189
813	956	834	695	995	366	410	889	600	353	140	425
590	783	838	908	672	185	819	806	918	501	892	816
955	229	930	128	778	536	542	413	560	484	559	504
975	99	941	186	867	420	984	476	518	66	606	622
405	976	141	937	592	797	378	631	485	211	33	566
666	521	723	28	635	297	957	441	7	683	251	992
626	511	101	591	21	665	716	636	996	551	548	274
585	363	979	332	96	117	609	249	129	923	644	710

637 517 738 51 43 357 969 563 222 203 322 547
470 860 344 798 355 863 248 628 921 845 841 184
847 502 65 565 134 886 848 951 940 471 594 406
774 818 663 45 468 14 645 972 303 959 770 337
855 260 124 752 824 623 619 574 451 781 962 9
825 368 498 640 196 428 877 473 166 197 463 354
415 764 118 815 587 739 276 183 67 321 94 298
564 577 136 785 897 801 444 36 808 364 895 657
820 759 179 311 495 658 147 742 376 691 16 579
167 896 370 434 625 804 467 571 805 745 387 379
664 6 2 690 652 795 967 300 422 726 827 837
572 191 831 61 150 901 225 706 924 327 729 593
320 253 17 152 543 667 954 715 254 893 697 137
263 180 11 143 74 990 19 278 875 763 119 345
605 264 408 22 576 699 802 497 309 854 929 86
680 678 842 168 482 113 394 649 679 727 296 782
835 787 615 618 338 389 25 743 692 52 550 843
114 919 207 828 80 40 284 178 730 10 238 966
242 569 794 832 103 755 392 182 75 481 849 647
821 259 974 104 294 880 418 973 603 674 998 530
616 59 32 262 348 651 195 20 133 340 894 632
156 388 403 643 829 176 581 960 29 245 198 217
762 711 717 41 239 862 747 668 148 914 555 206
449 926 280 352 417 111 932 852 342 784 633 173
887 989 162 107 218 170 328 646 208 790 915 528
290 811 769 630 648 317 840 520 283 145 164 268
525 856 800 1319 1015 1163 1032 1115 1334 1035 1004 1255
1286 1101 1309 1186 1325 1012 1335 1304 1336 1136 1337 1107
1172 1113 1249 1190 1227 1058 1010 1251 1111 1023 1338 1127
1302 1114 1033 1100 1228 1128 1112 1031 1020 1299 1109 1311
1339 1226 1081 1202 1076 1054 1295 1340 1069 1341 1198 1216
1342 1183 1343 1002 1148 1070 1176 1328 1303 1003 1121 1029
1277 1229 1089 1230 1256 1091 1175 1060 1042 1119 1182 1188
1074 1050 1132 1224 1333 1170 1078 1294 1168 1344 1149 1320
1239 1162 1203 1223 1052 1310 1269 1233 1210 1083 1265 1307
1275 1298 1045 1326 1345 1346 1272 1048 1317 1347 1261 1030
1263 1049 1285 1165 1348 1276 1349 1350 1027 1201 1235 1262
1189 1351 1313 1005 1063 1017 1195 1000 1150 1352 1016 1353
1062 1305 1130 1024 1354 1221 1073 1355 1179 1126 1356 1205
1161 1244 1240 1001 1129 1022 1197 1284 1087 1080 1006 1291
1137 1147 1152 1055 1181 1268 1252 1250 1327 1053 1160 1044
1134 1357 1072 1037 1151 1040 1225 1093 1008 1358 1056 1192
1246 1359 1271 1360 1047 1315 1090 1361 1013 1156 1362 1204
1157 1041 1082 1096 1184 1105 1123 1329 1363 1281 1097 1118
1241 1215 1364 1146 1288 1258 1365 1131 1366 1034 1092 1297
1300 1098 1367 1280 1368 1248 1018 1142 1245 1120 1177 1211
1267 1369 1173 1370 1371 1057 1372 1122 1214 1159 1237 1138
1021 1038 1166 1217 1064 1180 1208 1283 1106 1065 1185 1373
1278 1314 1374 1200 1322 1071 1220 1219 1316 1145 1169 1222
1375 1293 1067 1059 1104 1019 1260 1242 1257 1133 1376 1125
1043 1377 1167 1378 1379 1308 1140 1079 1085 1077 1232 1103
1380 1290 1381 1154 1331 1124 1266 1199 1253 1116 1270 1209
1158 1036 1102 1046 1382 1306 1095 1007 1088 1231 1108 1193
1238 1206 1039 1236 1187 1274 1383 1282 1094 1171 1384 1075
1264 1084 1254 1196 1259 1164 1155 1025 1385 1178 1386 1068
1086 1387 1318 1243 1207 1026 1321 1324 1323 1028 1213 1061
1218 1191 1212 1330 1388 1066 1332 1234 1117 1051 1153 1301
1273 1099 1389 1390 1144 1139 1143 1009 1247 1287 1014 1135
1391 1279 1174 1141 1392 1312 1110 1292 1296 1289 1011 1194
1393



(a) ***(b)*** ***(c)***

(d)

(e)

Figure V.5. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

369	423	529	171	397	412	474	15	689	64	112	390
267	21	118	304	6	28	574	465	356	338	717	638
339	110	243	458	508	450	361	608	635	115	593	749
151	742	575	716	428	109	486	584	294	722	521	141
87	695	373	225	452	602	679	13	208	85	468	634
709	298	70	297	514	601	439	359	320	251	222	292
319	765	456	614	142	645	625	288	383	217	125	476
396	718	274	236	729	483	567	114	487	492	303	542
426	628	675	713	296	505	370	730	409	395	764	641
669	460	337	264	230	204	623	358	293	418	150	520
134	328	469	762	440	533	651	551	535	286	438	462
103	355	347	63	158	16	561	205	410	643	703	432
519	693	247	620	346	40	736	366	129	652	164	327
371	506	363	154	241	447	434	624	585	163	748	376
380	144	407	59	143	667	252	10	536	548	766	360
335	455	311	497	200	466	249	751	35	119	165	341
36	741	398	448	656	18	65	233	734	17	531	512
595	660	104	586	81	435	299	101	92	309	454	436
710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579
352	739	698	306	253	577	248	735	515	342	285	61
219	138	568	705	658	427	511	532	210	357	307	153
633	711	98	433	619	691	60	20	137	733	485	324
659	419	719	684	269	392	754	68	259	284	467	295
48	300	224	541	600	715	630	613	172	66	425	558
604	38	692	34	557	263	258	384	51	317	375	111
325	146	701	472	388	607	289	94	644	377	745	580
500	598	496	240	443	594	364	554	74	637	187	712
655	747	576	145	47	501	417	490	622	162	491	626
578	699	750	14	128	493	282	226	459	351	589	209
582	545	740	596	147	275	121	362	470	53	193	291
333	290	179	523	527	42	30	402	738	525	113	374
664	646	507	609	685	756	189	700	148	260	654	464

191	350	378	176	257	538	599	603	11	702	372	571
237	12	96	170	192	653	88	416	540	25	180	522
642	166	72	246	194	707	680	33	639	159	215	75
728	227	516	681	344	90	99	559	127	513	265	7
23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555
69	708	345	494	271	238	726	393	534	132	385	587
686	647	368	203	3	481	482	332	199	167	687	560
539	495	743	44	583	91	610	690	336	31	261	403
379	255	725	498	323	386	198	572	220	484	73	322
677	606	340	272	343	58	503	381	26	421	517	761
79	676	334	348	760	49	663	235	597	566	313	627
39	414	618	518	479	55	752	186	694	563	649	175
223	133	549	420	277	510	727	305	391	228	632	404
196	206	84	52	184	755	212	581	763	429	107	697
724	590	77	737	445	71	682	502	509	120	254	413
758	640	149	616	43	195	242	239	211	611	453	757
218	86	19	310	387	2	105	117	424	244	504	116
449	591	8	229	605	399	629	182	57	461	314	588
177	234	190	102	106	714	122	617	612	216	673	312
431	326	621	80	382	721	93	45	156	665	553	767
475	174	329	394	415	547	188	331	543	365	308	46
67	670	330	157	354	302	389	480	367	232	768	27
207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631
41	471	131	201	245	54	202	444	556	544	720	270
32	688	759	442	279	478	136	197	123	273	671	437
262	457	672	268	349	95	50	488	706	546	183	221
178	160	528	696	62	9	214	615	573	489	430	97
744	22	753	524	446	213	746	731	181	231	283	661
570	401	276	78	400	4	562	526	683	301	89	477

Clé de chiffrement de la deuxième version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

156	453	55	623	217	430	383	574	180	389	257	147
405	685	168	355	721	754	738	227	187	474	143	551
127	578	506	528	477	713	723	319	702	417	384	67
512	575	469	301	86	237	703	204	366	2	268	247
340	452	642	21	12	343	128	214	390	220	262	76
495	281	286	66	397	169	445	170	219	24	48	531
261	338	478	282	635	44	191	209	735	323	410	514
255	234	43	54	516	291	97	344	720	695	129	332
461	117	522	270	89	537	298	161	515	346	758	737
253	35	75	739	240	719	131	27	548	98	455	195
78	683	710	178	595	387	400	403	633	353	350	250
687	37	462	235	119	53	200	385	192	530	179	419
392	488	61	457	45	125	1	576	58	312	552	509
546	706	276	241	617	73	585	681	751	753	517	677
399	91	473	557	701	26	302	637	632	32	565	755
135	539	47	266	110	505	17	631	647	727	303	577
427	411	493	23	743	458	259	122	285	114	489	544
182	72	582	194	264	120	439	279	669	718	292	207
101	407	446	158	475	536	284	630	215	724	25	256
172	622	650	734	87	638	450	521	416	665	762	519
700	251	491	335	287	608	742	34	426	104	370	290
7	315	77	480	330	486	183	619	730	19	331	141
599	555	229	202	359	731	757	273	504	308	167	155
230	542	654	304	165	423	360	733	705	662	648	763
28	433	92	664	694	716	175	760	656	190	171	415
263	627	470	51	100	589	4	707	311	148	88	696
16	466	498	590	111	224	363	468	239	408	162	621
36	333	373	626	561	113	341	70	717	511	379	3
371	109	508	378	463	673	693	221	367	30	388	8
212	750	748	667	374	33	358	661	289	116	447	674
566	173	364	747	314	401	500	185	328	140	634	198
676	449	527	490	260	106	588	79	672	68	759	502
591	142	443	94	420	581	698	108	471	572	296	188
436	38	395	437	545	614	163	525	605	704	121	160
587	318	294	764	307	615	107	761	337	151	422	604
326	69	412	223	300	213	472	249	678	690	580	658
765	11	57	573	99	124	184	523	402	501	226	479
649	361	137	193	618	181	174	39	248	299	579	484
62	670	345	351	636	382	347	454	41	118	644	675
74	620	745	218	252	610	265	438	280	564	60	362
404	503	271	712	418	569	550	406	236	13	600	603

96	82	547	126	568	369	199	534	602	612	485	596
365	31	441	203	682	49	216	729	85	456	722	154
641	258	254	524	668	483	616	435	598	317	532	549
768	613	46	465	628	680	460	225	297	186	246	645
321	133	459	520	196	584	145	413	59	692	231	375
688	102	444	210	293	322	643	714	274	242	339	201
749	65	325	767	691	736	726	424	766	601	144	376
349	159	429	130	583	233	18	529	709	646	625	208
269	380	377	64	594	487	451	244	639	305	176	711
20	112	756	653	71	684	464	157	29	563	189	606
553	651	699	232	50	352	138	728	497	467	152	310
103	10	507	327	715	586	245	434	15	14	746	640
205	81	56	541	526	309	177	393	134	663	562	741
725	368	597	391	394	348	272	288	238	660	40	153
136	533	295	396	571	744	84	146	342	425	164	629
123	689	567	52	320	708	166	740	316	275	559	611
481	686	267	657	93	556	197	752	386	5	336	494
139	554	428	90	83	732	313	592	671	243	448	431
518	132	95	496	277	607	535	80	306	22	540	381
570	372	6	334	652	356	211	659	149	543	499	421
228	278	492	513	357	329	679	432	482	510	115	655
9	624	283	63	150	666	354	324	476	697	409	609
440	414	593	105	442	560	558	206	222	42	538	398

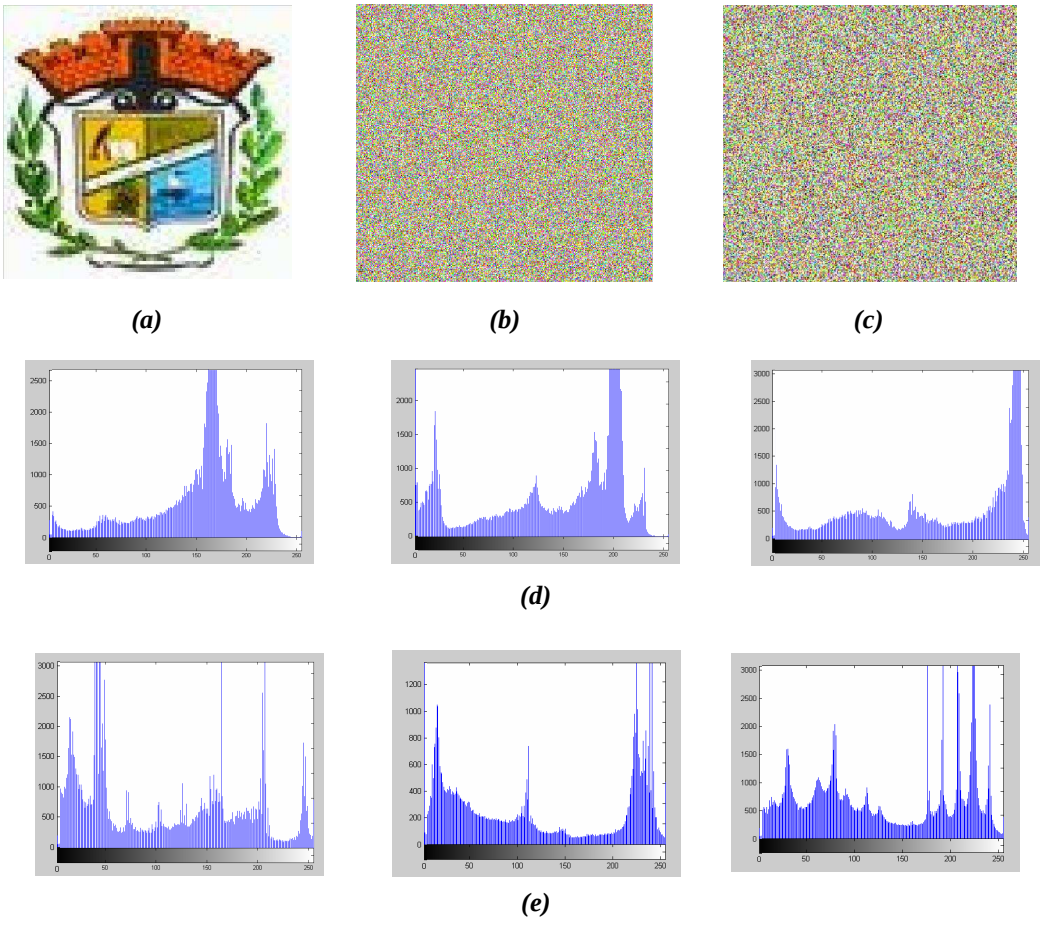


Figure V.6. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

65	9	12	62	15	21	94	7	28	11	81	3
61	36	78	70	91	25	56	97	85	4	30	24
83	32	57	40	54	31	63	8	95	26	75	55
19	96	20	38	49	92	47	44	1	98	52	89
35	67	6	72	69	53	74	60	14	43	84	23
22	82	88	73	90	76	39	13	59	33	58	68
41	50	86	46	5	51	27	99	79	10	71	66
87	48	16	2	45	64	42	80	17	34	77	29
37	93	128	625	414	134	459	456	642	446	447	205
635	230	755	387	503	179	646	371	252	744	662	574
575	475	768	734	494	638	389	441	312	571	445	214
367	308	498	643	659	705	175	676	237	530	674	370
527	626	415	330	333	472	600	238	339	137	730	257
262	144	660	139	217	323	739	329	578	420	618	735
463	199	729	207	340	336	409	551	295	296	477	222
468	276	151	258	725	178	764	245	132	366	525	305
196	717	363	484	658	136	342	397	408	198	598	443
589	455	256	239	287	216	181	290	338	111	334	751
345	392	560	697	355	585	418	18	610	162	534	154
467	241	250	694	464	461	614	500	743	170	244	741
404	259	211	645	318	650	593	442	187	282	639	288
648	267	655	220	573	436	552	344	164	225	727	165
226	667	693	416	116	265	536	532	686	522	471	587
316	260	486	514	488	268	601	628	767	365	382	314
528	195	493	430	108	227	212	145	194	753	716	675
547	152	517	374	182	539	356	677	724	760	377	434
546	119	702	204	590	481	394	114	509	395	301	419
473	490	656	728	168	583	538	426	745	518	149	141
131	279	644	140	580	701	325	348	341	285	203	298
651	263	723	235	462	691	622	722	129	437	354	163
335	398	647	122	469	603	553	737	269	511	101	240
110	361	400	402	317	186	733	277	766	142	264	172
399	613	703	588	231	315	106	669	531	380	233	180
319	504	275	113	425	153	372	748	424	281	157	234
249	581	466	200	289	533	411	630	579	223	221	376
679	569	596	188	421	758	519	146	427	243	123	331
671	508	678	206	304	364	273	440	561	324	297	713
670	274	118	576	715	611	752	620	379	489	369	750
343	278	103	479	621	485	224	526	453	229	604	352
765	653	413	714	499	169	595	762	746	465	410	403
537	100	609	327	117	683	617	570	740	492	373	417
310	505	155	543	246	390	689	102	747	710	594	616
497	381	406	756	563	548	487	632	757	720	512	636
749	474	631	294	433	476	347	661	197	718	393	709
577	174	712	665	592	328	599	435	311	375	719	385
280	480	299	261	292	423	242	159	283	696	542	761
641	349	582	458	565	171	432	431	731	520	732	597
541	605	666	209	368	202	612	337	680	156	272	652
619	633	657	135	125	150	148	506	255	495	104	313
535	540	401	521	412	320	554	634	572	684	654	332
523	627	449	690	183	460	251	682	602	664	742	685
470	309	161	350	515	266	452	640	501	444	567	606
502	429	322	483	711	454	428	726	357	544	378	177
184	291	407	763	624	591	201	218	510	391	556	124
478	293	566	557	623	629	189	721	321	219	270	232
208	302	708	608	586	362	213	388	672	707	248	568
307	271	405	130	115	663	185	253	143	681	698	545
176	422	754	695	450	396	668	109	439	210	607	105
692	673	759	637	112	516	699	121	584	286	736	358
107	147	326	353	562	228	127	513	215	687	555	491
700	649	529	190	738	191	507	138	615	704	383	524
496	193	359	133	384	564	306	550	438	173	236	386
549	167	346	120	448	457	126	300	558	192	254	451
559	482	706	303	284	160	247	688	166	351	158	360

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

498	116	41	239	486	387	497	89	154	262	303	25
129	485	272	506	489	496	446	502	195	127	234	350
240	282	491	338	501	304	256	484	29	209	382	212
238	512	511	494	335	166	495	79	437	381	299	499
480	508	505	481	358	73	482	483	492	392	427	359
507	269	510	236	487	493	243	372	167	231	182	37

94	106	490	87	333	500	71	64	16	504	488	160
503	438	97	227	1	317	336	331	324	76	150	478
509	90	203	716	401	386	194	442	477	328	529	4
762	474	132	3	709	407	735	376	86	596	2	48
651	228	704	200	345	380	201	626	18	5	23	24
385	660	531	732	145	248	562	677	373	663	725	648
736	712	453	84	365	52	569	128	170	183	526	754
43	294	746	146	224	83	112	745	434	448	764	582
400	541	436	635	339	597	390	632	717	26	738	551
383	721	237	93	156	144	463	623	375	627	727	130
152	173	728	731	281	722	220	68	707	656	153	204
193	680	252	54	95	550	429	465	759	70	542	34
55	528	149	414	763	590	232	729	218	646	639	600
458	49	472	57	678	630	664	374	593	533	750	585
190	344	384	117	631	676	539	205	208	300	577	657
697	196	696	394	141	689	131	559	445	513	99	69
669	706	693	578	32	142	756	247	169	768	100	742
217	290	59	264	393	755	330	701	703	332	743	726
266	444	221	549	199	163	659	370	270	250	621	702
268	614	740	552	21	96	435	766	245	622	326	28
278	417	316	752	15	343	733	634	430	618	311	181
340	189	325	532	557	624	765	636	747	9	470	571
140	595	341	538	178	710	308	402	296	573	63	271
125	610	553	449	215	699	20	411	519	263	692	357
690	162	464	439	471	126	666	105	242	720	147	681
604	420	12	523	714	615	739	11	35	155	184	136
192	426	118	318	546	705	368	580	65	253	283	186
682	460	53	670	719	399	724	440	749	364	450	222
45	202	561	121	389	19	109	164	47	261	223	246
77	558	408	82	476	179	423	535	56	616	8	139
38	760	80	406	521	405	527	298	441	249	454	396
723	447	708	748	91	348	684	120	180	715	314	210
617	655	751	658	683	312	391	168	607	61	598	33
404	424	673	570	601	418	295	642	346	613	694	643
159	661	111	257	50	327	287	674	589	289	39	62
219	285	443	143	30	640	274	462	757	362	124	310
397	753	60	259	214	254	216	81	92	412	548	574
555	293	276	534	695	667	671	605	473	302	665	176
433	594	255	265	591	587	516	456	267	206	518	455
369	319	306	98	431	603	292	395	291	378	602	103
191	638	107	688	40	174	581	356	85	102	469	679
711	347	46	213	409	119	138	566	547	371	342	377
211	628	641	165	554	609	425	78	421	413	108	575
525	612	568	10	280	114	349	520	536	468	517	652
451	649	88	197	198	416	226	110	134	157	185	172
592	556	337	415	66	524	734	75	241	279	113	698
744	654	315	422	351	27	515	320	687	611	686	700
297	572	653	650	361	543	629	761	14	475	619	175
586	51	579	288	355	540	466	637	244	730	606	564
353	36	691	235	230	563	379	398	44	522	567	576
7	403	718	388	363	685	565	758	599	329	133	273
137	459	229	122	625	584	17	275	322	251	187	367
410	334	645	560	158	123	633	583	305	101	151	741
428	537	457	452	313	354	207	737	277	366	675	233
258	309	148	620	479	647	58	161	352	767	323	22
42	432	467	713	588	225	668	177	321	419	104	6
360	188	644	301	135	74	72	461	672	31	307	286
171	67	260	284	530	545	608	544	115	13	662	514

Le tableau suivant (tableau V.2) représente une récapitulation des résultats de chiffrement des données tests présentées ci-dessus :

			Taille donnée (éléments)	Taille clé (bits)	Efficacité	Temps chiffrement (s)	Temps déchiffrement (s)
Données texte	Texte1	Version-Chiff1	1084	15323	15784.0	28.35	2,06
		Version-Chiff2			15030.0	28.02	2,13
	Texte2	Version-Chiff1	1151	15323	19498.0	29.16	2,67
		Version-Chiff2			18712.0	28.97.	2,65
Données images	Lena	Version-Chiff1	131 X 131	7680	139362.0	14.62	3.7
		Version-Chiff2			134074.0	14.21	4.01
	Logo	Version-Chiff1	420 X 395	7680	246426.0	15.06	3.11
		Version-Chiff2			225688.0	15.28	3.81

Tableau V.2. *Résultats obtenus par TabuCrypt.*

V.4. Discussion et évaluation des résultats

La première chose à remarquer d'après les résultats présentés dans la section précédente, est que le temps de chiffrement est proportionnellement dépendant de la taille de l'image à chiffrer. En effet, le temps de chiffrement de l'image Lena (34.826 secondes en moyenne) est plus petit par rapport au temps de chiffrement de l'image Logo (35.59 secondes en moyenne), du fait que la première est plus petite que la deuxième. La même remarque est valable pour le cas des deux données texte Texte1 et Texte2. Cette différence en temps revient au fait que le temps de chiffrement englobe le temps de lecture et de codage de la donnée à chiffrer en plus du temps de calcul, proprement dit, de la solution (temps de la recherche tabou). Le premier (temps de lecture et de codage) est strictement dépendant de la taille de la donnée à chiffrer, tandis que ce n'est pas le cas pour le deuxième (temps du calcul tabou). Donc, c'est surtout le temps de lecture et codage qui fait la différence en temps de chiffrement global d'une donnée par rapport à une autre.

De même, du coté du temps de déchiffrement c'est la phase de décodage de la donnée chiffrée qui fait la différence en temps de calcul.

Maintenant et en comparaison avec nos algorithmes développés et présentés précédemment à travers les deux précédents chapitres, le temps de calcul de ce dernier algorithme se trouve meilleur que le temps de calcul de certains de ces algorithmes et plus mauvais que le temps de calcul de certains autres algorithmes. La figure V.7 positionne le temps de calcul de l'algorithme *TabuCrypt* parmi nos quatre algorithmes PosESecL1, PosESecL2, OEEA et *AntCrypt*.

D'après le résumé de temps de calcul de nos cinq algorithmes développés par utilisation des trois métaheuristiques d'algorithmes évolutionnaires (PosESecL1, PosESecL2 et OEEA), de colonies de fourmis (*AntCrypt*) et de recherche taboue (*TabuCrypt*), nous constatons que *TabuCrypt* possède un temps de calcul meilleur que celui de PosESecL1 et de PosESecL2 mais plus grand que celui de OEEA et celui de *AntCrypt*.

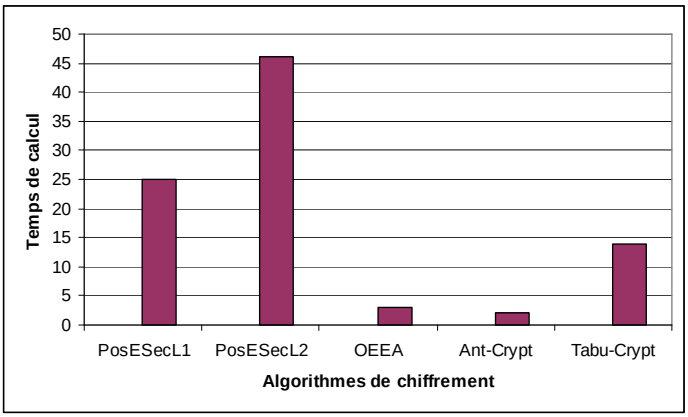


Figure V.7. *Comparaison des temps de calcul.*

Et comme *TabuCrypt* est un algorithme bâti par exploitation d'une métaheuristique, et nous l'avons expliqué précédemment, l'adaptabilité et l'efficacité d'application de telles méthodes dans un tel domaine à cause de leur large utilisation de l'aléatoire a été la source de notre motivation d'application de métaheuristique. L'algorithme développé sera, ainsi, doté d'un grand pouvoir confusionnel pour compliquer le plus possible la tâche de cryptanalyse.

En effet, l'aspect non déterministe innové à travers nos approches proposées, et par conséquent celle, ici, présentée, représente l'un des points forts de l'algorithme *TabuCrypt* tout comme nos autres algorithmes leur assurant, ainsi, une résistibilité contre les attaques différentielles. Ceci est assuré par une élection aléatoire, sur l'ensemble des générations, des voisins des éléments codant une solution intermédiaire pour calculer une autre solution intermédiaire ou une solution finale qui soit satisfaisante.
De même, l'attaque statistique sera presque impossible à appliquer grâce à ce dernier point en plus du fait que les données originales et leurs version chiffrées sur une instance, seront presque toutes différentes comme le montre le tableau suivant (tableau V.3) résumant les valeurs d'application des mesures de similarité NPCR, MAE et MSE.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.8927	0.97	0.68
Lena-version chiffrée 2	0.9104	1.05	0.71

Tableau V.3. *Niveaux de confusion de Tabu-Crypt.*

De son tour, l'attaque exhaustive sera mise à l'écart grâce à la taille de clé qui est largement sécurisée et qui est de taille égale à 15323 bits dans le cas de manipulation de données texte et qui égale à 7680 bits dans le cas de manipulation de données images.

Les attaques fréquentielle sont aussi pénalisées du fait, d'un coté, que ces dernières étudient les fréquences d'apparition des caractères dans un message (précisément un texte) écrit suivant une langue naturelle, alors que notre algorithme *TabuCrypt* peut traiter toutes sortes de message constitué de caractères du code Unicode. D'un autre coté, le message chiffré peut être constitué de caractères complètement ou partiellement différents de ceux constituant le message en clair.

V.5. Conclusion

La recherche Tabou est une méthode d'optimisation qui utilise la notion de voisinage d'une solution. Cette méthode permet, à partir d'une solution initiale, de visiter un ensemble de voisins. Ces voisins seront évalués et permettent l'évolution, à travers des règles spécifiques, vers une solution finale.

Ainsi et dans ce chapitre, nous avons présenté notre nouvel algorithme de chiffrement tabou, *TabuCrypt*, qui opère suivant le mode de chiffrement à base d'occurrence expliqué et validé dans le troisième chapitre.

Nous avons présenté en détails les différentes étapes du processus développé et nous l'avons testé sur différentes données test. Ainsi, notre objectif qui se résume en l'exploitation d'un voisinage en recherche locale de type taboue pour la manipulation de données texte et images, est atteint dans certains cotés vu les qualités de la méthode tabou (simplicité du principe, performance…) en donnant lieu à des résultats de bon niveau de confusion. Toutefois, le temps de convergence du processus de résolution est relativement lent par rapport à certains de nos précédents algorithmes proposés : OEEA et *AntCrypt*. Cela revient à l'aspect parallèle encapsulé à travers le principe soit des AEs ou des algorithmes de colonies de fourmis.

Conclusion et perspectives

Au cours de cette thèse nous avons proposé une nouvelle approche de cryptage de données texte et images par exploitation de métaheuristiques parmi lesquelles, nous nous sommes intéressés à celle inspirée du monde biologique et désignée par « approches biomimétiques ». Il s'agit des algorithmes évolutionnaires inspirés des principes de l'évolution naturelle des espèces. Par la suite, nous avons abordé une approche inspirée de modèles d'organisation naturels observés dans les sociétés animales, en particulier, dans une colonie de fourmis. Ces deux méthodes exploitées appartiennent à la catégorie des métaheuristiques à population. De même et pour l'autre catégorie qui est celle des métaheuristiques à trajectoire, nous sommes aussi arrivés à proposer un autre algorithme de cryptage utilisant le principe d'une recherche tabou.

Ainsi et après avoir étudié un panel assez diversifié des techniques de cryptage et avoir analysé les problèmes de sécurisation du transfert, nous avons élaboré nos propositions de stratégies de résolution d'un tel problème. Autrement dit et afin de détailler nos apports, notre thèse a été organisée autour de deux parties dont l'objectif de la première est de situer le lecteur dans le contexte de notre travail pour lui permettre de suivre les démarches réalisées dans cette étude. Dans la deuxième, nous avons détaillé la conception, la réalisation et l'évaluation de l'approche de résolution proposée.

Notre première contribution a porté sur l'élaboration et la conception de nouveaux algorithmes de chiffrement efficaces et robustes qui traitent le problème de taille de clés leur permettant, ainsi, d'être loin des attaques exhaustives.

Pour ce faire, nous avons commencé par une étude approfondie du problème tout en essayant de le ramener en un problème d'optimisation par une formulation adéquate aux métaheuristiques à appliquer : métaheuristique des algorithmes évolutionnaires, métaheuristique des algorithmes de colonies de fourmis et celle de recherche tabou. Pour la première, les obstacles à franchir étaient surtout la définition d'un codage adéquat de solutions. Dans ce sens, deux modes de codage différents ont été proposés : codage à base de positions et un deuxième à base d'occurrences donnant naissance, ainsi, à trois algorithmes de chiffrement, PosESecL1, PosESecL2 et OEEA de qualités différentes dont celui opérant suivant le mode de chiffrement par occurrences (OEEA) était le plus efficace et le plus robuste.

L'application de la deuxième métaheuristique a démontré l'efficacité du codage à base d'occurrences. À ce stade aussi, il a fallu définir avec précaution certains paramètres tels que la fonction d'évaluation, le mécanisme de manipulation de phéromone (incrémentation et évaporation) ainsi que celui de génération de la clé de chiffrement qui doit doter l'algorithme développé, *AntCrypt*, d'une résistance contre les attaques exhaustives vu que cet algorithme est de type symétrique.

De même, l'exploitation de la troisième métaheuristique a, aussi, abouti au développement d'un cinquième algorithme de chiffrement, *TabuCrypt*. Cela a nécessité la projection des composants d'une telle métaheuristique sur le problème de cryptage, à savoir la

liste tabou, stratégie de choix de voisins, table de hachage et les mécanismes avancés en recherche tabou (Intensification, diversification et aspiration). Les résultats étaient satisfaisants que ce soit sur le plan cryptographique que cryptanalytique, toutefois, il nécessite plus de temps de calcul que certains de nos autres algorithmes (*OEEA* et *AntCrypt*).

Notre deuxième contribution était l'innovation d'algorithmes de chiffrement non déterministes en se bénéficiant de l'aspect aléatoire de la sélection naturelle ou du déplacement des fourmis. En effet, l'aléatoire représente l'ennemi des cryptanalystes. En appliquant toutes les attaques redoutables, il est donc quasiment impossible de parvenir à l'information initiale ou au moins d'établir une relation entre les données chiffrées puisqu'on obtient toujours des versions chiffrées différentes à chaque application de l'algorithme même lors d'un chiffrement sur plusieurs instances d'une même donnée originale.

En fait, à travers notre étude nous sommes arrivés à présenter les métaheuristiques comme des méthodes de résolution approchées se voulant simples et adaptables à tout type de problèmes. Leur capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la solution trouvée. Du point de vue de la recherche opérationnelle, cet inconvénient n'est pas toujours un problème, tout spécialement quand une seule approximation de la solution optimale est recherchée. En nous situant dans ce cas, nous aurions pu mettre en évidence l'avantage de ces nouvelles approches pour résoudre le problème de cryptage de données texte ou images.

Toutefois et malgré que nos algorithmes développés soient de bonne qualité cryptographique, plusieurs autres idées et applications en relation avec ce présent travail pour le compléter, restent à concrétiser. En premier lieu, il est urgent d'aborder le problème de métaréglage [Dréo, 2004], non seulement pour régler automatiquement les paramètres des métaheuristiques, mais aussi pour maîtriser leurs fonctionnements.

De même et malgré que les algorithmes évolutionnaires ou même les algorithmes de colonies de fourmis permettent d'obtenir des solutions pas toujours optimales, mais possiblement satisfaisantes à de nombreux problèmes complexes, leur puissance d'optimisation est liée à la puissance sous-jacente des machines les exécutant.

Comme ce sont des algorithmes inspirés de la nature, ils sont fortement parallélisables et il se trouve que le récent développement des cartes graphiques, dédiées au calcul de rendu 3D dans un premier temps, puis généralisées au milieu des années 2000 avec les *General Purpose Graphic Processing Units (GPGPU)*, permet d'utiliser le grand nombre de cœurs de calcul qu'elles comportent pour autre chose que du rendu graphique.

Dans [Mait, 2011], l'auteur détaille et étudie comment les algorithmes évolutionnaires peuvent bénéficier de ces nouvelles architectures, avec leurs particularités. Plusieurs algorithmes ont été présentés, qui permettent l'utilisation efficace de ces processeurs pour l'évolution artificielle, que ce soit pour les algorithmes génétiques et stratégies d'évolution, mais aussi pour la programmation génétique. Différentes stratégies de parallélisation ont été utilisées, dont l'une nécessitant le développement d'un opérateur parallèle de réduction de la population. L'adaptation des paramètres de ces algorithmes pour permettre le portage de ces derniers sur GPGPU est détaillée, ainsi que certains points impactant directement les performances de ces portages.

Ainsi, de tels schémas peuvent être envisagés pour une parallélisation de nos algorithmes développés surtout PosESecL2 possédant un bon niveau confusionnel mais souffrant d'une lenteur de calcul vu la grande taille de chromosomes manipulés codant des données à chiffrer de grandes tailles.

D’un autre coté et vu que nos algorithmes de cryptage développés sont de type symétrique, une méthode de communication sécurisée de clé de chiffrement secrète peut être envisagée lors de la manipulation de données images. Il s’agit de marquer l'image cryptée dans des régions spécifiques, ou des régions déterminées pour l'utilisateur. Nous pouvons également utiliser des méthodes de marquage (watermarking) sans perte pour insérer la clef cryptée dans l'image cryptée. Il serait souhaitable de développer des évaluations sur la robustesse concernant les attaques. Ainsi, nous souhaitons faire des études concernant la cryptanalyse de nos méthodes pour classifier son niveau de sécurité face aux attaquants.

Par ailleurs, nos perspectives de recherche à long terme s’orientent vers l’étude et le développement d’autres méthodologies de cryptage de données. Nous nous intéresserons, en particulier à :

- La métaheuristique des automates cellulaires inventée par Ulam [Ulam, 1950] et Von Neumann [VonN, 1966] et basée sur le concept de la vie artificielle. Ce concept s’appuie sur des règles extrêmement simples permettant de construire des structures très complexes et esthétiques, d’où sa ressemblance avec le principe d’auto-organisation en biologie animale, notamment avec les mécanismes de la colonie de fourmis. Ces règles peuvent stipuler, par exemple, qu'une naissance nécessite un certain rassemblement de population, que les cellules ne peuvent survivre à un trop grand isolement et qu'une trop forte concentration les étouffe. Donc, nous comptons adapter ces règles au problème de cryptage où de génération en génération, l’application de ces règles permettra l’émergence d’une solution au problème.

- L’approche des « hyper-heuristiques » introduite dans [Cowl, 2000], [Cowl, 2002], [Burk, 2003], [Burk, 2005] et qui consiste à travailler dans un espace de recherche composé de métaheuristiques afin d’optimiser le choix de la métaheuristique à utiliser pour un problème donné. Il s’agit, donc, de mettre en confrontation un ensemble de métaheuristiques différentes pour en sélectionner la plus optimale.

Ministère de l'Enseignement Supérieur et de la recherche Scientifique

BADJI MOKHTAR-ANNABA UNIVERSITY
NIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار – عنابة

Faculté des sciences de l'ingénieur
Département d'informatique

THESE

Présentée en vue d'obtention du diplôme de ***DOCTORAT en Informatique***

Sécurisation évolutionnaire du transfert d'images

Option
Informatique

Par
SOUICI Ismahane

Directeur de Thèse
SERIDI Hamid Pr. Univ. 08 mai 1945 Guelma

Devant le jury

Président :	**KHADIR Tarek**	Pr. Univ. Badji Mokhtar Annaba
Examinateurs :		
	FARAH Nadir	Pr. Univ. Badji Mokhtar Annaba
	MEROUANI Hayet	MC. Univ. Badji Mokhtar Annaba
	KHOLADI Khiredine	Pr. Univ. Mentouri Constantine
	CHIKHI Salim	Pr. Univ. Mentouri Constantine

Année universitaire : 2012/2013

Remerciements

En premier lieu, je tiens à exprimer ma profonde gratitude à Monsieur SERIDI Hamid, Professeur à l'université 08 mai 1945 de Guelma, pour avoir dirigé ma recherche depuis le Magister et pour la confiance et l'intérêt qu'il m'a témoignés tout au long de l'élaboration de cette thèse. Je lui suis très reconnaissante pour le suivi régulier et formateur reçu durant ces quatre années de Doctorat, et bien avant, durant les trois années de Magister. Je le remercie pour la démarche scientifique rigoureuse et l'esprit d'auto-critique qu'il a su m'inculquer. J'espère avoir été à la hauteur de ces espérances.

J'adresse mes remerciements les plus sincères à Monsieur KHADIR Tarek, Professeur à l'université Badji Mokhtar d'Annaba, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Mes vifs remerciements vont aux membres du jury qui ont accepté de prendre de leur temps pour examiner mon travail : Monsieur FARAH Nadir, Professeur à l'université Badji Mokhtar d'Annaba, Madame MEROUANI Hayet, Maître de Conférences à l'université Badji Mokhtar d'Annaba, Monsieur KHOLLADI Khiredine, Professeur à l'université Mentouri de Constantine et Monsieur CHIKHI Salim, Professeur à l'Université Mentouri de Constantine.

Au cours de ces quatre années de recherche, j'ai eu à effectuer un stage de courte de durée en 2010, au laboratoire CReSTIC de l'université de Reims Champagne Ardenne, France, sous la direction de Monsieur AKDAG Herman, Professeur à l'université de Reims Champagne Ardenne, France. Qu'il trouve ici, l'expression de mes remerciements les plus sincères, pour son chaleureux accueil. Je tiens à remercier, aussi, Monsieur Cyril DE RUNZ, Maitre de Conférences à l'université de Reims Champagne Ardenne, France pour sa disponibilité, ainsi que pour ses discussions fructueuses que j'ai eues avec lui.

Merci à celles et ceux qui auront à lire tout ou une partie de ce manuscrit et qui y trouveront un intérêt quelconque.

Enfin, un grand MERCI à ma famille et, en particulier, mes parents pour le soutien et les encouragements qu'ils ont su m'apporter pour arriver au terme de cette thèse.
J'espère les honorer avec ce travail.

الملخص

في يومنا هذا أصبحت شبكات المعلوماتية معقدة و التنصتات غير القانونية عليها قائمة. لهذا أصبح من الضروري حماية المعلومات الحساسة المنقولة عبر هذه الشبكات, أين يعد علم التشفير واحد من الحلول الجد فعالة, الذي و على الرغم من كل التطورات التي سجلت بهذا المجال ماتزال بعض النقائص مسجلة خاصة ما يتعلق بحجم المفتاح المستعمل و مقاومة الهجمات المتطورة.

في هذا السياق ينصب موضوع بحثنا للدكتوراه حيث نسعى إلى تطوير خوارزمات تشفير باستعمال métaheuristiques التي تعد من الطرق التي توفر حلولا تقريبية و تفتح مجالات تصميمية لطرق حل مهمة لمشاكل التحسين.
هذه الأخيرة تنقسم إلى قسمين : metaheuristiques التي تستعمل مجموعة حلول métaheuristiques التي تستعمل حلا واحدا. وبما أن métaheuristiques توظف طابع العشوائية في معظم مراحل عملها و الذي يعد عنصرا مهم إستعماله في مجال التشفير لتعقيد مهمة المهاجمين, لهذا وقع إختيارنا على هذه الطرق لحل مشكل تشفير المعلومات النصية و الصور.

إذن وضمن الصنف الأول الذي يشمل métaheuristiques التي تستعمل مجموعة حلول قمنا بتطوير ثلاث خوارزمات تعتمد الطريقة التطورية المستوحاة من نظرية "دارون" و هذا حسب نوعين مختلفين من التشفير (التشفير المعتمد على الوضعيات و التشفير المعتمد على الظهور), بالإضافة إلى خوارزم رابع و الذي يستعمل métaheuristique مستوحاة من comportement الحقيقي للنمل.
بالنسبة للصنف الثاني والذي يضم métaheuristique التي تستعمل حلا واحدا, توصلنا إلى تكييف طريقة البحث الطابوه إستعمالها في حل المشكل لمدروس و هكذا قمنا بتطويرخوارزم تشفير طابوه خامس.

أخيرا قمنا بتجريب ومقارنة كل الخوارزمات المقترحة فيما بينها وبالنسبة لعدد من مراجع التشفير حيث أظهروا خصائص جيدة سواءا فيما يتعلق بدرجة الغموض أو مقاومة الهجمات الأكثر تطورا. قطة أخرى جد مهمة توصلنا إليها عبر عملنا هذا هي استحداث خوارزمات non déterministes , نقطة تساهم بدرجة كبيرة في زيادة قوة مثل هذه الخوارزمات.

Résumé

Aujourd'hui, les réseaux informatiques sont complexes et les écoutes illégales possibles. Il se pose donc un réel problème quant à la sécurité lors de la transmission de données. Pour des raisons éthiques, le transfert des données délicates ne peut se faire avec un tel risque et doit donc se protéger. La protection la plus adaptée pour ce type de communication réside dans la cryptographie. Cependant et malgré toutes ses évolutions et ses mises en œuvre, elle est toujours entravée par quelques défauts citant en particulier la taille des clés et la résistance contre les attaques avancées.

Ainsi, notre travail de thèse porte sur le développement de nouveaux algorithmes cryptographiques par exploitation des métaheuristiques constituant une partie importante des méthodes approchées et ouvrant des voies très intéressante en matière de conception de méthodes heuristiques pour l'optimisation. Ces dernières se scindent en deux catégories : les métaheuristiques à population de solutions et les métaheuristiques à une seule solution ou encore dites à trajectoire. Et vu que les métaheuristiques exploitent un aspect pseudo-parallèle durant différentes étapes de leurs schémas opératoires, chose qui est très intéressante à exploiter dans le domaine cryptographique pour compliquer de façon considérable la tâche des cryptanalystes, donc notre choix s'est porté sur l'utilisation de telles méthodes pour résoudre le problème de cryptage de données texte et images.

En effet, et pour la première catégorie de métaheuristiques à population, nous avons développé trois algorithmes de chiffrement exploitant les principes évolutionnaires inspirés de la théorie darwinienne, mais opérants suivant deux modes de chiffrement différents (chiffrement à base de positions et un chiffrement à base d'occurrences) ; et un quatrième algorithme utilisant une métaheuristique inspirée du comportement réel des fourmis (ACO).

Pour la deuxième catégorie qui est celle de métaheuristiques à trajectoire, nous avant arrivé à adapter la méthode de recherche taboue et l'exploiter pour résoudre notre problème étudié et ainsi développer un cinquième algorithme de cryptage tabou.

Les algorithmes proposés ont été évalués et comparés entre eux et avec les principaux algorithmes standards de cryptage où ils ont montré de bonnes caractéristiques relatives soit à leur degré confusionnel ou à leur résistibilité aux attaques les plus avancées. L'autre point crucial résultant de nos travaux de thèse était l'innovation d'algorithmes de chiffrement non déterministes, chose qui augmente considérablement leur robustesse.

Mots clés : Cryptage, attaques avancées, métaheuristiques, algorithmes évolutionnaires, recherche tabou, algorithme de colonies de fourmis.

Abstract

Today, computer networks are complex and the illegal wiretapping is possible. This raises a real problem of security when transmitting data. For ethical reasons, the transfer of delicate data can not be done with such a risk and must protect themselves. Protection best suited for this type of communication lies in cryptography. However, despite all its developments and implementations, it is still hampered by some shortcomings citing in particular the key size and resistance against advanced attacks.

Thus, the object of our thesis is the development of new cryptographic algorithms by exploitation of metaheuristics constituting a significant portion of the approximate methods and opening very interesting channels in the design of heuristic methods for optimization. The latter fall into two categories: metaheuristics of solutions population and metaheuristics of one solution or also tell trajectory metaheuristics. And since metaheuristics exploit a pseudo-parallel aspect during different stages of their operating patterns, something that is very interesting to use in the cryptographic domain to significantly complicate the task of the cryptanalysts, so our choice fell on the use of such methods to solve the problem of the encryption of text and images data.

Indeed, for the first class of metaheuristics of population, we developed three encryption algorithms exploiting the evolutionary principles inspired by the Darwinian theory but operating in two different encryption modes (positions based encryption and occurrences based encryption) and a fourth algorithm using a metaheuristic inspired by the ants real behavior (ACO).

For the second category which is trajectory metaheuristics, we arrived to adapt the method of tabu search and use it to solve the problem at hand. Thus, a fifth taboo encryption algorithm was developed.

The proposed algorithms were evaluated and compared among themselves and with principal encryption standard algorithms where they showed good characteristics for either their confusional degree or their resistibility against the most advanced attacks. The other crucial point arising from our thesis work was the innovation of non-deterministic encryption algorithms, something that greatly increases their robustness.

Table des matières

Chapitre III : CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

Chapitre IV : CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

Chapitre V : CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Dès que les hommes ont appris à communiquer, ils ont trouvé des moyens d'assurer la confidentialité d'une partie de leurs communications : l'origine de la cryptographie remonte sans doute aux origines de l'homme. En effet, le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire de les rendre inintelligibles sans une action spécifique.

Du bâton nommé « scytale » au Vie siècle avant JC, en passant par le carré de Polybe ou encore le code de César, on assista au développement plus ou moins ingénieux de techniques de chiffrement expérimentales dont la sécurité reposait essentiellement dans la confiance que leur accordaient leurs utilisateurs. Après la Première Guerre mondiale a lieu une première révolution technologique.

Mais ce n'est qu'à l'avènement de l'informatique et d'Internet que la cryptographie prend tout son sens. Les efforts conjoints d'IBM et de la NSA conduisent à l'élaboration du DES (Data Encryption Standard), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XXe siècle. À l'ère d'Internet, le nombre d'applications civiles de chiffrement (banques, télécommunications, cartes bleues…) explose. Le besoin d'apporter une sécurité accrue dans les transactions électroniques fait naître les notions de signature et authentification électroniques. La première technique de chiffrement à clef publique sûre (intimement liée à ces notions) apparaît : le RSA.

Donc, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information. C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à ramener et à modéliser le problème de cryptage comme un problème d'optimisation.

En effet, l'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solutions algorithmiques efficaces valables pour toutes les données.

Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles que les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens.

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques*. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, les chromosomes et les mécanismes d'intensification et de diversification pour la métaheuristique des algorithmes évolutionnaires), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Les métaheuristiques sont représentées essentiellement par les *méthodes de voisinage* comme le recuit simulé et la recherche tabou, et les *algorithmes évolutifs* comme les algorithmes génétiques et les stratégies d'évolution. Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi et hormis cette introduction et la conclusion générale qui reprennent les travaux de l'ensemble des chapitres et quelques perspectives majeures pour la poursuite de ce travail, le manuscrit est divisé en deux grandes parties.

Un état de l'art aussi complet que possible relatif soit au problème étudié qui est celui de cryptage, soit aux approches de résolutions exploitées qui sont les métaheuristiques, fera l'objet des deux premiers chapitres formant la première partie. En effet, le premier chapitre traite la cryptographie depuis sa première apparition jusqu'à nos jours en présentant un bref aperçu historique permettant de comprendre la distinction entre les trois disciplines (cryptologie, cryptographie et cryptanalyse), les fonctionnalités de base de la cryptographie (confidentialité, authentification, intégrité et la non-répudiation) et leurs différentes catégories (la cryptographie symétrique, asymétrique, hybride et quantique) tout en citant les fameux algorithmes appartenant aux principales catégories.
De son tour, le deuxième chapitre présente une introduction aux métaheuristiques tout en citant les concepts de base ainsi qu'une classification des métaheuristiques illustrée d'exemples d'algorithmes de chacune des classes.

Dans la deuxième partie de notre recherche, nous proposons un nouvel axe de recherche que représente l'application des métaheuristiques pour résoudre le problème de cryptage de données textes et images. Ainsi, les deux catégories de métaheuristiques ont été exploitées : métaheuristiques à population et les métaheuristiques à trajectoire.
Dans le cadre de la première, nous avons choisi d'exploiter deux métaheuristiques qui sont les algorithmes évolutionnaires (AEs) et les algorithmes de colonies de fourmis. Le principal avantage de ces méthodes heuristiques vient de leur capacité à traiter le problème de cryptage en ne possédant qu'un minimum d'informations sur celui-ci. Chose qui est primordiale dans ce problème pour augmenter la confusion des algorithmes développés.

Ainsi, le troisième chapitre résume l'application d'AEs où les différentes étapes partant du codage sont explicitées. En effet, pour un problème à résoudre, il est nécessaire d'adapter la représentation des individus aux objectifs recherchés. Cette adaptation permet de faire converger l'AE plus ou moins rapidement et de tenir compte naturellement des contraintes de la modélisation du problème. Dans le cas présent, l'espace de recherche que nous cherchons à optimiser est l'ensemble représenté par les solutions possibles du problème de cryptage et par une fonction d'évaluation de chaque solution.

Le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, les algorithmes ont été groupés en deux catégories distinctes suivant les modes de chiffrement utilisé implémentant différents codages : chiffrement à base de position ou chiffrement à base d'occurrences.

D'une manière globale, nous allons définir un individu de la population comme une donnée chiffrée possible. Cet individu doit pouvoir être évalué numériquement par la fonction d'évaluation proposée. Cette fonction de fitness calcule la qualité d'une donnée chiffrée en fonction du besoin de confusion. Nous montrerons aussi l'application de notre méthode sur quelques exemples d'images de différentes tailles. Une interprétation et une discussion des résultats obtenus seront abordées pour pouvoir, par la suite, comparer les nouveaux algorithmes proposés de cryptage évolutionnaire où l'algorithme opérant suivant le mode de chiffrement par occurrences a montré une efficacité en termes de temps de calcul, de pouvoir de confusion, de résistibilité aux attaques et de longueur de clé meilleure que celles des algorithmes opérants suivant le mode de chiffrement par positions.

Cette conclusion a été démontrée par l'application d'une deuxième méthode de résolution exploitant les algorithmes de colonies de fourmis illustrée sur un quatrième chapitre où les résultats obtenus été très satisfaisants.

Au niveau du cinquième chapitre, un autre algorithme de cryptage a été proposé s'inscrivant cette fois-ci, sous la deuxième catégorie de métaheuristiques dans le but d'explorer l'espace de recherche par exploitation de la notion de voisinage. Il s'agit d'un algorithme utilisant le principe d'une recherche tabou pour pouvoir choisir la configuration représentant la meilleure solution au problème de cryptage de données texte ou images.

Dans chacun des chapitres décrivant nos algorithmes de cryptage proposés, nous présentons des exemples et résultats expérimentaux sur des données tests. Nous avons aussi effectué des bilans en faisant ressortir les avantages et faiblesses des méthodes développées.

CHAPITRE I

CRYPTOGRAPHIE

I.1. Introduction

Depuis les temps historiques les plus reculés, l'homme a perçu le besoin de cacher, de dissimuler ou faire mystère des informations personnelles ou confidentielles, et cela bien avant l'ère informatique afin de les rendre inintelligibles à des lecteurs indésirables. C'est pourquoi, les codes ont existé.

Les origines de la cryptographie semblent remonter à plus de 4000 ans en Egypte. Plusieurs indications archéologiques tendent à montrer que les « écritures secrètes » sont en fait anciennes que l'invention de l'écriture elle-même. Polybius développa un système de codage des lettres de l'alphabet consistant à remplacer chaque lettre de l'alphabet par deux nombres, donnant la ligne et la colonne où se trouve cette lettre dans une matrice. Jules César utilisait une simple méthode de substitution de lettres pour communiquer secrètement avec ses généraux : c'est un chiffre par décalage,…etc.

C'est cependant au cours de la seconde guerre mondiale que la cryptographie s'inscrit véritablement comme élément central des stratégies militaires. Le cas le plus connu est certainement l'histoire entourant le décodage du code Enigma par les Polonais et les Britanniques. Une conjonction d'espionnage classique et d'efforts de mathématiciens polonais permet de déduire la clé utilisée et ainsi de décoder les messages encodés avec Enigma. On trouve apparemment bien moins de détails sur les efforts cryptographiques durant la guerre froide, probablement parce que ces informations sont encore « Top Secret ».

On en est maintenant à l'époque moderne où le champ d'application de la cryptologie s'est élargi et a trouvé un regain d'actualité avec toutes les applications nouvelles suscitées par l'utilisation de l'Internet. La révolution d'Internet et l'utilisation de plus en plus d'informations massives sous forme numérique facilitent les communications et rendent de ce fait plus fragiles les informations que l'on détient, c'est pourquoi il devient nécessaire de protéger le contenu de certains messages des inévitables curieux. En effet, les réseaux ouverts créent des brèches de sécurité et il est plus aisé à un adversaire d'accéder aux informations. Dans une communication à distance, des questions cruciales se posent et leurs réponses s'imposent ; comment être sur :

- que l'on parle à la bonne personne (authenticité),
- que nos propos ne sont pas altérés (intégrité),
- que la conversation n'est pas espionnée (confidentialité),
- de l'identité de l'émetteur (non répudiation) ?

Indéniablement, avec l'essor fulgurant des nouvelles technologies, le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements.

Alors que la cryptographie consiste à sécuriser les données, tandis que, la cryptanalyse est l'étude des informations cryptées afin d'en découvrir le secret. Grâce à la cryptanalyse, les militaires ont pu mener leurs guerres en découvrant les correspondances de leurs ennemis et en contrôlant les réseaux de communications mais d'autre part si elle est utilisée par une personne mal intentionnée il peut générer des dégâts onéreux aux entreprises ou aux sociétés.

I.2. Les Fondements de la Cryptographie

I.2.1. Terminologie

Comme toute science, la cryptographie possède son propre langage. Dans ce qui suit les mots clés de domaine cryptographique.

I.2.1.1. Cryptographie

Le terme cryptographie vient en effet des deux mots grecs : *Kruptus* qu'on peut traduire comme secret et *Graphein* pour écriture. Ainsi la cryptographie est l'art de dissimuler une information écrite en clair (plain text) en *cryptogramme* (cipher text) pour qu'elle soit incompréhensible que par son destinataire légitime par le biais d'une clé appelé « clé de chiffrement » (processus de *chiffrement*). Pour rendre l'information à nouveau intelligible par le biais d'une clé appelé « clé de déchiffrement » le processus inverse est appliqué (processus de *déchiffrement*).

On distingue généralement deux types de clefs :

- **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

Le schéma suivant illustre le processus cryptographique :

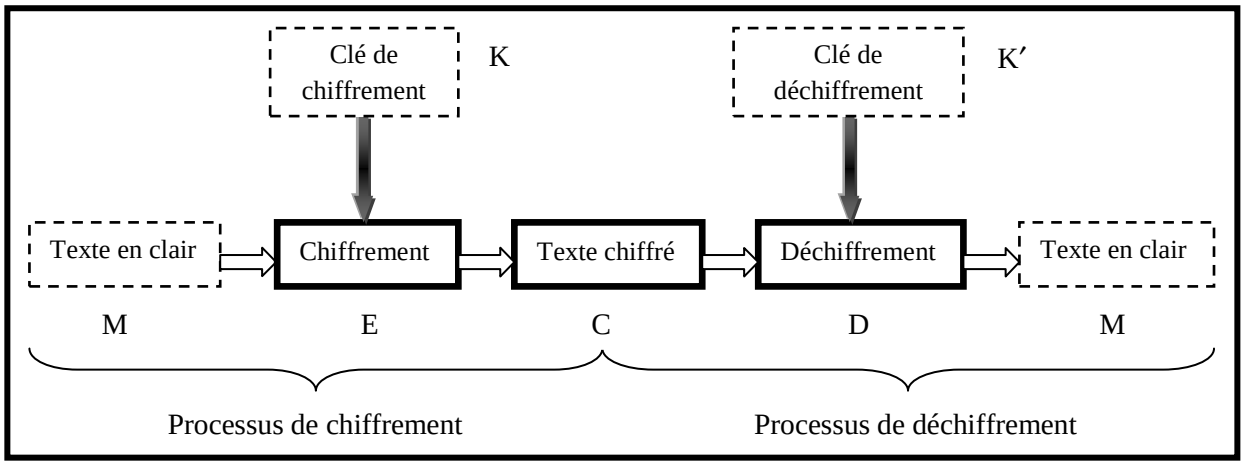


Figure I.1. *Processus cryptographique.*

I.2.1.2. Cryptanalyse

a. Définition

C'est l'art d'étude des crypto systèmes en cherchant leurs failles et leurs vulnérabilités afin de retrouver des messages clairs correspondant à des messages chiffrés sans avoir à connaître les clés utilisées dans le chiffrement. Lorsque tous les éléments de la méthode utilisée pour coder des messages sont repérés, on dit qu'on a **cassé** ou **brisé** le système cryptographique utilisé. Plus un système est difficile à briser, plus il est sûr.
La personne qui pratique la cryptanalyse est appelée : cryptanalyste. Il tente à décrypter le message chiffré pour découvrir son secret. On a distingué entre le verbe « décrypter » et « déchiffrer » puisque ce dernier est réservé pour le déchiffrement par le destinataire légitime.

b. Types de cryptanalyse

On distingue plusieurs types d'attaques :

- ***Attaque sur texte chiffré seul (ciphertext-only)*** : l'attaquant a seulement la possibilité d'intercepter un ou plusieurs messages chiffrés. La cryptanalyse est plus ardue de par le manque d'informations à disposition.
- ***Attaque à texte clair connu (known-plaintext attack*)** : se base sur la connaissance d'une partie du texte en clair pour déduire le reste du message. La tâche est de retrouver la clef utilisée pour chiffrer ce message.
- ***Attaque à texte clair choisi (chosen-plaintext attack)*** : se base sur la possibilité de choisir un texte clair et d'obtenir son chiffrement et en ayant la possibilité de générer les versions chiffrées de messages clair avec un algorithme considérer comme une **boîte noire** tel que les algorithmes à clé publique puisque l'algorithme est public.
- ***Attaque à texte chiffré choisi (chosen-ciphertext attack)*** : le cryptanalyste possède des messages chiffrés et essaye de les déchiffrer de son choix. Sa tâche est de retrouver la clef.

c. Familles d'attaques cryptanalytiques

Il existe plusieurs familles d'attaques cryptanalytiques, les plus connues sont les suivantes :

- **L'analyse fréquentielle** : examine les répétitions des lettres du message chiffré afin de trouver la clé. Cette technique est découverte par Al-Kindi au IXe siècle [www1] contre les chiffrements mono-alphabétiques. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est basée sur le fait que, dans chaque langue, certaines lettres ou combinaisons de lettres apparaissent avec une certaine fréquence.
- **L'attaque par dictionnaire** : le mot testé est pris dans une liste prédéfinis contenant les mots de passe les plus courants et aussi des variantes de ceux-ci. Ces listes sont généralement dans toutes les langues les plus utilisées, elles contiennent des mots existants ou des mots diminutifs (par exemple « powa » pour « power » ou « K7 » pour « cassette »). Elle est souvent couplée à l'attaque par force brute.
- **L'attaque par force brute** : s'appuie sur le cassage d'un mot de passe en testant tous les mots de passe possibles. C'est le seul moyen de récupérer la clé dans les algorithmes les plus modernes et encore inviolés comme AES.
- **La cryptanalyse linéaire** : c'est une attaque à texte clair inventée par le japonais Mitsuru Matsui [www1]. L'idée est de trouver des approximations linéaires entre les bits de sortie, les bits d'entrée et les bits de la clé. Si certaines de ces approximations apparaissent avec une probabilité suffisante, on a alors démontré que la correspondance

entre entrée et sortie n'est pas purement aléatoire. Le nombre de clés à envisager pour déchiffrer le message est restreint.

- **La cryptanalyse différentielle** : découverte par deux cryptologies israéliennes : Bihan et Shamir [www1]. Elle consiste à comparer les sorties de l'algorithme quand on lui met en entrée deux messages ayant une différence fixe. On étudie comme variant les sorties si les deux messages ne diffèrent que par un seul bit. Si en déplaçant ce bit à l'intérieur des messages, certains bits des sorties restent inchangés, on a alors trouvé une faille dans l'algorithme et celui-ci est attaquable.

I.2.1.3. Cryptologie

La cryptologie du grec kruptos « secret, caché » et logos « discours » qui embrasse à la fois la cryptographie et la cryptanalyse. Elle se partage entre la cryptographie, qui inclut la conception des mécanismes destinés à assurer les fonctions suivantes : confidentialité, intégrité, authentification et traçabilité (non répudiation), et la cryptanalyse dont le but est de déjouer les protections ainsi mises en place.

I.2.1.4. Fonction de hachage

Une fonction de hachage est une fonction mathématique qui à partir d'un message (d'une donnée) génère une autre chaîne, l'empreinte (généralement plus courte). Cette empreinte est très sensible au texte initial (une petite modification du texte provoque une grande modification d'empreinte).

I.2.1.5. Signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message. Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur.

I.2.1.6. Les certificats

Pour assurer l'intégrité des clés publiques, ces dernières sont publiées avec un certificat. Un certificat (ou certificat de clés publiques) est une structure de données qui est uniquement signée par une autorité certifiée.

CA, Certification Autorité, est une autorité en qui les utilisateurs peuvent faire confiance. Il contient une série de valeurs, comme le nom du certificat et son utilisation, des informations identifiants le propriétaire et la clé publique, la clé publique elle-même, la date d'expiration et le nom de l'organisme de certificats. Le CA utilise sa clé privée pour signer le certificat et assure ainsi une sécurité supplémentaire. Si le récepteur connaît la clé publique du CA, il peut vérifier que le certificat provient vraiment de l'autorité concernée et, ainsi, de s'assurer que le certificat contient des informations viables et une clé publique valide.

I.2.2. Fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur fiabilité et confidentialité au travers d'un canal peu sûr (téléphone, réseau informatique ou autre). Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions en plus de la fiabilité et la confidentialité. Il s'agit de garantir l'intégrité et l'authenticité des données échangées. Les postulats de sécurité associés à la cryptographie sont :

a. Intégrité

Vérifier l'intégrité des données consiste à déterminer si les données, ressources, traitements ou services n'ont pas été altérées durant la communication de manière fortuite ou intentionnelle.

b. Confidentialité

La confidentialité est le maintien du secret des informations. Elle consiste à rendre l'information discrète ou inintelligible à d'autres personnes que les seuls acteurs de la transaction. La confidentialité peut être vue comme la «protection des données contre une divulgation non autorisée» [Gher, 2004].

c. Authentification

L'authentification consiste à assurer l'identité d'un utilisateur c.à.d. de garantir à chacun de correspondant que son partenaire est bien celui qu'il croit être.
On distingue deux types d'authentification :

- ***Le contrôle d'accès :*** C'est l'opération permettant d'être certain de l'identité d'une personne utilisateur pour permettre l'accès à des ressources uniquement pour la personne autorisée (par exemple l'utilisation d'un mot de passe pour un disque dur).
- ***Authentification de l'origine des données :*** Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services, l'authentification et l'intégrité.

d. La non répudiation (traçabilité)

La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction ; l'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information.

I.2.3. Problèmes de la cryptographie

Quelque soit le cryptosystème utilisé pour le chiffrement, ceci reste toujours cassable un jour ou l'autre. La sécurité d'un cryptosystème repose en fait sur la complexité des algorithmes définis et sur les puissances de calcul disponibles pour une attaque. Ainsi la solution adoptée actuellement est de faire « *retarder le travail des cryptanalystes* », c.à.d. faire en sorte que la durée nécessaire pour déchiffrer un code soit supérieure à la durée de validité des données.

I.3. Algorithmes cryptographiques

Au cours d'un échange visant à communiquer de façon secrète deux protagonistes, appelé ici l'émetteur et le récepteur à travers un canal peu sûr, l'information à transmettre sera donc chiffrée par un procédé de chiffrement en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de en utilisant une clé de déchiffrement sans laquelle son procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique : *algorithme cryptographique* (ou *chiffre***)**. La sécurité des données chiffrées est entièrement dépendante de deux facteurs : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un *cryptosystème***.**

I.3.1. Le principe de Kerckhoffs

Pour briser un cryptosystème, un opposant cherche à obtenir deux éléments d'information :

1. Quel est le type de système de codage utilisé ? et,
2. Quelle est la clé d'encodage utilisée ?

Bien entendu, son travail est simplifié (mais certainement pas terminé) s'il connait le type de système utilisé. Avec le temps cette information finit par circuler. Cette hypothèse de travail est appelée le principe de Kerckhoffs. Ce principe consiste à affirmer que la sécurité d'un système de chiffrement ne devrait pas être fondée sur le secret de la procédure utilisée, mais essentiellement sur le secret de la clé.

Auguste Kerckhoffs écrit en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire », où il disait [Kerc, 1883] :

« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants:

1) le système doit être matériellement, sinon mathématiquement, indéchiffrable.
2) il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.
3) la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
4) il faut qu'il soit applicable à la correspondance télégraphique.
5) il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
6) enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer ».

Les points 2 et 3 sont les axiomes fondamentaux de la cryptographie suivant lesquels l'attaquant possède tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé spécifique pour le chiffrement. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ce secret sera découvert ou volé.

I.3.2. Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplait (P, C, K, E, D) satisfaisant les points suivants :

1) P est un ensemble fini de blocs de textes clairs possibles.
2) C est un ensemble fini de blocs de textes chiffrés possibles.
3) K est un ensemble fini de clefs possibles.
4) Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k (e_k (x)) = x$ pour tout texte clair $x \in P$.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$ qui doit être transmise préalablement loin des yeux d'Oscar (à travers un canal de communication sûr). Supposant qu'Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne : $x = x_1 x_2 \ldots x_n$ avec : $n \in Z$, $n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera : $y = y_1 y_2 \ldots y_n$.

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1 x_2 \ldots x_n$. Le procédé de communication est illustré sur la Figure I.2.

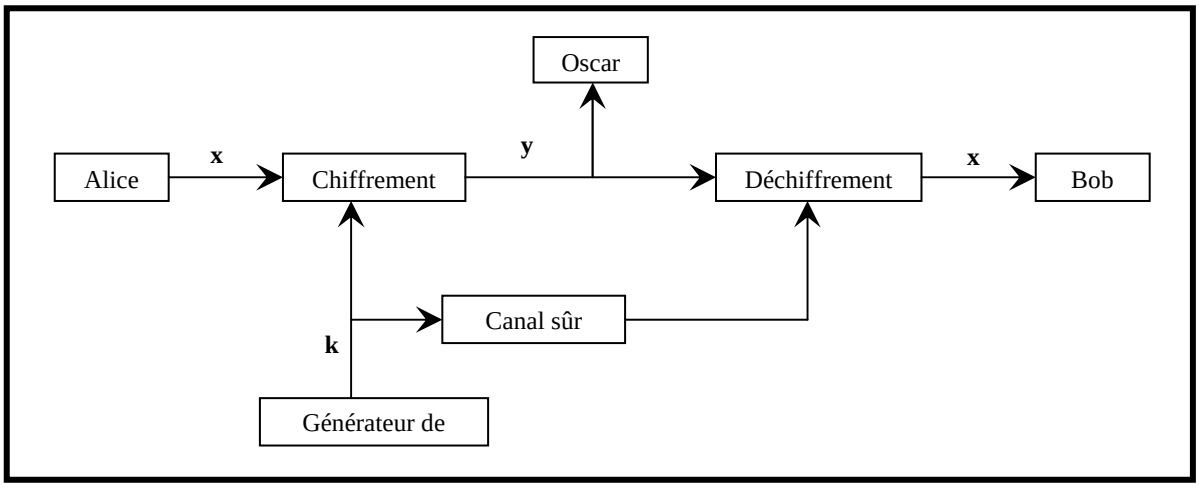


Figure I.2. *Le procédé de communication.*

I.3.3. Classes de cryptographie

Le schéma suivant illustre les différentes classes de la cryptographie :

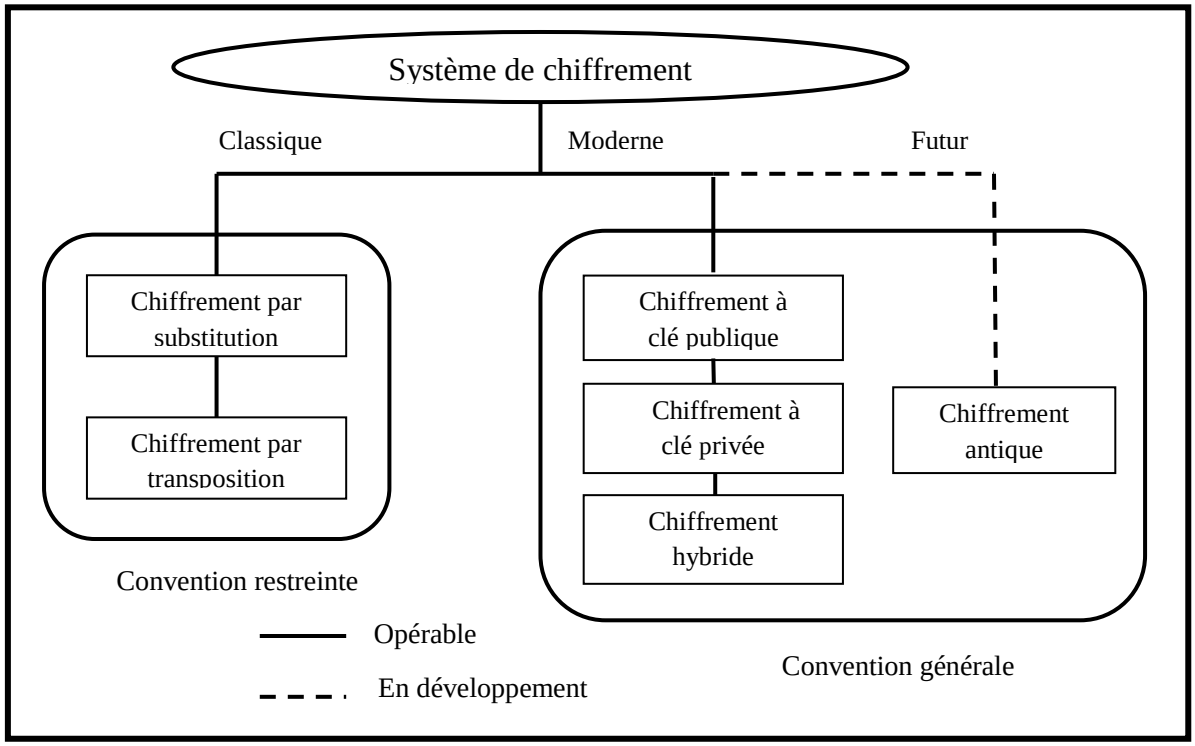


Figure I.3. *Les classes de la cryptographie*

I.3.3.1. La cryptographie classique

Les premiers algorithmes utilisés pour le chiffrement d'une information étaient assez rudimentaires dans leur ensemble et ils sont trop simples pour offrir la moindre sécurité. Pour cacher la substance d'un texte, ils utilisent la substitution de caractères par d'autres ou les transposer dans des ordres différents. De ce fait, la confidentialité de l'algorithme de chiffrement était donc la pierre angulaire de ce système pour éviter un décryptage rapide. On appelle généralement cette classe de méthodes : le chiffrement **à usage restreint**.

a. Cryptographie par substitution

La substitution signifie que chaque lettre (ou groupe de lettres) est substituée par une (ou groupe) lettre(s), chiffre(s) ou symbole(s). Le déchiffrement consiste à effectuer la substitution inverse. Selon la façon de substituer, on quatre catégories :

- **Substitution simple (mono-alphabétique)**

Le codage par substitution mono-alphabétique (ou encore les alphabets désordonnés) est le plus simple à imaginer. Chaque lettre dans le message clair est remplacée dans le message chiffré par une autre lettre différente unique pour toutes les occurrences de celle-ci.

Dans la littérature, plusieurs algorithmes ont été proposés, entre autres, nous citons : le chiffre de César, le chiffre Atbash, le carré de Polybe, etc.

- **Substitution poly-alphabétique**

Au lieu de remplacer une lettre par une même autre lettre dans tout le message comme dans la substitution simple, elle est remplacée périodiquement par différentes lettres. L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles.

- **Substitutions homophoniques**

Au lieu d'associer un seul caractère crypté à un caractère en clair, on dispose d'un ensemble de possibilités de substitution de caractères dans lequel on choisit aléatoirement. Par exemple : C=>S, K ; G=>G, J ; Q=>K ; S=>S, Z ; PH=>F ; …etc.

- **Substitution par polygrammes**

Au lieu de substituer des caractères, on substitue par exemple des digrammes : groupe de deux caractères. Pour se faire, deux moyen sont utilisés : soit par table (Chiffre de Playfair) ou par transformation mathématique (Chiffre de Hill).

b. Cryptographie par transposition

Elle consiste à permuter les lettres du message à chiffrer entre elles, afin de le rendre inintelligible. Plusieurs variations de transposition sont utilisées, parmi eux on trouve :

- **Transposition simple (à base matricielle)**

Elle consiste à écrire le texte en clair dans une matrice de ***n*** colonnes (une lettre dans chaque case), et ensuite de construire le texte chiffré en prenant les lettres à partir de cette matrice colonne par colonne. La clé dans ce cas est le nombre ***n***.

- **Transposition avec substitution simple**

L'idée dans ce cas est de combiner la transposition avec une substitution simple. Il s'agit ainsi de chiffrer le message clair par une méthode de substitution simple, et en suite d'en appliquer une transposition. Une autre astuce est souvent utilisée qui consiste à appliquer une fonction de permutation sur l'ordre d'arrangement des colonnes. On cite à titre d'exemple : le chiffre de DELASTELLE.

c. Conclusion

Les nouvelles techniques de communications (moyens de transports rapides, journaux, télégraphe, télégraphie sans fil) donne une nouvelle impulsion à la cryptologie. L'interception devient simple et le décryptement des informations devient vital. La cryptologie entre dans son ère moderne et les algorithmes classiques ne sont plus efficaces.

I.3.3.2. La cryptographie moderne

La cryptographie entre dans son ère moderne avec l'utilisation intensive des ordinateurs à partir des années septante. Vue la nécessité croissante de sécuriser les données dans tous les domaines (économique, industriel, informatique,...etc.) …. La cryptographie est appelée à devenir une technique de plus en plus fondamentale pour la protection des informations possédées et/ou échangées par des individus ou des organisations. Dans la cryptographie moderne, on utilise aussi des problèmes mathématiques que l'on ne sait pas (encore) résoudre, par exemple factoriser des grands nombres (chiffre RSA).

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ la *cryptographie à clef secrète,* ou encore appelée *symétrique,*
- ✓ la *cryptographie à clef publique,* dite également *asymétrique*.

a. La cryptographie symétrique

- **Principe**

Le cryptage à clé privée ou symétrique (ou encore dit conventionnel) utilise la même clé pour chiffrer et déchiffrer un message. Autrement dit, les clefs de chiffrement et de déchiffrement sont identiques ou on peut facilement calculer l'une à partir de l'autre. La connaissance de cette clef est cruciale pour la confidentialité des informations échangées. L'emploi d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Ces systèmes sont vulnérables parce qu'il est généralement possible de découvrir la clé à partir des messages codés. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1}. En excepte le système à masque jetable qui est hors porté de cette attaque.

Les algorithmes symétriques sont de deux types :

- ✓ Les algorithmes de *chiffrement en continu*, qui agissent sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé *chiffrement en flux* (Stream cipher).
- ✓ Les algorithmes de *chiffrement par blocs* (Bloc cipher), qui opèrent sur le texte en clair par groupes de bits appelés blocs.

Plusieurs algorithmes de chiffrement symétriques ont été définis. Nous présentons, ci-dessous, les plus connues de ces méthodes.

- **Quelques algorithmes de chiffrement symétrique**

A ce niveau, on va présenter, plus ou moins en détails, les plus fameux des algorithmes de chiffrement s'inscrivant sous ce mode.

1) Masque jetable (one-time pad)

Ce chiffre appelé aussi chiffre de Vernam ou encore le chiffrement parfait; est un algorithme de cryptographie inventé par Gilbert Vernam en 1917. Ce chiffrement est le seul qui soit théoriquement impossible à casser puisque la sécurité de ce système repose sur la génération complètement aléatoire de la clé [www2]. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront équiprobables. Malgré cela, il présente d'importantes difficultés de mise en œuvre pratique, il ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

Il consiste à combiner le message en clair avec une clé présentant les caractéristiques suivantes :

- choisir une clé K_M aussi longue que le texte à chiffrer.
- utiliser une clé constituée de caractères choisis aléatoirement.
- ne jamais utiliser 2 fois la même clé (d'où le nom de masque jetable).
- pour chiffrer un message faire le ou exclusif du message et de la clé : $C=M \oplus K_M$.
- pour déchiffrer un message l'opération est la même : $M= C \oplus K_M = M \oplus K_M \oplus K_M$.

2) DES (Data Encryption Standard)

Jusqu'aux années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence de besoins civils, le NBS (National Bureau of Standards) lança le 15 mai 1973 un appel d'offres dans le Federal Register (l'équivalent du Journal Officiel américain) [Stin, 1996] pour la création d'un système cryptographique qui doit :

- Reposer sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement.
- Etre facile à implémenter, logiciellment et matériellement et être très rapide aussi.
- Avoir un haut niveau de sécurité, lié uniquement à la clé et non à sa confidentialité.

Ce cryptosystème permet de chiffrer des messages de 64 bits avec une clef de 56 bits. De ce fait, c'est un système de chiffrement par blocs. Pour chiffrer un texte, il faut d'abord le découper en blocs de 64 bits puis appliquer le chiffrement sur chacun des blocs. Ainsi, les données en entrée de cet algorithme (texte clair et les données en sortie (texte chiffré) seront des bocs de 64 bits.
Sa clé est une chaîne binaire de 64 bits, mais en fait seuls 56 bits servent réellement à définir la clé. Les bits 8, 16, 24, 32, 40, 48, 56, 64 sont des bits de parité. Le $8^{ème}$ bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Ceci permet d'éviter les erreurs de transmission. Il y a donc pour DES 2^{56} clés possibles, soit environ 72 milliards possibilités. Malgré ça, c'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, alors qu'elle a été largement suffisante au moment de sa conception.

Nous donnons ci-dessous une idée simple et intuitive du fonctionnement du DES qui est un algorithme relativement simple puisqu'il combine des permutations et des substitutions. Il se déroule en quatre étapes [www3] :

1. **Préparation-Diversification de la clé :** le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K 16 sous-clésK_1,…, K_{16} à 48 bits.
   La figure ci-dessous montre comment obtenir à partir d'une clé de 64 bits 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES.

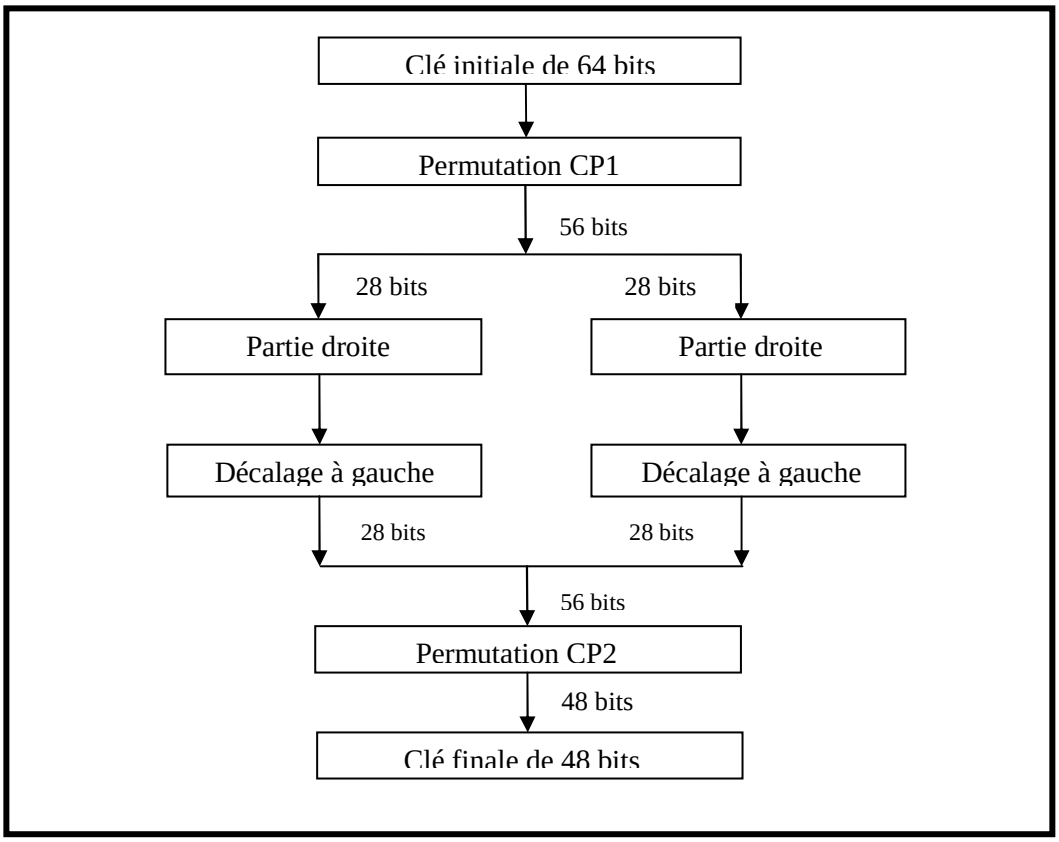


Figure I.4. *Génération des clés.*

La première étape consiste à faire une permutation noté CP1 dont la matrice est présentée ci-dessous, puis éliminer les bits de parité afin d'obtenir une clé d'une longueur de 56 bits. Elle est composée de deux matrice G_i et D_i chacune de 28 bits. Ces deux blocs subissent ensuite une rotation à gauche. Et enfin, ces derniers sont regroupé en un bloc de 56 bits ensuite une permutation CP2 a eu place pour fournir en sortie une clé de 48 bits. Des itérations de l'algorithme permettent de donner les 16 clés K_1, …, K_{16}.

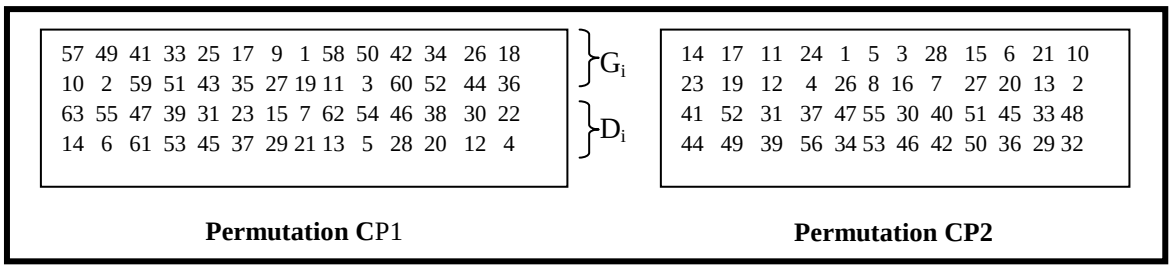


Figure I.5. *Permutation CP1 et CP2.*

2. **Permutation initiale :** pour chaque bloc de 64 bits x du texte, on calcule une permutation finie y=PI(x). y est représenté sous la forme : y = G_0 D_0 , G_{16} étant les 32 bits à gauche de y, D_0 les 32 bits à droite (voir Figure I.6). Quant à leurs significations, par exemple, la permutation initiale (IP) signifie que le $58^{ème}$ bit de la chaine à chiffrer, *x*, est le premier bit de IP(*x*),…

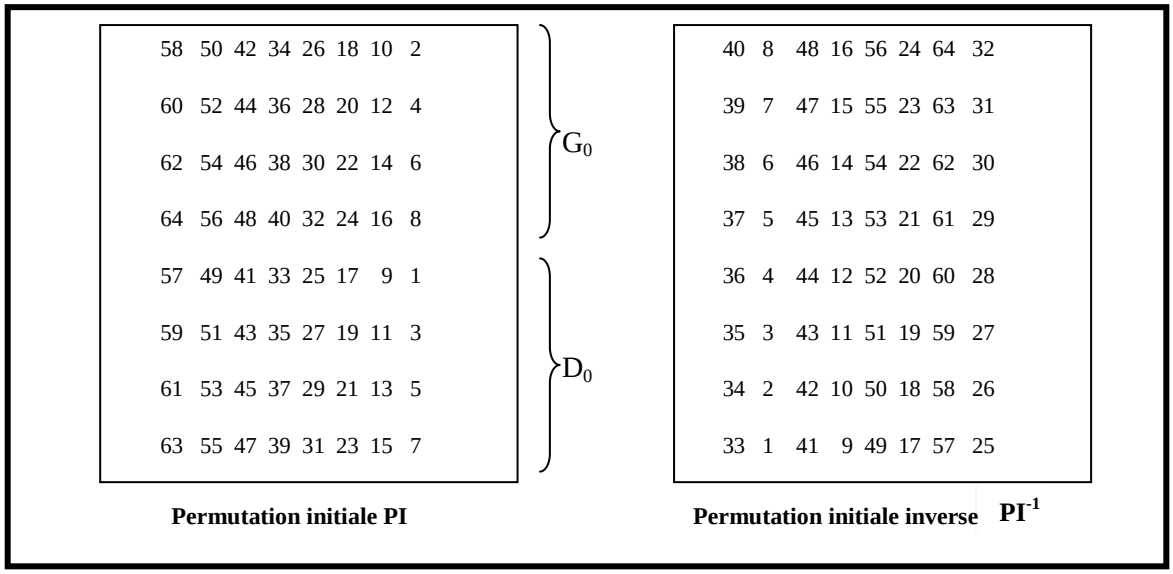


Figure I.6. *La permutation initiale et son inverse.*

3. **Itérations (Rondes) :** On applique 16 rondes d'une même fonction. A partir de $G_{i-1}D_{i-1}$ (pour i de 1 à 16), on calcule G_iD_i en posant :
   - $G_i = D_{i-1}$
   - $D_i = G_{i-1} \oplus f(D_{i-1},K_i)$

D'après la formule qui correspond au calcul de D_i, on constate que la fonction *f* utilise deux arguments ayant des tailles différentes : D_{i-1} de 32 bits et k_i de 48 bits. Ainsi, D_{i-1} sera étendu en 48 bits grâce à une matrice appelée table d'expansion (notée E), dont les 48 bits sont mélangés et 16 d'entre eux sont dupliqués (voire Figure I.7).

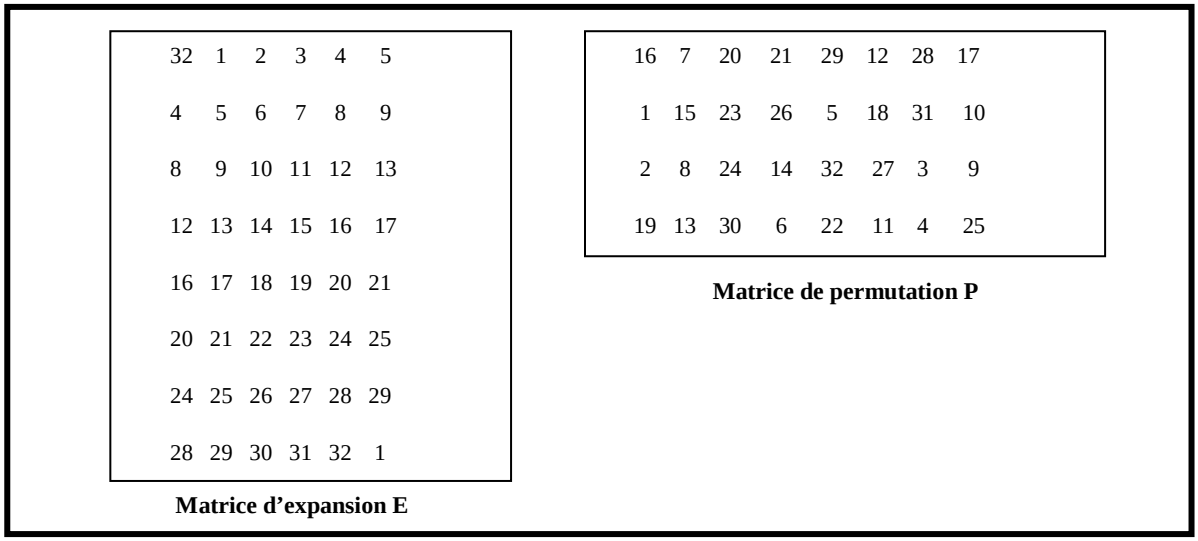


Figure I.7. *Matrice d'expansion E et de permutation P.*

L'opération qui suit consiste à calculer un « ou exclusif » entre le résultat obtenu jusqu'à maintenant, codé sur 48 bits, et G_{i-1} codé sur 32 bits. Donc, le résultat final de *f* doit être codé sur 32 bits au lieu de 48 bits. Pour cela, la chaine de $48 = 8 \times 6$ bits sera transformée en une chaîne de $32 = 8 \times 4$ bits en utilisant des dispositifs appelés *boites-S*. Elles sont au

nombre de huit, où chacune calcule un bloc de 4 bits à partir d'un bloc de 6 bits. Enfin, on applique une permutation à ce 32 bits pour obtenir la valeur finale de *f* (voir la Figure I.7).
L'ensemble de ces résultats en sortie de P est soumis à un « ou exclusif » avec le G_0 de départ (comme il est indiqué dans la Figure I.9) pour donner D_1, tandis que le D_0 initial donne G_1, et ainsi de suite pour les 16 itérations comme il est mentionné auparavant.
La succession d'opérations constituant la fonction ***f*** est représentée à travers la Figure I.8**.**

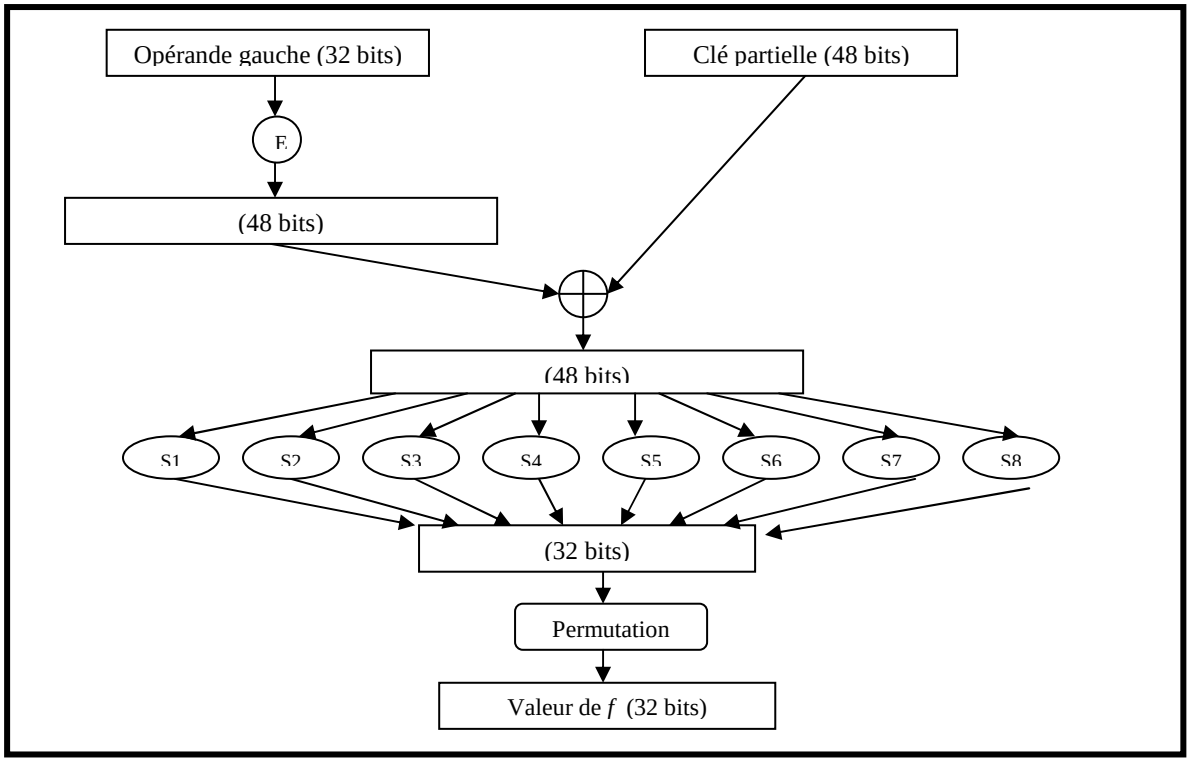


Figure I.8. *Schéma de la fonction f.*

4. **Permutation initiale inverse :** A la fin des itérations, les deux blocs G_{16} et D_{16} sont récoltés, puis soumis à la permutation initiale inverse : $Z = PI^{-1}(G_{16}\ D_{16})$. Le résultat en sortie est un texte codé de 64 bits.
   Cette description peut être résumée par le schéma général illustré par Figure I.9, où on a seulement représenté quelques-unes des 16 étapes.

Remarque : le déchiffrement suit le même algorithme avec la même clef *K*. Seules les sous-clés sont appliquées dans le sens inverse en commençant par la clé k_{16} jusqu'à la clé k_1.

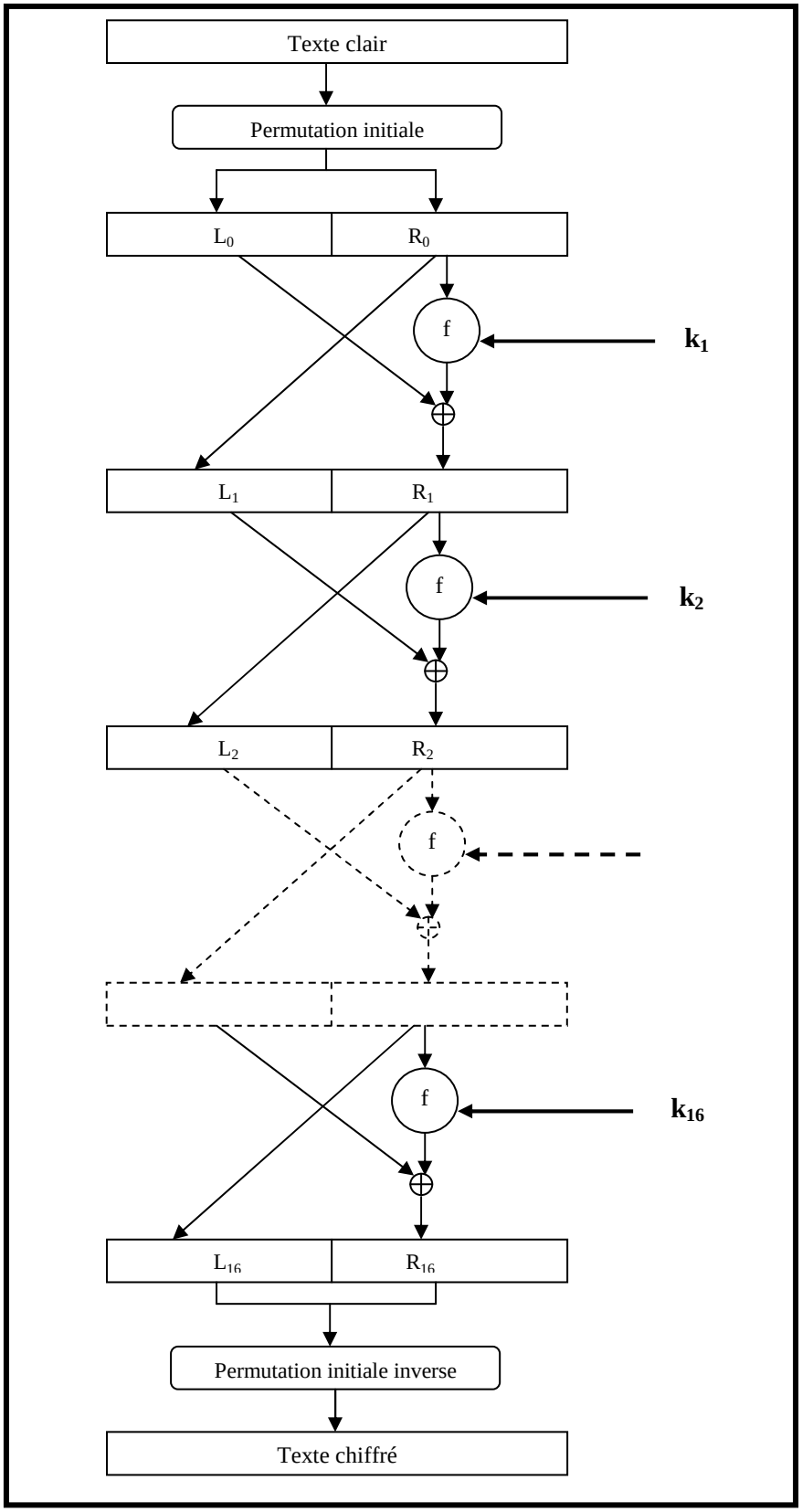


Figure I.9. *Schéma général de DES.*

Le DES a fait l'objet de très nombreuses attaques. On peut citer quelques une :

- **Cryptanalyse différentielle :** c'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires. Ainsi, un DES à 8 ou à 10 tours peut facilement être cassé, mais le DES complet à 16 tours est resté hors de portée de cette attaque.

- **Cryptanalyse linéaire :** c'est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Elle n'est utilisable que pour un DES restreint à quelques tours, mais le DES réel n'est pas menacé par cette attaque.

- **Recherche exhaustive :** les laboratoires RSA ont lancé en Janvier 1997 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES pour démontrer que la taille des clés DES a devenu insuffisante. Ils ont réussi le 17 Juin 1997. Bien qu'une telle recherche demande des moyens considérables, elle a montré que le DES n'offre plus aujourd'hui une grande sécurité.

3) Triple DES (3DES)

Pour palier l'insuffisance cryptographique observée du cryptosystème DES, dû à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. Triple DES est apparait comme une solution pour remédier les faiblesses de DES. Comme son nom l'indique, le principe du triple DES est d'effectuer 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes. Ce principe peut être formulé comme suit :

$$\textbf{Triple-DES}_{k1,\,k2} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}^{-1}_{k2} \;\textbf{O}\; \textbf{DES}_{k1}$$

On peut constater que le DES sera retrouvé comme cas particulier de la formule ci-dessus, lorsque $k_1 = k_2$. Le déchiffrement de son tour est formulé par :

$$\textbf{Triple-DES}^{-1}_{k1,\,k2} = \textbf{DES}^{-1}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}^{-1}_{k1}$$

Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc composée de 112 bits. Une autre variante à trois clés DES différentes peut être conçue. D'une façon plus formelle, son principe peut être donné par la formule suivante :

$$\textbf{Triple-DES}_{k1,\,k2,\,k3} = \textbf{DES}_{k1} \;\textbf{O}\; \textbf{DES}_{k2} \;\textbf{O}\; \textbf{DES}_{k3}$$

Malgré cela, cette variante reste aussi fragile à une attaque de coût en 2^{112} s'appuyant sur l'un des deux messages intermédiaires.

4) AES (**Advanced Encryption Standard**)

Avec le temps et les progrès de l'informatique, les 2^{56} clés possibles du DES se voient incapables de fournir une barrière infranchissable dû à la faiblesse des clés de 56 bits. Désormais avec des moyens modestes, percer les messages chiffrés par DES en un temps raisonnable, ne pose aucun problème. De ce fait, l'AES a vu le jour. Il est issu d'un appel à candidatures international lancé en janvier 1997 et ayant reçu 15 propositions. Au bout de cette évaluation, ce fut le candidat Rijndael du nom de ses deux concepteurs Joan Daemen et Vincent Rijmen [www4] qui a été choisi par le NIST en Octobre 2000 pour être l'algorithme AES et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. II a été déclaré vainqueur de la deuxième ronde dans laquelle

s'opposaient les 5 candidats finalistes (MARS, RC6, Rijndael, Serpent, TwoFish). L'AES a une taille longue de sa clé allant jusqu'à 256 bits qu'il le permet de résister toujours a la cryptanalyse. L'AES est devenu le nouveau standard du chiffrement symétrique comme le signifie cet acronyme. L'AES est libre d'utilisation comme l'est l'algorithme DES.

Techniquement, le chiffrement AES travaille avec des blocs de 128 bits seulement et la longueur des clés utilisées peut être de 128, 192 ou de 256 bits. Chaque bloc subit une séquence de transformations que nous résumons à travers les points suivants :

1. Addition de la clé secrète et du bloc en question avec un « ou exclusif ».
2. ByteSub : les 128 bits sont répartis en 16 blocs de 8 bits (16 octets), qui sont ensuite placés dans une matrice de 4×4. Chaque octet est transformé par une fonction non linéaire S (S-box) conçu pour résister à la cryptanalyse linéaire et différentielle.
3. ShiftRow: les lignes de cette matrice sont soumises à une rotation vers la droite où l'incrément pour la rotation varie selon le numéro de la ligne. La $2^{ème}$ ligne est décalée d'une colonne, la $3^{ème}$ ligne de 2 colonnes, et la $4^{ème}$ ligne de 3 colonnes.
4. MixColumn: chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne. Cela revient à multiplier la matrice 4×4 par une autre matrice 4×4.
5. AddRoundKey: une clé dite *de tour* est générée à partir de la clé secrète par un sous-algorithme dit *de cadencement*. Cette clé de tour est ajoutée par un « ou exclusif » au dernier bloc obtenu.

Ces différentes opérations, définissant un *tour*, sont répétées plusieurs fois. Cependant dans le dernier tour il n'y a pas d'opération MixColumn. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours. La figure suivante résume le principe de fonctionnement de cet algorithme de chiffrement.

Le déchiffrement consiste en appliquer les opérations inverses dans chacune des étapes (*InvSubBytes*, *InvShiftRows*, *InvMixColumns*). *AddRoundKey* (à cause du XOR) est son propre inverse. On réitère ce processus le nombre de tour-1. Pour le dernier tour on exclu l'opération *InvMixColumns* ; et comme dans le chiffrement pas d'opération *InvMixColumns* dans le dernier tour.

De nombreux travaux de cryptanalyse ont été publiés mais pour l'instant il n'a pas été cassé et la recherche exhaustive demeure la seule solution. En particulier, on cite :

- **Attaques sur des versions simplifiées :** *Niels Ferguson* et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours. En effet, cette dernière attaque repose sur le principe des clés apparentées. Dans une telle attaque, la clé demeure secrète mais l'attaquant peut spécifier des transformations sur la clé et chiffrer des textes à sa guise. Il peut donc légèrement modifier la clé et regarder comment la sortie de l'AES se comporte.

- **Attaques sur la version complète :** plusieurs chercheurs ont mis en évidence des possibilités d'attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL. Le XSL fait appel à une analyse heuristique dont la réussite n'est pas systématique. Elles sont impraticables car le XSL demande au moins 2^{87} opérations voire 2^{100} dans certains cas. Le principe est d'établir les équations (quadratiques / booléennes) qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer et l'AES est toujours considéré comme sûr.

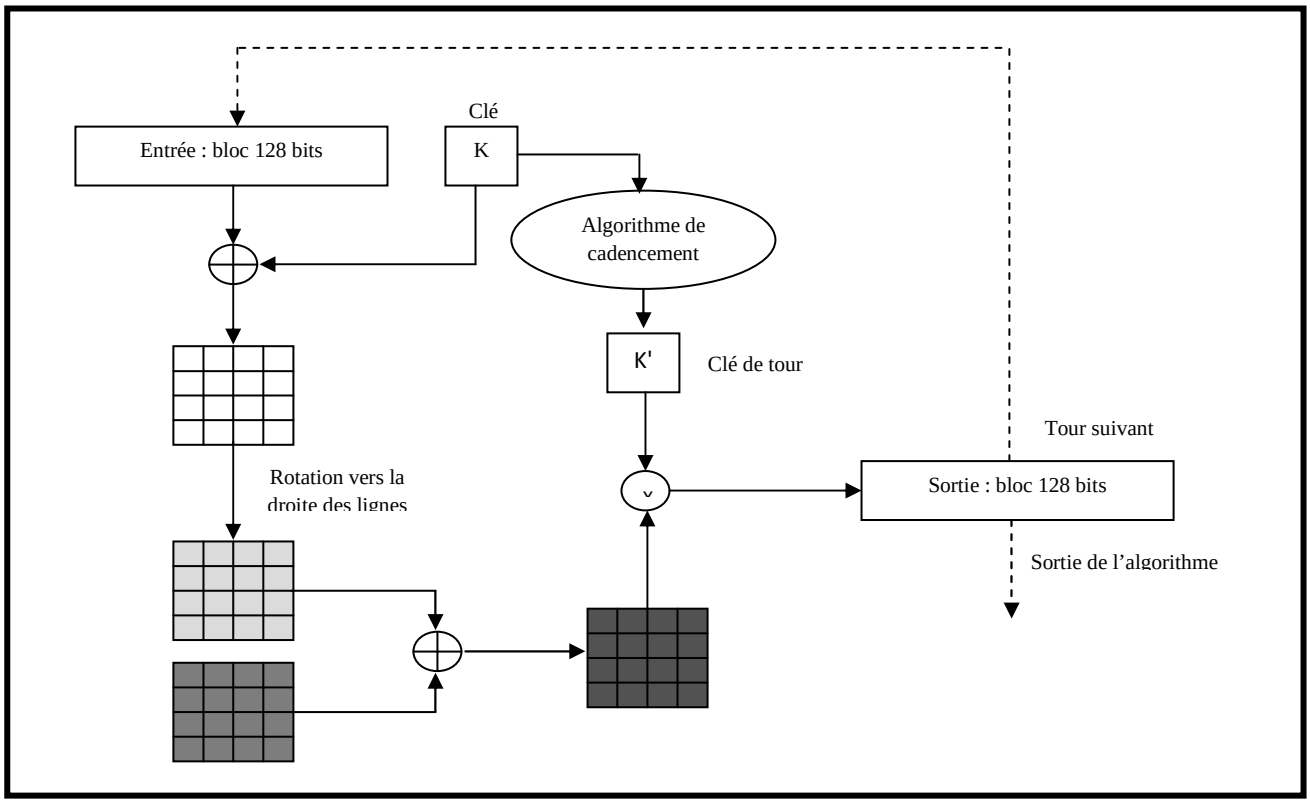


Figure I.10. *Schéma général de l'AES.*

5) IDEA (International Data Encryption Algorithm)

L'**IDEA** qui est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 rondes inventé en 1990 à Zürich par James L. Massey et Xuejia Lai. C'est la taille des clés qui apporte une grande sécurité au système (que l'on n'a pas, à l'heure actuelle, officiellement réussi à "casser").

Chaque bloc est divisé en 4 sous-blocs de 16 bits : X_1, X_2, X_3 et X_4. Ces quatre sous-blocs deviennent les entrées de la première ronde de l'algorithme.
À chaque ronde, la séquence d'évènements est la suivante (voir Figure I.11) :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé ;
5. combinez par OU exclusif les résultats des étapes (1) et (3) ;
6. combinez par OU exclusif les résultats des étapes (2) et (4) ;
7. multipliez le résultat de l'étape (5) avec la cinquième sous-clé ;
8. additionnez les résultats des étapes (6) et (7) ;
9. multipliez le résultat de l'étape (8) par la sixième sous-clé ;
10. additionnez les résultats des étapes (7) et (9) ;
11. combinez par OU exclusif les résultats des étapes (1) et (9) ;
12. combinez par OU exclusif les résultats des étapes (3) et (9) ;
13. combinez par OU exclusif les résultats des étapes (2) et (10) ;
14. combinez par OU exclusif les résultats des étapes (4) et (10).

La sortie de la ronde est constituée des 4 sous-blocs produits par les étapes (11), (13), (12) et (14). Changez les deux blocs intérieurs (sauf lors de la dernière ronde) et cela donne l'entrée de la ronde suivante.
Après la huitième ronde, il y a une transformation finale :

1. multipliez X_1 et la première sous-clé ;
2. additionnez X_2 et la deuxième sous-clé ;
3. additionnez X_3 et la troisième sous-clé ;
4. multipliez X_4 et la quatrième sous-clé.

Enfin les 4 sous-blocs sont réassemblés pour former le texte chiffré.

Chaque ronde utilise 6 sous-clés K_i^r de 16 bits : K_1^1, ..., K_6^1, …, K_1^8, …, K_6^8. La transformation finale utilise 4 sous-clés : $K_1^9, K_2^9, K_3^9, K_4^9$; donc un total de 52 sous-clés est utilisé, les premiers 8 sous-clés sont extraites directement à partir de clé, des groupes de 8 clés sont extraits par rotation à gauche de 25 bits de la clé K, ce qui donne 6 rotations en total.
Concernant le processus de déchiffrement on utilise celui de chiffrement avec un seul changement dans la génération des clés. En fait, on utilise K pour générer les sous clés K_i^r ; à partir de ces derniers, d'autres clés $K_i^{'r}$ sont obtenues dans le Tableau I.1 [Omar, 2006] ; ensuite on utilise $K_i^{'r}$ à la place des K_i^r dans l'algorithme de chiffrement IDEA. Dans le Tableau I.1, $-K_i$ dénote l'opposé modulo 2^{16} de K_i. K_i^{-1} dénote l'inverse multiplicative de $K_i \bmod(2^{16}+1)$, se trouvant aussi dans {0, 1, …,216-1}.

Ronde r	$K_1^{'(r)}$	$K_2^{'(r)}$	$K_3^{'(r)}$	$K_4^{'(r)}$	$K_5^{'(r)}$	$K_6^{'(r)}$
r =1	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
2 ≤ r ≤ 8	$(K_1^{10-r})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$(K_4^{10-r})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
r= 9	$(K_1^{10-r})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$(K_4^{10-r})^{-1}$	--	--

Tableau I.1. *Les sous clés de déchiffrement $K_i^{'r}$ générées à partir des sous clés K_i^r.*

La méthode de génération des sous-clés de l'IDEA est toujours régulière et donc pourrait être une faiblesse à l'algorithme. Cependant, il est considéré comme étant hautement sécuritaire. En effet, pour résoudre IDEA avec l'attaque en force brute, il faudrait effectuer 2^{128}, donc 10^{38} opérations [www5].

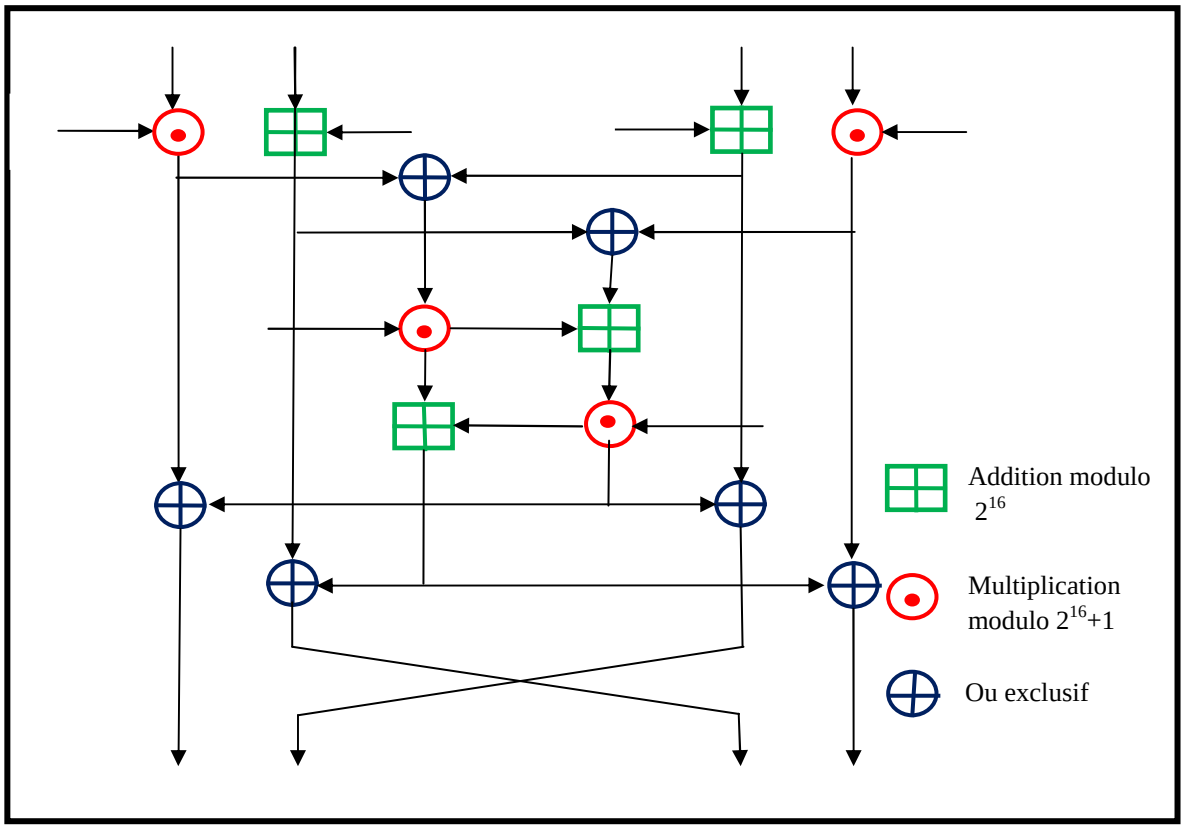


Figure I.11. *Schéma général de l' IDEA.*

- **Conclusion**

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. AES est immunisé contre les attaques et il reste néanmoins le meilleur choix dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause. D'autre part, IDEA est considéré par les spécialistes comme l'un des meilleurs cryptosystèmes à clé privée. Il est utilisé par le PGP pour le chiffrement de données.

b. La cryptographie asymétrique

- **Principe**

La *cryptographie asymétrique* ou encore dite *à clé publique* utilise deux clés différentes pour chaque utilisateur : une est privée et n'est connue que de l'utilisateur et qui est sensé d'être en mesure de faire signature ou déchiffrement. L'autre est publique diffusée en général dans un annuaire et donc accessible par quiconque afin de permettre aux interlocuteurs de mettre en œuvre les opérations réciproques (vérification de signature ou chiffrement de message). Les clés, publique et privée sont mathématiquement liées par l'algorithme de cryptage de telle manière qu'un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de *fonction à sens unique avec trappe*. Sachons qu'une fonction est appelée à *sens unique* si elle est

aisément calculée, mais extrêmement difficile de déduire la fonction inverse. Et elle sera dite *à trappe*, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la *trappe*. L'utilité d'utilisation d'une telle fonction réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas assez facile puisqu'elles s'appuient généralement sur des problèmes mathématiques réputés difficiles.

Ce cryptage présente l'avantage de permettre le placement de signature numérique dans le message et ainsi permettre l'authentification de l'émetteur grâce à la fonction de hachage. Le principal avantage consiste à résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé puisque la clé privée n'est connue que par l'utilisateur. Bien que plus lent que la plupart des cryptages à clé privée il reste toujours préférable pour 3 raisons :

- ✓ plus évolutif pour les systèmes possédant des millions d'utilisateurs.
- ✓ authentification plus flexible.
- ✓ supporte les signatures numériques.

Les systèmes asymétriques les plus connus sont [Mene, 1996] : RSA basé sur la difficulté de la factorisation des grands entiers, El Gamal basé sur la difficulté de résoudre le problème du logarithme discret dans un corps fini et les systèmes sur les courbes elliptiques basés sur la difficulté de certains calculs sur les courbes elliptiques.

- **Quelques algorithmes asymétriques**

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivants :

1) RSA

La méthode de cryptographie RSA a été inventée en 1977. Elle titre son nom des noms de ses trois inventeurs: *R. Rivest, A. Shamir* et *L. Adleman* [www6]. Le RSA est le premier et est encore le système cryptographique à clé publique le plus utilisé de nos jours et toujours considéré comme sûr. Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers ; à l'heure actuelle, il est pratiquement impossible de les reconstituer en un temps raisonnable. Ainsi, la sécurité de RSA semble satisfaisante malgré qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser.

On peut résumer le fonctionnement de ce cryptosystème dans les étapes suivantes :

- **Fonction d'encodage E (publique) :** la clé publique k utilisée pour l'encodage comporte deux entiers: k = (e,n). L'opération de chiffrement se fait au moyen de l'élévation à la puissance **e** modulo n : $E_k(M) = M^e \bmod n$.
- **Fonction de décodage D (privée) :** la clé secrète k′ utilisée pour le déchiffrement est aussi un couple d'entiers : k′(d,n). Pour reconstituer le message initial, la fonction inverse d'encodage est appelée, elle est ainsi : $D_{k'}(M) = M^d \bmod n$.
- **Détermination des clés :**
  - **Détermination de n :** pour calculer n on doit initialement choisir deux entiers premiers *p* et *q* très grands et leurs valeurs sont secrètes connus que part l'utilisateur. Le choix de *p* et *q* affecte grandement le niveau de sécurité de RSA. Pour cela, il faut évidemment se prémunir contre les algorithmes de factorisation dont la complexité dépend essentiellement de la taille du plus petit facteur premier de *n* [Zimm, 2005], donc si possible choisir *p* et *q* de même taille.

- **Détermination de e :** pour calculer e, on calcule premièrement un entier z = (p-1)*(q-1) puis tout simplement choisir un entier e premier avec z.
- **Détermination de d :** pour calculer d, on procède ainsi : e*d≡1[z].

La sécurité de RSA est basée sur l'hypothèse que la fonction E est à sens unique, ce qui rend impossible à décrypter un texte chiffré. Mais à l'aide de la trappe qui est la factorisation n = p*q, il est possible d'en trouver d et donc la clé privée.

Depuis son apparition, plusieurs attaques ont été découvertes contre le RSA. Et même si aucune de ces attaques n'est réellement destructive, elles démontrent toutefois qu'il faut implémenter RSA avec beaucoup de précautions. La fameuse attaque est :

- **Recherche exhaustive :** comme la fonction de chiffrement RSA est déterministe, si l'ensemble des messages possibles est connu et de petite taille, il sera facile de décrypter par une recherche exhaustive [Ster, 2004]. Pour éviter cette attaque, il est indispensable de randomiser les messages avant chiffrement.

2) Chiffrement d'ElGamal

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste c.à.d. un même message M peut avoir plusieurs chiffres différents [Lafo, 2006]. Il est basé sur la difficulté du problème des logarithmes discrets c.à.d. la difficulté de trouver l'unique *a* noté $\log_\alpha ß$ / $0 \leq a \leq p-2$ tel que : $\alpha^a \equiv ß \pmod p$ où *p* est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$. Cependant, et pour éviter les attaques connues, *p* doit être convenablement choisi, et $p-1$ doit avoir un grand facteur premier.

D'une manière formelle, l'algorithme ElGamal peut être résumé comme suit :

Soit *p* un nombre premier tel que le problème du logarithme discret dans Z_p soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et $K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod p\}$
Les valeurs *p*, α et ß sont publiques, et *a* est secret.
Pour k = (*p*, α, *a*, ß), et pour un $k \in Z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$\mathbf{e_k(x, k) = (y_1, y_2)}$$

Où

$$\mathbf{y_1 = \alpha^k \bmod p}$$

Et

$$\mathbf{y_2 = x\, ß^k \bmod p}$$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit comme suit :

$$\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$$

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par $ß^k$, en produisant y_2 ;
- ✓ la valeur α^k est également transmise en tant que partie du texte chiffré ;
- ✓ Bob, qui connaît l'exposant secret *a*, peut calculer $ß^k$ à partir de α^k. Il peut alors « enlever le masque » en divisant y_2 par $ß^k$ et obtenir le texte clair x.

Une attaque possible à ce cryptosystème est celle dite ***man in the middle***. Son principe dans le cas d'ElGamal fonctionnant avec le mode *Diffie-hellman* est illustré sur la figure I.12.

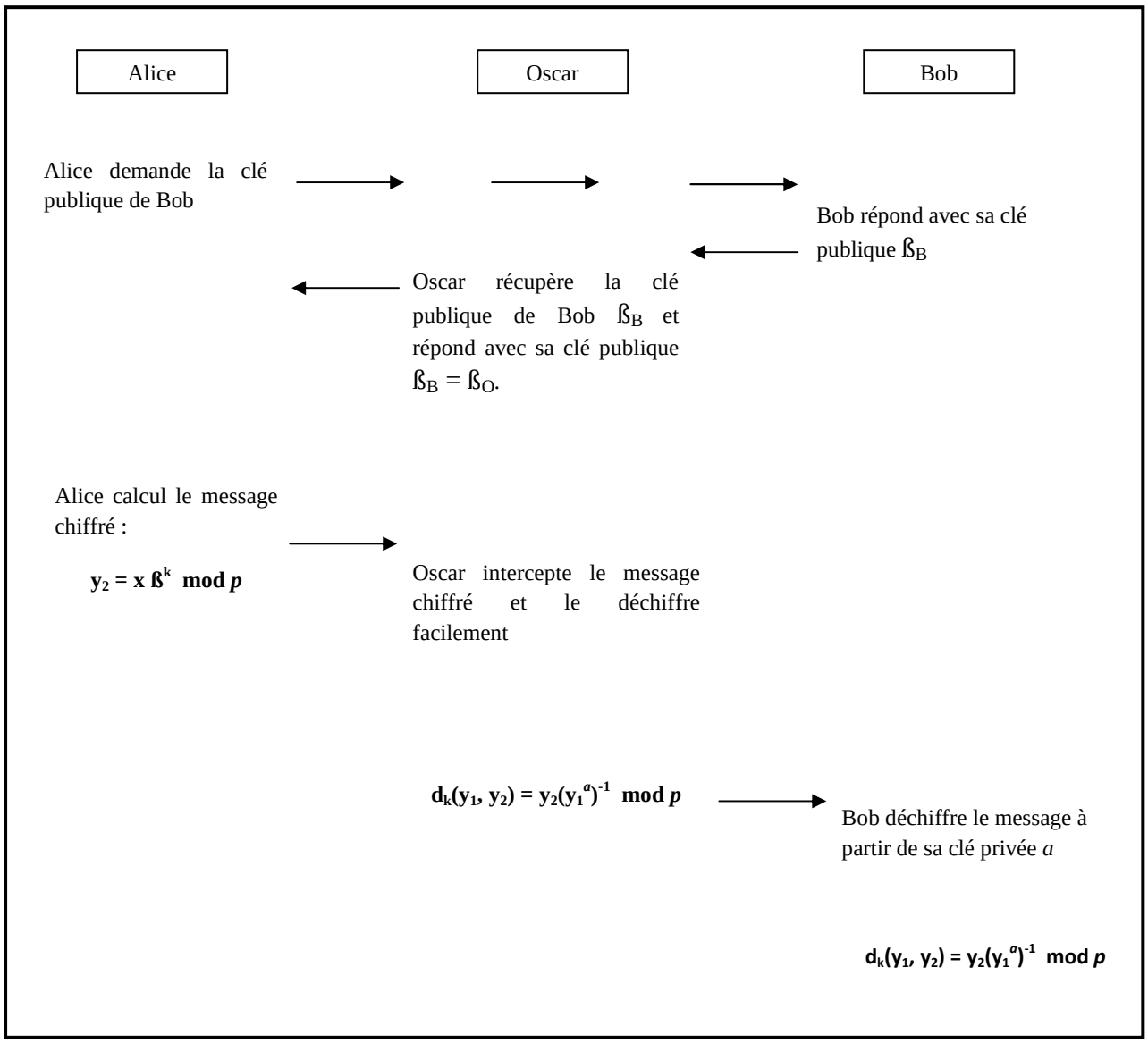


Figure I.12. *Principe de l'attaque « man in the middle ».*

- **Conclusion**

Les calculs faits en 1995 ont ouvert un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent des chiffres plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique serait capable de remettre en cause ce cryptosystème.

De même, casser l'algorithme ElGamal est aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Et pour assurer une bonne sécurisation de cet algorithme, Zimmermann en 2005 [Zimm, 2005] a recommandé l'utilisation des clés d'au moins 1024 bits.

c. Comparaison entre les cryptosystèmes symétriques et asymétriques

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage.

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ?
À partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que l'importance de la taille de clé n'est légitime que dans le cas de la clé secrète, puisque les seules attaques possibles sont les attaques exhaustives [www7]. Mais, dans le cas de la clé publique, la taille de la clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

Méthode	Avantages	Inconvénients
À clefs Secrètes	▪ Rapidité de calcul en général (dépend de la taille de la clé). ▪ Adaptée au cryptage de flux de données.	▪ Moins sécurisé (DES). ▪ Problème de communication de clefs entre émetteur et récepteur. ▪ Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clefs Publiques	▪ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre. ▪ Permet la signature électronique. ▪ Un couple de clés publique/privée suffisant pour 'n' correspondants.	▪ Lente. ▪ Problèmes de gestion de clefs publiques.

Tableau I.2. *Comparaison entre les méthodes de chiffrement symétriques et asymétriques.*

d. La cryptographie hybride

- **Principe**

La cryptographie asymétrique est beaucoup plus lente que la cryptographie symétrique qui brille par sa rapidité. En revanche, cette dernière souffre d'une grave lacune ; assurer une transmission secrète de la clé. Pour palier ce défaut et cumuler les avantages des deux méthodes on a fait recourt à la cryptographie hybride. On code tout d'abord les données avec une clé privée dite *clé de session*, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Comme la clé est courte, on utilise l'algorithme asymétrique puisqu'il prend peu de temps. En revanche, chiffrer l'ensemble du message avec un algorithme asymétrique serait plus lourd. Il suffit ensuite d'envoyer le message chiffré avec une clé privée et accompagné de cette dernière chiffrée avec une clé publique. Le destinataire procède inversement, il commence à déchiffrer la clé symétrique avec sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, via un déchiffrement symétrique pour retrouver le message original. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

- **Quelques algorithme hybride**

1) PGP (**Pretty Good Privacy**)

PGP est inventé par *Philip Zimmermann*, qui dise que tout individu a droit à la confidentialité, notamment les organisations des droits de l'homme dans des pays soumis à la dictature. Il l'a mis à disposition gratuitement sur Internet. Cela lui a valu de sérieux ennuis avec la justice américaine, car les logiciels de cryptage sont considérés comme du matériel de guerre et sont interdits à l'exportation.

Il est souvent utilisé pour garantir l'authentification et le contrôle d'intégrité des fichiers et du courrier électronique. Mais avant de crypter un texte avec PGP, les données doivent tout d'abord compressées dont le but de réduire le temps de transmission, ainsi d'économiser l'espace disque et, surtout, de renforcer la sécurité cryptographique puisque les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement alors que la compression réduit ces modèles dans le texte en clair.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes qui sont [www8] :

✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
✓ PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

L'opération de déchiffrement se fait également en deux étapes, qui sont [www8] :

✓ PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
✓ PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

Depuis 1978, la recherche universitaire civile a intensément attaqué la cryptographie à clé publique, sans pour autant réussir à la remettre en cause. Mais cela ne fournisse aucune garantie totale sur la sécurité de cette manière de chiffrer, car une attaque menée par le gouvernement, par exemple, qui ne se manque pas de ressources très développées ou même en utilisant quelques nouvelles percées mathématiques classées top-secret, peut tenir à bout ces cryptosystèmes conventionnels utilisés dans PGP.

Tout de même, l'optimisme semble justifié. Les algorithmes de clé publique, les algorithmes de contraction de message, et les chiffres par blocs utilisés dans PGP ont été conçus par les meilleurs cryptographes du monde [Loid, 2005]. Les chiffres de PGP ont subi des analyses de sécurité approfondies et des examens méticuleux de la part des meilleurs cryptographes dans le monde non classé top secret. De plus, et même si les chiffres par blocs utilisés dans PGP possèdent quelques faiblesses, la compression du texte clair utilisée avant le chiffrement réduit de façon considérable ces faiblesses.

2) GPG (**GNU Privacy Guard**)

GPG est la version GNU de PGP. En raison qu'il est sous licence GPL (General Public License) il permet la fourniture du code-source, la libre modification et la libre redistribution de ce code-source. Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

- **Conclusion**

Aujourd'hui, de nombreux gouvernements ont restreint l'usage du PGP, qui a été largement diffusé par son développeur, en pensant qu'un cryptage trop fiable ferait le jeu des terroristes et des trafiquants. Toutefois, il y'a pas mal d'applications qui utilisent encore ce système de chiffrement, telles que les paiements en ligne qui se font grâce au procédé SSL

fonctionnant selon le principe du PGP. De sa part, GPG, qui est un cryptosystème utilisant des algorithmes de chiffrement à clé publiques (DSA, RSA et ElGamal), est largement utilisé dans les communications par messagerie, c.-à-d. pour chiffrer des mails, ainsi que pour signer des données.

I.3.3.3. Cryptographie quantique

a. Principe

La cryptographie à base d'algorithmes aura toujours des faiblesses. Même si la cryptanalyse bloque devant un algorithme, la force brute pourra toujours décrypter n'importe quel code si on lui donne assez de temps. Même avec le plus solide des chiffrages, le contenu du message peut être subtilisé et dupliqué. Les clés, quant à elles, peuvent être volées en chemin ou présenter des faiblesses qui les rendent prévisibles.

Si maintenant on base notre cryptographie, non pas sur un algorithme mathématique, mais sur des lois de la physique quantique, il n'y plus de force brute qui peut briser un code ici. Le cryptage ne se trouve pas dans des formules, mais dans des photons, ainsi tout le monde peut accéder à des formules, mais pas tout le monde peut accéder aux photons sans que le message devienne inutile. De ce fait, l'information n'est plus sécurisée par des subterfuges mathématiques, mais plutôt par des lois fondamentales de physique. Au de-là, la cryptographie quantique a vu le jour.

La figure I.13 [Nave, 2002] présente un exemple relatif à la possibilité soit qu'un photon traverse le filtre ou pas, selon l'orientation de sa polarisation. Sachons qu'un filtre permet de distinguer entre les photons polarisés horizontalement (0°) et verticalement (90°) ; un autre entre les photons polarisés en diagonale (45°, 135°).

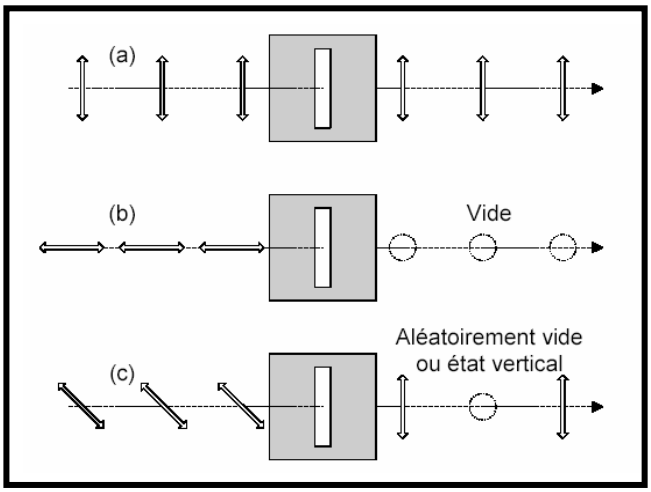


Figure I.13. *Photon unique traversant un filtre ne laissant passer que la lumière polarisée verticalement :*
(a) *Les états polarisés verticalement traversent le filtre sans être absorbés*
(b) *Les états polarisés horizontalement sont tous absorbés*
(c)*Les états polarisés diagonalement sont aléatoirement absorbés ou transmis*

Pour mieux comprendre ce type de cryptographie, on va l'aborder à partir d'un exemple dont le protocole est bien : le protocole BB84. Avant de commencer, on doit mentionner qu'il existe deux modes de polarisations possibles [www9] :

- **Mode 1 :** "0" est codé par un photon d'axe de polarisation 0° et "1" par un photon de polarisation 90°.
- **Mode 2 :** "0" est codé par un photon d'axe de polarisation 45° et "1" par un photon de polarisation 135°.

Pour plus de clarté, ces deux modes peuvent être schématisés comme suit :

Figure I.14. *Les deux modes de polarisation.*

Cet exemple se déroule en plusieurs étapes que l'on va présenter ci-dessous. On a Alice l'expéditrice, Bob le destinataire et Oscar l'éventuel espion [www9]:

1. Alice génère aléatoirement un bit selon un mode (mode 1 ou mode 2) choisi lui aussi aléatoirement, ce que la physique quantique permet, et transmet le photon obtenu à Bob. Elle répète cette opération autant de fois que nécessaire.
2. Pour chaque photon reçu, Bob choisit aléatoirement un mode, c'est-à-dire un polariseur, et note si le photon traverse le polariseur.
3. Ensuite une phase de réconciliation : Alice et Bob communiquent entre eux et pour chaque photon ils comparent si leurs modes coïncident, si c'est le cas, ils ont le même bit, qu'ils conservent, sinon ils ne conservent pas le bit en question. Vue le nombre de photons inutiles, il est donc nécessaire de prévoir un excédent suffisant de photons pour obtenir une clé de longueur donnée. Comme Bob a une chance sur deux de choisir le bon mode, en moyenne on observe ***N*** erreurs pour ***2N*** photons envoyés.
4. Enfin, Alice et Bob contrôlent la sûreté de la clé : parmi les bits conservés ils en choisissent un certain nombre qu'ils comparent publiquement, s'ils ont été espionnés ils obtiendront en moyenne 25% de bits différents (Comme Oscar ne peut pas cloner les photons, il est dans l'obligation de les intercepter, de les mesurer et de renvoyer à Bob un photon similaire à celui qu'il a mesuré. Cependant il a une chance sur deux de choisir le mauvais mode et donc de renvoyer un photon différent de celui qu'Alice avait envoyé à Bob. Puis comme Bob a lui aussi une chance sur deux d'avoir le bon mode, en cas d'espionnage pour chaque photon on a une chance sur 4 (25%) de détecter l'intrusion). Dans ce cas, ils n'utiliseront pas la clé. S'ils n'ont pas de différences, alors ils peuvent conserver la clé pour l'utiliser ultérieurement.

Alice et Bob décident alors de sacrifier une partie de leur clé commune et les comparent publiquement par le canal radio (comme exemple de canal). Dans le Tableau I.3 ils ont quatre bits différents c.à.d. N bits parmi 2N. Ainsi la clé obtenue : 0011.

Photon envoyé par Alice								
Mode choisi par Bob	Mode 1	Mode 2	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								
Après réconciliation								

Tableau I.3. *Exemple sans Oscar.*

Photon envoyé par Alice								
Mode choisi par Oscar	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 2	Mode 2
Résultat mesuré et renvoyée à Bob par Oscar								
Mode choisi par Bob	Mode 1	Mode 1	Mode 2	Mode 2	Mode 1	Mode 1	Mode 2	Mode 1
Résultat de la mesure de Bob								

Tableau I.4. *Exemple avec Oscar.*

Plusieurs cas de figure se présentent à nous dans le Tableau I.4 [Camp, 2010] :

1. Oscar et Bob ont tous les deux le bon mode, alors Oscar mesure bien la bonne polarisation qu'il renvoie à Bob et puis de même pour Bob. Le bit est valable pour la clé et Oscar est passé inaperçue. Exemple : premier photon envoyé.
2. Oscar a le bon mode, mais pas Bob, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon. Oscar passe encore inaperçue, mais sa bonne mesure lui est inutile. Exemple : le deuxième photon envoyé.
3. Oscar a le mauvais mode, mais Bob a le bon mode, alors d'après ce que l'on a vu, Bob a une chance sur deux d'obtenir une mesure compatible avec ce qu'à envoyer Alice. Donc Oscar a une chance sur deux d'être détecté, de plus sa mesure est inutile. Exemple : sixième photon envoyé (avec détection) et dernier photon envoyé (sans détection).
4. Oscar et Bob ont le mauvais mode, alors lors de la réconciliation Alice et Bob décident de ne pas utiliser le photon, Oscar passe donc inaperçue. Exemple : cinquième photon envoyé.

Une fois la réconciliation terminée il ne reste que les bits de la possibilité 1 et 3 qui sont équiprobables. Pour la première possibilité Oscar passe inaperçue, et pour la troisième il a une chance sur deux de se faire détecter. On retrouve bien le 25% ($\frac{1}{2}\frac{1}{2} = \frac{1}{4}$) vu ci-dessus.

b. Conclusion

Le MIT (Massachussets Institute of Technology) a récemment classé la cryptographie quantique parmi les «10 inventions qui changeront le monde». Si on voit peu les avantages à l'heure actuelle, les possibilités pour le futur sont impressionnantes. Le centre canadien IRCA est un leader mondial en recherche quantique, comptant parmi ses membres des chercheurs montréalais au front de cette révolution du monde des télécommunications. Avec le développement du calcul et de la cryptographie quantique, on peut espérer qu'Alice et Bob pourront un jour communiquer en toute quiétude. Entre temps, considérons d'un œil sceptique les soi-disant algorithmes « incassables » qui n'attendent que le prochain superordinateur pour céder comme une coquille d'œuf [Camp, 2010].

I.3.3.4. Algorithme de chiffrement évolutionniste OTL

Omary Fouzia a développé en 2006 un nouvel algorithme de chiffrement en le simulant à un problème d'optimisation dont la résolution est par les algorithmes génétiques. Le but de cet algorithme est de modifier au maximum les fréquences d'apparition des caractères dans le message à chiffrer et d'établir le plus de désordre dans leurs positions. Ces caractères appartiennent à l'ensemble des 256 caractères du code ASCII. L'application de l'algorithme est précédée d'une phase de brouillage du texte initial (M_0) qui peut combiner plusieurs méthodes simples comme les substitutions, les permutations, le chiffrement affiné, etc..., pour obtenir un texte initialement chiffré (M_0').

Pour ce faire, le codage adopté pour représenter les individus, qui sont les différents messages, est résumé à travers la figure ci-dessous :

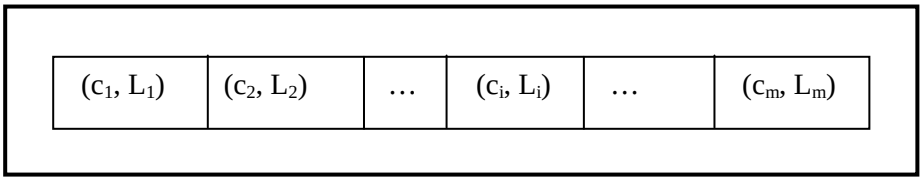


Figure I.15. *Codage des individus.*

Avec :

c_i : un caractère appartenant au message.
L_i : la liste des positions du caractère c_i.
m : le nombre des différents caractères du message.
$L_i \cap L_j = \emptyset, \ \forall i, j \in [1, m]$

L'algorithme OTL cherche à changer itérativement la répartition des listes L_i sur les différents caractères du message (sans modifier le contenu des listes) de telle manière que la différence entre le cardinal de la nouvelle liste affectée à chaque caractère c_i et le cardinal de la liste L_i d'origine soit maximale [Omar, 2006]. Cette variation, en terme de répartition, est assurée grâce aux opérateurs génétiques choisis (MPX pour le croisement avec un taux compris entre 60% et 100%, et une simple permutation pour la mutation avec un taux compris entre 0.1% et 5%). Et pour juger la pertinence des individus construits, la fonction

d'évaluation utilisée est celle donnée ci-dessous. Les meilleurs individus sont ensuite sélectionnés par le biais de la méthode classique de la roulette, en vue de se reproduire.

$$F(X_j) = \sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right|$$

Le processus évolutionnaire est répété jusqu'à la satisfaction d'un critère d'arrêt exprimé à l'aide de la fonction F. Cette dernière est bornée car : $0 \le F(X) \le 2 * l$ (l est la taille du message) pour tout individu X. En fait [Omar, 2006] :

$$\sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right| \le \sum_{i=1}^{m} (card(L_{j_i}) + card(L_i)) \le 2 * l$$

L'opération de déchiffrement, quand à elle, se fait en deux étapes :

1) Tout d'abord, le texte chiffré est représenté suivant le codage proposé en une suite de listes de positions, et c'est grâce à la clé génétique que les caractères vont retrouver leurs listes de position correspondantes dans le message en clair. En effet, la clé, qui peut être d'un usage symétrique ou asymétrique, est une permutation de {1,2,…, m}. Comme résultat, nous obtenons le message M_0'.
2) La deuxième étape, correspond au déchiffrement du message M_0' pour obtenir M_0.

I.4. Conclusion

Dans ce chapitre, nous avons présenté les différentes catégories de cryptographie depuis sa première apparition jusqu'à nos jours. Un état de l'art aussi riche que possible sur les plus célèbres algorithmes de chiffrement, ainsi les fameuses ruses des cryptanalystes ont été exposées. D'après cette étude, un système cryptographique est considéré comme sûr si personne n'a encore mis en défaut sa sécurité. Nous avons vu que ni la cryptographie classique ni symétrique n'a pu s'assurer ce besoin dû aux leurs inconvénients. Pour résoudre cela, les cryptographes ont cherché à déplacer la difficulté ; plutôt que d'utiliser de simples substitutions et de faire reposer la sécurité sur le nombre de clés possibles, ils ont essayé de faire reposer la sécurité sur des difficultés calculatoires. Cela a donné naissance aux algorithmes de cryptographie asymétrique. Cependant il s'avère que l'augmentation constante de la puissance de calcul de nos machines nécessite d'augmenter constamment la difficulté de nos algorithmes. Il se peut en effet qu'une nouvelle technologie anéantisse toute la difficulté d'un algorithme. La cryptographie hybride est aussi un sujet d'attaque puisqu'elle n'est qu'une combinaison de la cryptographie symétrique et asymétrique. Une nouvelle tendance basée sur la cryptographie quantique a aussi été développée. Cette cryptographie est sûre, néanmoins elle est basée sur des principes beaucoup plus théorique et lourds.

Enfin, à partir de la richesse et de la variété des modèles mathématiques exploités dans le domaine de la cryptographie, de nouvelles approches cryptographiques peuvent être proposées par exploitation de méthodes approchées. Le précédent chapitre présentera de telles méthodes de résolution de problèmes : les métaheuristiques.

CHAPITRE II

METAHEURISTIQUES

II.1. Introduction

Un très grand nombre de méthodes existent en RO et en IA pour résoudre différentes sortes de problèmes tels que les problèmes d'optimisation combinatoire. D'une manière très générale, les méthodes de résolution suivent quatre approches différentes pour la recherche d'une solution [JinK, 1999] : l'approche de construction, l'approche de relaxation, l'approche de voisinage et l'approche d'évolution. Ces méthodes peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible les échecs (calculs de bornes) et d'heuristiques spécifiques pour orienter les différents choix. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles (développées depuis une trentaine d'années) telles les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Malgré les progrès réalisés (notamment en matière de la programmation linéaire en nombres entiers), comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante.

Les méthodes approchées constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille si l'optimalité n'est pas primordiale. En effet, ces méthodes sont utilisées depuis longtemps par de nombreux praticiens. On peut citer les méthodes gloutonnes et l'amélioration itérative : par exemple, la méthode de Lin et Kernighan qui resta longtemps le champion des algorithmes pour le problème du voyageur de commerce [LinS, 1973].

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées *métaheuristiques* [Reev, 1993], [Aart, 1997]. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste tabou et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Grâce à ces métaheuristiques, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation classiques de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant [Lapo, 1996], [Osma, 1996]. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle.

Ainsi, les métaheuristiques sont conçues pour résoudre des problèmes d'optimisation complexes où d'autres méthodes d'optimisation ne sont pas efficaces. Ces méthodes ont fini par être reconnues comme l'une des approches les plus pratiques pour résoudre des problèmes nombreux et complexes, et ceci est particulièrement vrai pour les divers problèmes du monde réel qui sont de nature combinatoire. L'avantage pratique de métaheuristiques réside dans leur efficacité et leur application générale. Dans la littérature et au début des recherches, des heuristiques spécialisées ont été généralement développées pour résoudre des problèmes complexes d'optimisation combinatoire.
D'autre part, avec l'émergence de stratégies des solutions plus générales, y compris les métaheuristiques telles que la recherche taboue, algorithmes génétiques, recuit simulé, le principal défi est devenu l'adaptation des méta-heuristiques à un problème particulier ou une catégorie de problème. Cela nécessite généralement beaucoup moins de travail que de développer une heuristique spécialisée pour une application spécifique, ce qui rend les métaheuristiques un choix attrayant pour la mise en œuvre dans les logiciels à usage général. En outre, une bonne mise en œuvre de métaheuristique est susceptible de fournir des solutions quasi optimales en temps de calcul raisonnables.

Ainsi, une métaheuristique est définie de manière similaire à une heuristique (qui est une méthode conçue pour un problème d'optimisation donné et qui produit une solution non nécessairement optimale lorsqu'on lui fournit une instance de ce problème), mais à un niveau d'abstraction plus élevé (d'après E. Taillard).

II.2. Principes

Les métaheuristiques sont apparues dans les années 80. Ce sont des méthodes d'optimisation, de type stochastique, conçus pour les problèmes difficiles. Des progrès importants ont été réalisés avec l'apparition de nouvelles générations de méthodes approchées puissantes et générales, souvent appelées métaheuristiques [Reev, 1993] [Aart, 1997].

Le terme « métaheuristique » a été initialement utilisé par F. Glover pour distinguer la méthode tabou des heuristiques spécifiques [Glov, 1986]. Notons que ce terme est également utilisé par J-L. Laurière dans son système de résolution Alice [Laur, 1978].

L'origine du mot méta heuristique nous aide à comprendre sa signification. En grec: « Méta » signifie « au-delà », ou « au niveau supérieur », et « Heuristique» veut dire « Trouver ». Ainsi, l'interprétation de ces mots peut signifier que l'on effectue des recherches à un très haut niveau et que la procédure algorithmique regroupe plusieurs heuristiques.

D'après Glover, metaheuristique "se réfère à une stratégie maître qui guide et modifie d'autres heuristiques pour produire des solutions au-delà de ceux qui sont normalement générés dans une quête d'optimalité locale" [Glov, 1997].

Selon [Glov, 1997] « métaheuristiques dans leurs formes modernes sont basées sur une variété d'interprétations de ce qui constitue une recherche intelligente », où le terme de recherche intelligente a été mis en évidence par Pearl [Poire, 1984] (en ce qui concerne heuristiques dans un contexte de l'intelligence artificielle) et [vobs, 1993] (en ce qui concerne le contexte de la recherche opérationnelle). Dans ce sens, on peut aussi considérer la définition suivante: « Une métaheuristique est un processus à générations itératives qui guide

une heuristique subordonnée en combinant intelligemment différents concepts pour explorer et exploiter les espaces de recherche en utilisant des stratégies lesarning pour structurer l'information afin de trouver des solutions quasi-optimales » [Osma, 1996]

Pour résumer, la définition suivante semble être la plus appropriée : « Une métaheuristique est un processus maître itératif qui guide et modifie les opérations d'heuristiques subordonnées pour produire efficacement des solutions de haute qualité. Il peut manipuler une solution unique complète (ou incomplète) ou un ensemble de solutions à chaque itération. Les heuristiques subordonnées peuvent être des procédures de niveau élevé (ou faible), ou une recherche locale simple, ou tout simplement une méthode de construction. La famille de métaheuristiques comprend, mais n'est pas limitée à, des procédures de mémoire d'adaptation, de recherche tabou, des systèmes de fourmis, avides de recherche randomisée d'adaptation, de recherche à voisinage variable, des méthodes évolutionnaires, des algorithmes génétiques, de recherche de points, des réseaux neuronaux, de recuit simulé, et leurs hybrides. " [VobS, 1999]

Le but des métaheuristiques est de minimiser ou de maximiser une, ou plusieurs fonctions objectives. Comme exemple, on peut demander de trouver le plus court chemin entre un point A et un point B, sachant que des obstacles sont présents donc il s'agit d'un problème de minimisation ; ou si on veut trouver comment effectuer le plus grand nombre de tâches en un temps limité alors c'est la maximisation du travail à faire.

L'évolution des métaheuristiques se fait de manière itérative. Ceci a l'avantage de permettre l'arrêt de l'algorithme quand on le souhaite, et de récupérer la meilleure solution trouvée jusqu'à présent sans être obligé d'attendre la fin de l'exécution. Un autre point important des métaheuristiques est qu'elles font évoluer des solutions en les améliorant à chaque itération.

Donc, les métaheuristiques sont généralement des algorithmes itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction. Les itérations successives permettent d'évoluer d'une solution peu satisfaisante à la solution la plus adaptée. Les métaheuristiques se comportent, donc, comme des algorithmes de recherche tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution.

Les métaheuristiques sont souvent inspirées de systèmes naturels, qu'ils soient pris en physique -cas du recuit simulé-, en biologie de l'évolution -cas des algorithmes génétiques- ou encore en éthologie -cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires.
Les métaheuristiques désignées par M sont, souvent, des algorithmes utilisant un échantillonnage probabiliste. Elles tentent de trouver l'optimum global (G) d'un problème d'optimisation difficile (avec des discontinuités (D), par exemple), sans être piégées par les optima locaux (L). Ceci est indiqué à la figure suivante :

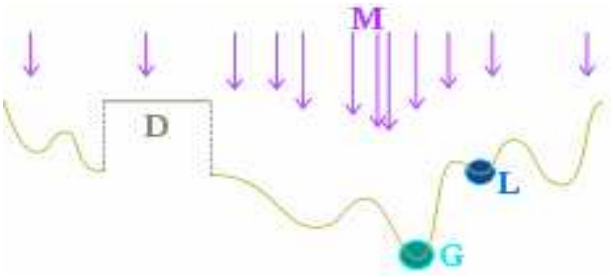


Figure II.1. *Principe général des métaheuristiques.*

II.3. Organisation d'une métaheuristique

D'une manière générale, les métaheuristiques s'articulent autour des principales notions suivantes :

II.3.1. Le Voisinage

Le voisinage d'une solution est un sous-ensemble de solutions qu'il est possible d'atteindre par une série de transformations données.
Formellement parlant [JinK, 1999], soit X l'ensemble des configurations admissibles d'un problème, on appelle *voisinage* toute application $N : X \to 2^X$. On appelle *mécanisme d'exploration* du voisinage toute procédure qui précise comment la recherche passe d'une configuration $s \in X$ à une configuration $s' \in N(s)$. Une configuration *s* est un *optimum local* par rapport au voisinage *N* si $f(s) \leq f(s')$ pour toute configuration $s' \in N(s)$.

Une méthode typique de voisinage débute avec une configuration initiale, et réalise ensuite un processus itératif qui consiste à remplacer la configuration courante par l'un de ses voisins en tenant compte de la fonction de coût. Ce processus s'arrête et retourne la meilleure configuration trouvée quand la condition d'arrêt est réalisée.

Un des avantages de cette stratégie de recherche réside précisément dans la possibilité de contrôler le temps de calcul : la qualité de la solution trouvée tend à s'améliorer progressivement au cours du temps et l'utilisateur est libre d'arrêter l'exécution au moment qu'il aura choisi. Dans la littérature, plusieurs voisinages ainsi que plusieurs stratégies de parcours de ces voisinages ont été définies [Gold, 1989], [Hert, 2003].

II.3.2. Diversification, intensification et apprentissage

La *diversification* (ou *exploration*, synonyme utilisé presque indifféremment dans la littérature des algorithmes évolutionnaires) désigne les processus visant à récolter de l'information sur le problème optimisé. L'*intensification* (ou *exploitation*) vise à utiliser l'information déjà récoltée pour définir et parcourir les zones intéressantes de l'espace de recherche. Autrement dit, l'intensification insiste sur la capacité d'examiner en profondeur par une méthode des zones de recherche particulières alors que la diversification met en avant la capacité de découvrir des zones de recherche prometteuses.

La *mémoire* est le support de l'apprentissage, qui permet à l'algorithme de ne tenir compte que des zones où l'optimum global est susceptible de se trouver, évitant ainsi les optima locaux.

Les métaheuristiques progressent de façon itérative, en alternant des phases d'intensification, de diversification et d'apprentissage, ou en mêlant ces notions de façon plus

étroites. L'état de départ est souvent choisi aléatoirement, l'algorithme se déroulant ensuite jusqu'à ce qu'un critère d'arrêt soit atteint.

Les notions d'intensification et de diversifications sont prépondérantes dans la conception des métaheuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherches. En effet, l'application systématique du seul principe d'exploitation ne permet pas une recherche efficace. En effet, l'exploitation conduit à confiner la recherche dans une zone limitée qui finit par s'épuiser. Le cas de l'amélioration itérative rapidement piégée dans un optimum local illustre cruellement ce phénomène. Une autre illustration souvent évoquée est fournie par le problème de convergence prématurée des algorithmes génétiques : du fait de la sélection, la population finit par n'être constituée que d'individus tous similaires. L'une des préoccupations majeures dans les algorithmes génétiques consiste d'ailleurs à préserver le plus longtemps possible une diversité suffisante dans la population. Face à ce type de difficulté, la solution consiste à diriger la poursuite de la recherche vers de nouvelles zones, *i.e.*, à recourir à l'exploration. Les deux notions ne sont, donc, pas contradictoires mais complémentaires, et il existe de nombreuses stratégies mêlant à la fois l'un et l'autre des aspects.

II.4. Méthodes générales et méthodes spécifiques

Contrairement aux algorithmes traditionnels dédiés à un problème spécifique, les métaheuristiques constituent des mécanismes très généraux qui peuvent être adaptés pour traiter de nombreux problèmes différents. Pour expliquer l'efficacité d'un algorithme spécifique, on invoque souvent le fait qu'il utilise des connaissances spécifiques du problème. Pour expliquer l'efficacité d'une métaheuristique, deux types d'arguments opposés sont généralement avancés.

Selon certains, des mécanismes généraux suffisamment puissants ont par eux mêmes la faculté de mener efficacement la recherche sans disposer d'information spécifique du problème considéré (contexte de boîte noire) : c'était notamment le point de vue classique porté sur les algorithmes génétiques [Gold, 1989]. Malheureusement, de même qu'il ne peut pas exister de stratégie avantageuse dans un jeu de hasard comme la roulette, il existe également des limitations théoriques fondamentales qui ruinent les espoirs d'une méthode aveugle dans le cas le plus général. En fait, le théorème « *No Free Lunch (NFL)* » [Wolp, 1997] montre que pour les problèmes de type boîte noire, toutes les méthodes sont équivalentes et font aussi bien, ou plutôt aussi mal, que l'énumération aléatoire.

Selon le point de vue opposé, la puissance d'une métaheuristique est d'abord liée à son aptitude à intégrer des connaissances spécifiques du problème. La connaissance du problème la plus fondamentale réside dans le codage du problème et dans le choix de la fonction de voisinage. Plusieurs auteurs insistent sur l'importance du codage du problème, voir par exemple [Radc, 1995]. Dans le cas du voyageur de commerce, plusieurs types de codages différents ont été proposés et conduisent à des performances très variées [Mich, 1992]. En général, il n'existe pas de codage universellement efficace. Un « bon » codage doit permettre de restreindre l'espace de recherche et d'intégrer des connaissances du problème.

Les métaheuristiques tentent également d'améliorer leur efficacité en incorporant des connaissances supplémentaires dans leurs opérateurs. Plus les opérateurs d'une méthode utilisent des connaissances spécifiques, plus la méthode dispose de moyens potentiels pour conduire efficacement la recherche. En contrepartie, l'intégration de ces connaissances spécifiques (en supposant que ces connaissances soient disponibles) nécessite un effort pour spécialiser ou adapter la méthode. La méthode tabou vise à incorporer le plus possible de connaissances du problème pour atteindre le maximum d'efficacité : l'utilisateur doit notamment définir de façon pertinente le type de caractéristique figurant dans la liste tabou.

De leur coté, les algorithmes génétiques se sont éloignés du modèle standard pour intégrer également des connaissances du problème : codage non binaire, opérateurs spécifiques. Au contraire, le recuit simulé est parfois présenté comme une méthode facile à adapter à un problème et qui ne tente pas d'exploiter de connaissances spécifiques.

En général, une méthode offrant des possibilités d'intégrer des connaissances du problème a plus de chance de produire de bons résultats, mais demande un effort d'adaptation et de spécialisation. Au contraire, une méthode très générale qui prétend n'intégrer aucune connaissance propre ne peut pas être compétitive.

II.5. Classification des métaheuristiques

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents.
Dans la littérature, plusieurs classifications de métaheuristiques ont été proposées. Nous présentons à ce niveau un aperçu des principales classifications.

II.5.1. Les métaheuristiques inspirées et non inspirées d'un phénomène naturel

Une manière intuitive de classifier les métaheuristiques consiste à séparer celles qui sont inspirées d'un phénomène naturel, de celles qui ne le sont pas.

Les algorithmes génétiques ou les algorithmes par colonies de fourmi entrent clairement dans la première catégorie, tandis que la méthode de descente, ou la recherche Tabou, vont dans la seconde.

II.5.2 Les métaheuristiques avec fonction objective statique ou dynamique

Les métaheuristiques peuvent être aussi classées selon la façon dont ils utilisent la fonction objective. Certains algorithmes conservent la fonction objective donnée dans la représentation de problème telle qu'elle est (comme la recherche locale guidée (GLS)) et la modifie lors de la recherche. L'idée de cette approche est d'éviter les minimums locaux en modifiant l'espace de recherche. En conséquence, lors de la recherche la fonction objective est altérée en essayant d'intégrer des informations intégrées au cours du processus de recherche.

II.5.3. Les métaheuristiques avec une ou plusieurs structures de voisinage

La plupart des métaheuristiques travaillent sur une seule structure de voisinage. En d'autres termes, la topologie du paysage de fitness ne change pas en cours de l'algorithme. D'autres métaheuristiques telles que la recherche par voisinage variable utilisent un ensemble de structures de voisinage qui donne la possibilité de diversifier la recherche en échangeant entre les structures de voisinage qui ont des formes différentes.

II.5.4. Les métaheuristiques avec et sans mémoire

Les métaheuristiques utilisent l'historique de leur recherche pour guider l'optimisation aux itérations suivantes. Dans le cas le plus simple, elles se limitent à considérer l'état de la recherche à une itération donnée pour déterminer la prochaine itération : il s'agit alors d'un processus de décision markovien, et on parlera de méthode *sans mémoire*. C'est le cas de la plupart des méthodes de recherche locale (recherche à voisinage variable, recherche locale

itérée, recherche locale stochastique, recherche locale guidée), algorithme d'estimation de distribution, recuit simulé, GRASP.

Beaucoup de métaheuristiques utilisent une mémoire plus évoluée, que ce soit sur le court terme (solutions visitées récemment, par exemple) ou sur le long terme (mémorisation d'un ensemble de paramètres synthétiques décrivant la recherche).

II.5.5. Les métaheuristiques implicite, explicite, directe

En considérant les métaheuristiques comme des méthodes itératives utilisant un échantillonnage de la fonction objective comme base d'apprentissage (définition plus particulièrement adaptée aux métaheuristiques à populations) apparaît le problème du choix de l'échantillonnage.

Dans la très grande majorité des cas, cet échantillonnage se fait sur une base aléatoire, et peut donc être décrit via une distribution de probabilités. Il existe alors trois classes de métaheuristiques, selon l'approche utilisée pour manipuler cette distribution.

La première classe est celle des méthodes *implicites*, où la distribution de probabilité n'est pas connue *a priori*. C'est le cas par exemple des algorithmes génétiques, où le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. L'évolution différentielle, Scatter search, algorithmes à essaim de particules, stratégies d'évolution et algorithmes de colonie de fourmis sont des méthodes implicites.

Par opposition, on peut donc classer les méthodes *explicites*, qui utilisent une distribution de probabilité choisie à chaque itération. C'est le cas des algorithmes à estimation de distribution.

Dans cette classification, le recuit simulé occupe une place particulière, puisqu'on peut considérer qu'il échantillonne la fonction objective en utilisant directement celle-ci comme distribution de probabilité (les meilleures solutions ayant une probabilité plus grande d'être tirées). Il n'est donc ni explicite ni implicite, mais plutôt « direct » (exemple : le recuit simulé) [JinK, 1999].

II.5.6. Les métaheuristiques évolutionnaires et non évolutionnaires

On trouve parfois une classification présentant les algorithmes d'optimisations stochastiques comme étant « évolutionnaires » (ou « évolutionnistes ») ou non. L'algorithme sera considéré comme faisant partie de la classe des algorithmes évolutionnaires s'il manipule une population *via* des *opérateurs*, selon un algorithme général donné.

Cette façon de présenter les métaheuristiques dispose d'une nomenclature adaptée : on parlera d'opérateurs pour toute action modifiant l'état d'une ou plusieurs solutions. Un opérateur construisant une nouvelle solution sera dénommé *générateur*, alors qu'un opérateur modifiant une solution existante sera appelé *mutateur*.

Dans cette optique, la structure générale des algorithmes évolutionnaires enchaîne des étapes de *sélection*, de *reproduction* (ou *croisement*), de *mutation* et enfin de *remplacement*. Chaque étape utilise des opérateurs plus ou moins spécifiques.

Quelques algorithmes évolutionnaires : l'algorithme génétique, la programmation génétique, l'algorithme d'évolution différentielle, les stratégies évolutionnaires et l'algorithme d'estimation de distribution [JinK, 1999].

II.5.7. Les métaheuristiques à base de population et les métaheuristiques à trajectoire

Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de voisinage est alors primordiale.

Les plus connues dans cette classe sont le recuit simulé, la recherche tabou, la recherche à voisinage variable, la méthode GRASP. L'autre approche utilise la notion de population. La métaheuristique manipule un ensemble de solutions en parallèle, à chaque itération. On peut citer les algorithmes génétiques, l'optimisation par essaims particulaires et les algorithmes de colonies de fourmis.

La frontière est parfois floue entre ces deux classes. On peut ainsi considérer qu'un recuit simulé où la température baisse par paliers, a une structure à population. En effet, dans ce cas on manipule un ensemble de points à chaque palier, il s'agit simplement d'une méthode d'échantillonnage particulière.

II.5.7.1. Les métaheuristiques à trajectoire (à solution unique)

Ici, nous résumons les plus connues des métaheuristiques à trajectoire.

a. La méthode de descente

Le principe de la méthode de descente (dite aussi *basic local search*) consiste, à partir d'une solution *S*, à choisir une solution *S'* dans un voisinage de *S* telle que *S' améliore* la recherche (généralement telle que $f(S') < f(S)$). On peut décider soit d'examiner toutes les solutions du voisinage *N* et prendre la meilleure de toutes ces solutions (ou prendre la première trouvée), soit d'examiner un sous-ensemble du voisinage.
La méthode de descente est la méthode la plus élémentaire de recherche locale. On peut la formaliser comme suit :

Algorithme II.1 *descente_simple* (solution initiale *S*)

Début
  Répéter
    Choisir *S'* dans N(*S*)
    Si $f(S') < f(S)$ alors $S \leftarrow S'$
  Jusqu'à ce que $f(S') \geq f(S)$, $\forall S' \in$ **S**
Fin

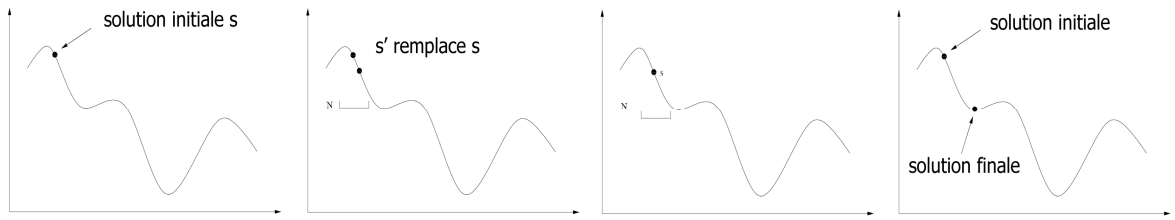


Figure II.2. *Evolution d'une solution dans la méthode de descente.*

On peut varier cette méthode en choisissant à chaque fois la solution S' dans N(S) qui améliore le plus la valeur de f. C'est la méthode de plus grande descente [Cost, 2006].

b. Recherche aléatoire (Random Search : RS)

C'est la plus simple des méthodes stochastiques. Cette méthode consiste à tirer à chaque itération une solution au hasard. La fonction objective *f* est évaluée en ce point. La nouvelle valeur est comparée à la précédente. Si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continu. Sinon on repart du point précédent et on recommence le procédé, jusqu'à ce que les conditions d'arrêt soient atteintes. L'algorithme II.2 [Luke, 2010] présente un pseudo code de la recherche aléatoire dans le cas d'un problème de minimisation.

Algorithme II.2 *Recherche aléatoire*

Début
$S_0 \leftarrow$ solution aléatoire;
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$;
Répéter
S ← solution aléatoire ;
Si ($f(S) < f_{min}$) *Alors*
$f_{min} \leftarrow f(S)$;
$x_{min} \leftarrow S$;
Fin Si
Jusqu'à conditions d'arrêt satisfaites.
Fin

c. Hill Climbing (HC)

On part d'une solution si possible "bonne" (par exemple donnée par une heuristique) et on balaie l'ensemble des voisins de cette solution ; s'il n'existe pas de voisin meilleur que notre solution, cela veut dire qu'un optimum local a été trouvé et on arrête la recherche. Sinon, on choisit le meilleur des voisins et on recommence. Une autre implémentation consiste non pas à passer au meilleur des voisins à chaque étape, mais au premier meilleur voisin trouvé. La convergence vers un optimum local pouvant être très lente, on peut éventuellement fixer un nombre de boucles maximum, si on veut limiter le temps d'exécution [Sea, 2010].

Cette méthode a l'inconvénient de rester bloquée dans un optimum local : une fois un optimum local trouvé, on s'arrête, même si ce n'est pas l'optimum global. Selon le paysage des solutions, l'optimum local peut être très bon ou très mauvais par rapport à l'optimum global. Si la solution de départ est donnée par une heuristique déterministe, l'algorithme sera déterministe. Si elle est tirée au hasard, l'algorithme devient non déterministe et donc plusieurs exécutions différentes sur la même instance pourront donner des solutions différentes et de qualités différentes [Sea, 2010].

La notion de voisinage est primordiale. Si les voisins sont très nombreux, on a de fortes chances de trouver l'optimum global, mais visiter un voisinage peut être très long: on visitera une grande partie de l'espace des solutions. Si le voisinage est très restreint, on risque fort de rester bloqué dans un optimum local de "mauvaise qualité". Le choix de la notion de voisinage est un compromis entre efficacité et qualité.

Le pseudo-code de l'algorithme Hill Climbing est celui présenté par l'algorithme II.3 [Sea, 2010]:

Algorithme II.3 *HILL CLIMBING*

Début
$S_0 \leftarrow$ Solution aléatoire
$f_{min} \leftarrow f(S_0)$
$x_{min} \leftarrow S_0$
Répéter
Engendrer un N-échantillon $S_i(x)$ voisinage de S(x) *ET* calculer
$f(S(x)) = \min[f(S_i(x))]$ avec $1 \leq i \leq N$

Si ($f(S) < f_{min}$) *then*
$x_{min} \leftarrow S$
Sinon
Sortir de Répéter
Fin Si
Jusqu'à condition d'arrêt satisfaite.
Fin.

d. Recuit Simulé

Le recuit simulé a été introduit et mis au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985. Il a été repris sous différentes formes par Lawrence Davis en 1987 [Lawr, 1987].
Cet algorithme a été dérivé de l'algorithme de Metropolis, développé par les scientifiques du projet ex-Manhattan : Nicholas Metropolis, Arianna et Marshall Rosenbluth, Augusta et Edward Teller en 1953. Ce dernier permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. Partant d'une solution donnée, en la modifiant, on en obtient une seconde. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une « mauvaise » solution permet alors d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

La description des phénomènes physiques et quantiques liés au processus de recuit s'appuie sur la statistique de Boltzmann. Pour qu'un métal ait une qualité optimale, il faut que son état d'énergie soit minimal. Pour améliorer la qualité d'un métal, on le chauffe, puis on le refroidit par paliers en attendant à chaque fois que l'état d'énergie soit stabilisé. Au niveau de l'atome, cela signifie qu'à une température chaude, les atomes bougent beaucoup, et en se

refroidissant, ils trouvent leur place optimale. Plus la température descend, moins les atomes bougent et le métal devient de plus en plus résistant.

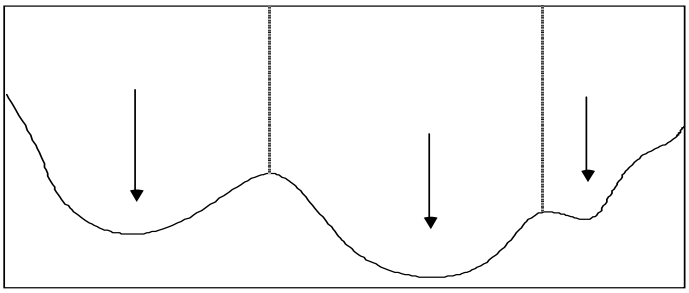


Figure II.3. *Un paysage d'énergie. Suivant la configuration initiale, indiquée par les flèches, la dynamique aboutit à température nulle dans l'un quelconque des minima relatifs, séparés par les barrières indiquées en pointillés. A température élevée, les processus probabilistes permettent au système de sauter les barrières séparant les vallées.*

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. À cette solution correspond une énergie initiale $E = E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T = T_0$ élevée est également choisie. Ce choix est alors totalement arbitraire et va dépendre de la loi de décroissance utilisée.

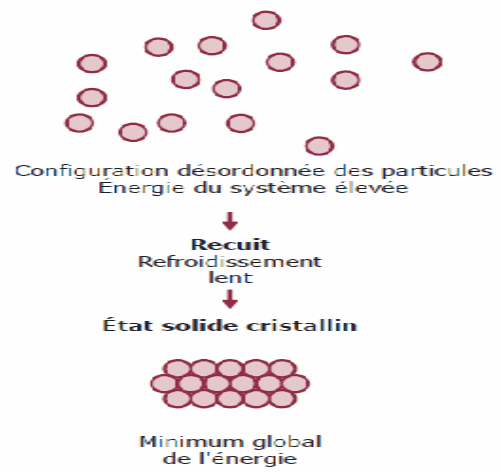


Figure II.4. *Transposition de procédé recuit à la résolution d'un problème d'optimisation.*

À chaque itération de l'algorithme une modification élémentaire de la solution est effectuée. Cette modification entraîne une variation Δ_E de l'énergie du système (toujours calculée à partir du critère que l'on cherche à optimiser). Si cette variation est négative (c'est-à-dire qu'elle fait baisser l'énergie du système), elle est appliquée à la solution courante. Sinon, elle est acceptée avec une probabilité $e^{-\frac{\Delta_E}{T}}$. Ce choix de l'exponentielle pour la probabilité s'appelle règle de Metropolis.

On itère ensuite selon ce procédé en gardant la température constante.

Algorithme II.4 *Recuit simulé*

```
Début
    Initialisation du temps
    Tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint] Faire
        Initialisation de la température à Tmax
        Tant que [la température est supérieure à 0] et [le temps limite n'est pas atteint] Faire
            Incrémentation du temps
            Modification de la solution courante
            Si [la nouvelle solution est meilleure que l'ancienne] Alors
                la sauvegarder
            Sinon
                Si [le système est stable depuis un nombre défini d'itérations] Alors
                    Diminuer la température
                Fin si
            Fin si
        Fin tant que
    Fin tant que
Fin
```

L'algorithme varie de Hill-Climbing (algorithme 3) dans sa décision du moment de remplacer S, la solution candidate d'origine, avec R, son enfant nouveau peaufiné. Plus précisément, si R est meilleur que S, nous allons toujours remplacer S par R comme d'habitude. Mais si R est pire que S, on peut toujours remplacer S par R avec une certaine probabilité P(t, R, S):

$$P(t, R, S) = e^{\frac{\text{Quality}(R) - \text{Quality}(S)}{t}}$$

Où t> 0. Cette équation est intéressante à deux égards. Notez que la fraction est négative car R est pire que S. Premièrement, si R est bien pire que S, la fraction est plus grande, et donc la probabilité est proche de 0. Si R est très proche de S, la probabilité est proche de 1. Ainsi, si R n'est pas bien pire que S, nous allons toujours choisir R avec une probabilité raisonnable.

Deuxièmement, nous avons un paramètre ajustable t. Si t est proche de 0, la fraction est de nouveau un grand nombre, et donc la probabilité est proche de 0. Si t est élevé, la probabilité est proche de 1. L'idée est d'abord de mettre t comme un grand nombre, ce qui provoque l'algorithme de se déplacer à chaque solution nouvellement créée indépendamment de sa qualité. Nous faisons une marche aléatoire dans l'espace. Puis t diminue lentement, éventuellement à 0, à quel point l'algorithme ne fait rien de plus que de simples Hill-Climbing.

e. Recherche Tabou (Tabu Search : TS)

La recherche avec tabou, ou simplement dite recherche tabou, est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans un article paru en 1986 [Glov, 1986], mais reprenant de nombreuses idées proposées antérieurement dès les années 60 dans les deux articles simplement intitulés « Tabu chearch » [Glov, 1989] proposant la plus part des principes de cette recherche telle qu'elle est décrite actuellement. Maintenant, elle est d'usage très classique en domaine d'optimisation combinatoire pour résoudre les problèmes NP-durs.

L'idée principale de la recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui optimise la fonction objective. Le voisinage choisi est exploré de manière déterministe, il est interdit de reprendre des solutions récemment visitées.

L'un des principes fondamentaux de cette recherche qui la caractérise des autres méta-heuristiques est la construction d'un historique de la recherche itérative ou, ce qui est équivalent, au fait de doter la recherche de mémoire [Glov, 1997]. La recherche tabou examine un échantillonnage de solutions *N(S)* et retient la meilleure solution *S* des solutions obtenues.

A chaque itération, l'algorithme tabou choisit le meilleur voisin non tabou, même si celui-ci dégrade la fonction de coût. Pour cette raison, on dit de la recherche tabou qu'elle est une méthode agressive [www10].

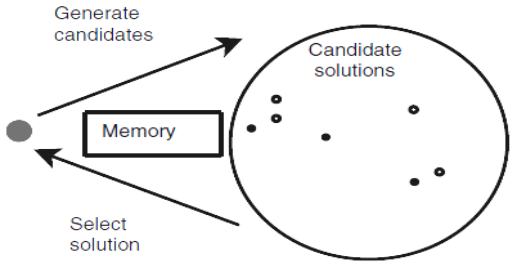
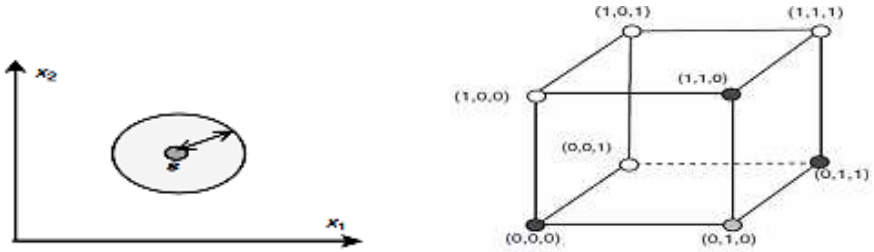


Figure II.5. *Principe de base d'une méta-heuristique à mémoire.*

La recherche tabou est essentiellement axée sur une exploration non triviale de l'ensemble des solutions en utilisant la notion de voisinage. Formellement pour toute solution *s* de *S*, un ensemble de $N(s) \in S$ qu'on appelle ensemble des solutions voisines de *s*.



(a) *(b)*

Figure II.6. *Les types de solutions du voisinage.*
(a) *Le voisinage pour un problème à variables continus : Le cercle symbolise les voisins de la solution s,*
(b) *Le voisinage pour un problème à variables discrètes : Les nœuds du cube sont les solutions et leurs voisins adjacents.*

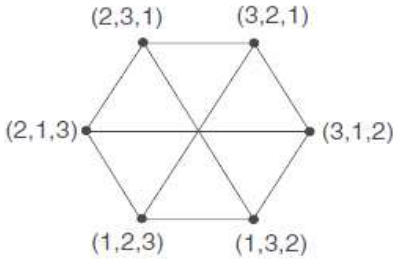


Figure II.7. *Voisinage d'une permutation de taille égale à 3.*
Les voisins de la solution (2.3.1) sont : (3.2.1), (2.1.3), et (1.3.2).

Vue d'un côté pratique l'ensemble N(s) des solutions voisines de s n'est autre que l'ensemble des modifications que l'on peut apporter à cette dernière. On appelle *mouvement* une modification apportée à une solution.

Dans la littérature et dans le cas de la recherche tabou, l'application d'un mouvement *m* à une solution *s* est notée ; d'où l'expression suivante du voisinage :

$$N(s) = \{s' / s' = s \oplus m, m \in M\}$$

La figure II.8 [Kamm, 2006] illustre l'ensemble des mouvements possibles dans le cas d'une solution composée de quatre éléments.

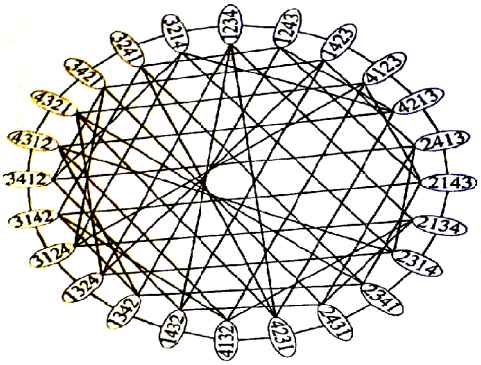


Figure II.8. *Illustration de l'ensemble des mouvements possibles d'une solution de 4 éléments.*

La recherche tabou est une méthode itérative qui explore un espace de solutions. Comme toute méthode itérative, il est intéressant d'éviter de revisiter des solutions pour essayer de converger vers la solution optimale le plus rapidement possible et ceci n'est possible qu'à l'aide d'une mémoire caractérisée par une taille définie par l'utilisateur et le problème à résoudre. Cependant, l'utilisation d'une mémoire peut s'avérer dans bien des cas peu efficace, voire mauvaise. Outre le fait de mettre cette idée en pratique est difficile, cela suppose que l'on doit mémoriser chaque solution visitée et de tester à chaque itération et pour chaque solution admissible si cette dernière a été déjà examinée. Les *tables de hachage* permettent de faire cela de manière efficace, mais ceci n'empêche pas la croissance linéaire de la taille de la mémoire avec le nombre des itérations effectuées.

Une table de hachage est définie par un tableau *T* de *J* entiers, avec *J* choisi relativement grand et assez voisin de la capacité de la machine utilisée tout en prenant en considération la taille du problème.

Le but de la table de hachage est :

- ✓ d'interdire le retour aux solutions obtenues pendant les *t* dernières opérations ;
- ✓ ainsi éliminer les cycles de longueurs égales ou inférieures à *t*.

Toutefois, le fait d'interdire un grand nombre de mouvements aura pour conséquence de rendre l'obtention de bons résultats très délicate faute de mouvements disponibles. Si ce nombre diminue, cela augmentera les chances d'une meilleure exploration aux alentours des optimums locaux et d'obtenir ainsi de meilleures solutions. Il ne faut cependant pas trop diminuer ce nombre, car, dans ce cas, il devient très probable de se trouver prisonnier d'un ensemble très restreint de solutions et de les visiter tout le temps de la recherche.

Pour palier à ce compromis du choix de nombre des mouvements à interdire et de bénéficier simultanément des avantages d'un petit nombre, qui permet une visite détaillée du voisinage d'une même solution, et d'un grand nombre qui permet de franchir le voisinage de différentes solutions, ce nombre doit varier au cours du processus itératif. Pour ce faire, plusieurs méthodes existent. Ce nombre peut être tiré au hasard à partir d'un intervalle donné à chaque itération ou après un nombre d'itérations, comme il peut aussi croître ou décroître en fonction des résultats obtenus au cours de la recherche [Tail, 1991], [Tail, 1995] et [Tail, 1999].

Autrement dit, une liste tabou avec trop d'éléments peut devenir très restrictive. En effet, il a été observé que trop de contraintes tabou forcent le programme à visiter des solutions voisines peu alléchantes à la prochaine itération. Par contre une liste tabou contenant trop peu d'éléments peu s'avérer inutile et mener à des mouvements cycliques [Ayas, 2004].

En plus, dans beaucoup de problèmes, l'interdiction de revisiter des solutions mènerait à des incohérences comme la déconnection de la solution courante de la solution optimale ou bien le blocage de la recherche itérative par cause d'absence de solutions voisines non visitées[Kamm, 2006].

La figure suivante illustre ces derniers cas incohérents :

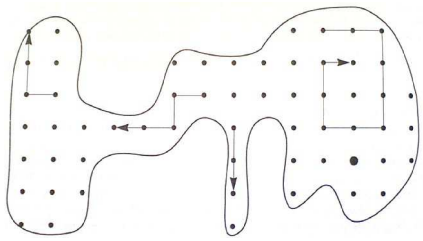


Figure II.9. *Déconnections et blocages dans la recherche tabou.*

Il est à signaler qu'une liste tabou peut être soit statique ou dynamique. Pour la première, la taille de la liste est fixée, en générale au début. Elle dépend de la taille du problème à résoudre et du voisinage choisi. Pour la deuxième, la taille doit changer durant le processus de recherche sans utiliser aucune information sur la mémoire de recherche.

Algorithme II.5 *Recherche Tabou*

Début

Φ Fonction de coût
Variables locales : S solution courante, Liste Taboue L, Meilleure solution M, itération courante K, nombre d'itérations N
Paramétrages : taille de liste taboue, critère d'aspiration
Choix d'une solution initiale S_0
S← S_0
M← S
K ←0
Tant que K< N *faire*
K← K+1
Mise à jour de L
Génération des candidats E par opération de voisinage
C ← Best (E)
Si Φ (S) < Φ(M) *ou* C n'est pas taboue *ou* C vérifie l'aspiration *alors*
S ← C
Sinon
E ← E \ C
Fin tant que
Retourner S

Fin.

f. GRASP

À proprement parler, GRASP (pour *Greedy Randomized Adaptive Search Procedure*) est une méthode hybride, car elle cherche à combiner les avantages des heuristiques gloutonnes, de la recherche aléatoire et des méthodes de voisinage. Un algorithme GRASP répète un processus composé de deux étapes : la construction d'une solution suivie par une descente pour améliorer la solution construite. Durant l'étape de construction, une solution est itérativement construite : chaque itération ajoute un élément dans la solution partielle courante. Pour déterminer l'élément qui sera ajouté, on utilise une liste des meilleurs candidats obtenus avec une fonction gloutonne et on prend *au hasard* un élément dans cette liste. La liste des meilleurs candidats est dynamiquement mise à jour après chaque itération de construction. Cette étape de construction continue jusqu'à ce qu'une solution complète soit obtenue [JinK, 1999].

À partir de cette solution, une descente est appliquée pour améliorer la solution. Une procédure GRASP répète ces deux étapes et retourne à la fin la meilleure solution trouvée. Les deux paramètres de cette méthode sont donc la longueur de la liste de candidats et le nombre d'itérations autorisées.

Algorithme II.6 *GRASP*

```
Début
    x, x*, f* : type Données
    Début
    f* ← infini
    Pour i = 1 à nombre d'itérations faire
        x ← Construction Aléatoire Gloutonne ()
        x ← Recherche Locale(x)
        Si f(x) < f* alors
            x* ← x
            f* ←f(x)
        Fin Si
  Fin Pour
Fin.
```

II.5.7.2. Les métaheuristiques à population

Dans cette section, nous présentons un sommaire des plus connues des métaheuristiques à population.

a. Les algorithmes évolutionnaires

Les *algorithmes évolutionnistes* ou *algorithmes évolutionnaires* (*evolutionary algorithms* ou encore dits *evolutionary computation* en anglais), sont une famille d'algorithmes s'inspirant de la théorie de l'évolution pour résoudre des problèmes divers. Ils font ainsi évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes stochastiques, car ils utilisent itérativement des processus aléatoires.

La grande majorité de ces méthodes sont utilisées pour résoudre des problèmes d'optimisation, elles sont en cela des métaheuristiques, bien que le cadre général ne soit pas nécessairement dédié aux algorithmes d'optimisation au sens strict. On les classe également parmi les méthodes d'intelligence calculatoire.

Selon la génétique et la théorie de l'évolution [www11] :

- ✓ Un enfant hérite son patrimoine génétique pour moitié de sa mère et pour moitié de son père (reproduction sexuée).
- ✓ Les enfants ne sont pas identiques aux parents car des altérations des gènes peuvent se produire (mutations).
- ✓ Parmi les mutations, certaines peuvent être favorables et d'autres défavorables.
- ✓ Il naît beaucoup de descendants : mais seuls les individus les mieux adaptés pourront survivre et transmettre leurs gènes à leur tour à leur descendance.

Dans ce modèle, on observe que [www11] :

- ✓ Le hasard joue un rôle moteur pour produire de nouveaux individus différents de leurs parents.
- ✓ La sélection naturelle effectue le tri entre les variations favorables et les autres.

Ainsi, basés sur la théorie de l'évolution naturelle des espèces énoncée par Darwin, ces algorithmes présentent des qualités intéressantes pour la résolution des problèmes d'optimisation. Les individus ou chromosomes d'un algorithme évolutionnaires sont des codages des solutions possibles du problème. Comme dans la nature, ces individus forment une population qui va évoluer dans le temps selon des lois de sélection qui vont favoriser les mieux adaptés à se croiser entre eux en produisant des populations meilleures. L'évolution des individus d'une population à une autre se fait à l'aide de la reproduction. Les individus parents vont se reproduire pour donner des individus enfants qui seront plus performants après avoir subis des opérations génétiques de croisement et de mutation.

Et comme ces reproductions se font avec une part de hasard par analogie à la nature, donc, les parents candidats à la reproduction sont choisis d'une manière probabiliste proportionnelle à leurs aptitudes et l'étape de reproduction est choisie d'une façon totalement aléatoire.

Finalement, passant d'une génération à une autre, les individus forment une progéniture plus performante qui s'approche au mieux de la solution optimale [Kamm, 2006].

La figure suivante résume le principe de résolution de problèmes par algorithmes évolutionnaires.

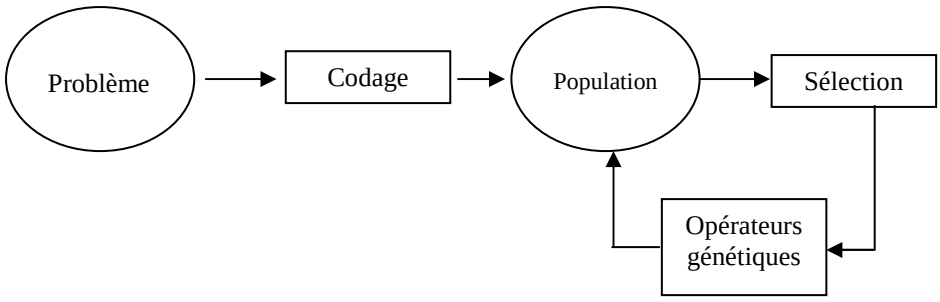


Figure II.10 . *Principe des algorithmes évolutionnaires.*

Ainsi, un algorithme évolutionnaire est construit autour des notions suivantes :

- **Le codage :**

La première étape de la résolution d'un problème à l'aide d'un algorithme génétique est la modélisation appropriée des solutions. Cette modélisation est appelée *codage* et permet de représenter les solutions sous forme de chromosomes.

Le codage se base sur deux notions importantes : le génotype et le phénotype.

- ✓ **Génotype** : représente l'ensemble des valeurs des gènes d'un chromosome.
- ✓ **Phénotype** : c'est la représentation de la solution du problème qui traduit les données contenues dans le génotype.

S'il y a passage immédiat du phénotype au génotype, le codage est dit direct, sinon il est dit indirect et une procédure de passage est indispensable. Le codage d'une solution doit décrire toutes les données du problème et respecter toutes ses contraintes.

- **Population initiale :**

L'espace de recherche étant infini, il est très difficile de répartir plus ou moins équitablement la population initiale sur l'espace de recherche, de sorte que cet espace soit au maximum parcouru. Un certain nombre d'individus doivent être générés aléatoirement s'il n'existe aucune autre méthode d'initialisation.

- **Evaluation de la qualité d'une solution :**

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance (*fitness*) de mesurer cette efficacité pour permettre à l'algorithme de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution. Autrement dit, la fonction de performance *f*, doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre.

- **La sélection :**

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.
Plusieurs modes de sélection ont été définis tels que : sélection par roulette (wheel), sélection par rang, sélection steady-state, élitisme, sélection uniforme, etc.

- **Opérateurs génétiques :**

1) Croisement : C'est un opérateur génétique qui permet à deux chromosomes parents de produire deux chromosomes enfants où de nouvelles séquences de gènes pour les chromosomes enfants sont crées à partir d'une base de configuration des séquences héritées des chromosomes parents. Cet opérateur se produit selon une probabilité *Pc* fixée par l'utilisateur selon le problème à optimiser.

Il existe plusieurs manières d'effectuer un croisement soit par cross-over où l'on a besoin de deux parents qui génèrent à la fin du croisement deux enfants, soit par copie où l'on n'a besoin que d'un seul parent qui nous donnera à la fin du croisement un enfant qui est le parent lui-même (sa copie).

Dans la littérature, il existe plusieurs opérateurs de croisement qui dépendent essentiellement du type du codage et de la nature du problème à traiter [Mesg, 1999]. Pour le codage binaire, nous distinguons plusieurs opérateurs de croisement tels que :

- ✓ le croisement à un point.
- ✓ le croisement multipoints.
- ✓ le croisement uniforme.

2) Mutation : permet de changer (permuter) un gène d'un chromosome par un autre d'une manière aléatoire, ce qui est à première vue très ressemblant avec un cross-over. La différence est que le cross-over essaie de converger vers une solution qui lui parait la meilleure (s'intéresse à la qualité), mais que la mutation permet la diversité. En quelque sorte, la mutation sert à éviter une convergence prématurée de l'algorithme. Par exemple, lors de la

recherche d'une solution optimum, la mutation sert à éviter la convergence vers un optimum local.
Cet opérateur est aussi appliqué avec une probabilité P_m. Il est nécessaire de choisir pour ce taux une valeur relativement faible de manière à ne pas tomber dans une recherche aléatoire et conserver le principe de sélection et d'évolution.

Dans la littérature plusieurs opérateurs de mutation ont été définis, tels que : la transposition de deux allèles consécutifs, la transposition de deux allèles quelconques, l'inversion d'allèles, etc.

Les algorithmes évolutionnaires se scindent en grandes familles, dont les principales sont :

1) Programmation évolutionnaire

La programmation évolutionnaire développée par *L.J. Fogel* [Foge, 2003], se base sur l'évolution d'une population d'automates à états finis (Figure II.11) pour résoudre des problèmes de prédiction. Ce modèle évolutionniste accentue l'utilisation de la mutation et n'utilise pas dans sa version originale la recombinaison des individus par croisement [Renn, 2000]. La table de transition des automates est modifiée grâce à des mutations aléatoires uniformes dans l'alphabet discret correspondant. Chaque automate de la population parente génère un enfant par mutation, et les meilleures solutions entre les parents et les enfants sont sélectionnées pour survivre, sachons que l'évaluation de la performance des individus correspond au nombre de symboles prédits correctement.

Par la suite, la programmation évolutionnaire a été développée et son domaine a été élargi par *D.B. Fogel*, afin qu'il puisse travailler dans l'espace réel, où la sélection déterministe est remplacée par un tournoi stochastique.

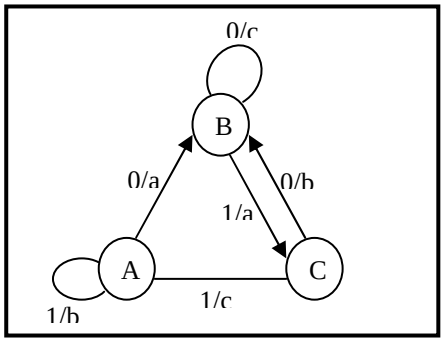


Figure II.11. *Exemple d'un automate à états finis ayant trois états différents S={A,B,C}, un alphabet d'entrée I={0,1}, et un alphabet de sortie O={a,b,c}. Chaque arrête entre deux états indique une transition possible, et la fonction de transition $\delta : S \times I \rightarrow S \times O$ est spécifiée par les labels au niveau des arrêtes ayant la forme i / o, signifiant que $\delta (s_i , i) = (s_j , o)$.*

2) Stratégies d'évolution

Les stratégies d'évolution sont dédiées à l'optimisation des problèmes continus dans l'espace de vecteurs des réels [Magn, 2001]. Les premiers efforts pour la mise en place des stratégies d'évolution ont eu lieu en 1973 à l'université de Berlin par Rechenberg [Rech, 2003] au cours de la résolution d'un problème aérodynamique. C'est avec ces méthodes évolutionnistes que la notion d'auto-adaptativité pour la mutation permettant de contrôler cette fonction de mutation, a été apparue. Une mise en œuvre de ce principe consiste à augmenter l'intensité de la mutation lorsque la proportion de descendants de bonne qualité, c'est à dire le nombre de mutation à succès, dépasse 20% de la population totale [Renn,

2000]. Elle est diminuée dans le cas opposé. Une interprétation possible de cette règle est la suivante : si la proportion de mutation réussie est élevée, l'espace de recherche exploré est restreint autour d'un optimum local, il faut donc diversifier la population en augmentant le taux de mutation, ce qui revient à ajuster la variance de la mutation au cours du temps par le processus d'évolution. C'est ce que les méthodes évolutionnistes auto-adaptatives visent de le faire : automatiser le réglage des paramètres de l'algorithme évolutionniste pour remédier aux méthodes empiriques du type « essais-erreurs » qui sont employées dans la pratique. De plus, ces approches utilisent un opérateur de sélection de type déterministe: les solutions dont le *fitness* est mauvais sont éliminées de la population. En outre, dans le modèle originel, les populations des parents et de leurs descendants sont généralement de taille différente.

3) Algorithmes génétiques

Les algorithmes génétiques (AGs) sont probablement les algorithmes les plus connus et les plus utilisés dans le calcul évolutionnaire. Ils ont été développé dans les années soixante par *Holland* qui les a appliqué à l'optimisation paramétrique pour la première fois en 1975 [Holl, 1975], en posant, ainsi, les fondements de cette technique d'application. Cependant, cette technique n'a pas été appliquée sur des problèmes réels de grande taille, à cause des machines calculatoires, de l'époque, et qui n'ont pas été suffisamment puissantes. Ce n'est que vers la fin des années quatre vingt, précisément, avec l'apparition de l'ouvrage de référence écrit par *Goldberg* [Gold, 1989], que les algorithmes génétiques ont été connus dans la communauté scientifique. Dans ce domaine, d'autres travaux peuvent faire la référence aussi, tel que celui de Michalewicz [Mich, 1996]. Leur particularité est qu'ils sont fondés sur le *Néo-Darwinisme*, c'est-à-dire l'union de la théorie de l'évolution et de la génétique moderne. Ainsi, les variables sont généralement codées en binaire, par analogie avec les quatre lettres de l'alphabet génétique d'ADN, sous forme de gènes dans un chromosome. Ensuite, des opérateurs génétiques, à savoir le croisement et la mutation, sont appliqués à ces chromosomes.

Les AGs sont utilisés pour retrouver une solution résolvant un problème donné, et ce sans avoir de connaissance à priori sur l'espace de recherche. Seul un critère de qualité est nécessaire pour évaluer les différentes solutions en quantifiant, ainsi, leur capacité à résoudre le problème donné. Donc, le but scientifique et technologique visé par ces algorithmes est de pouvoir traiter des problèmes d'optimisation globaux, grâce à la *généralité* avec laquelle on représente l'espace de recherche qui peut contenir des booléens (système actif ou non), des entiers (nombre de composants à optimiser), des réels (intensités associées aux composants réglables), ou des fonctions discrétisées (optimisation de forme), et grâce aussi à la *robustesse* de la convergence [Dupa, 2004].

4) Programmation génétique

L'idée de faire évoluer des programmes date des années cinquante où *Friedberg*, en 1958, a fait plusieurs tentatives pour avoir des ordinateurs auto-programmables en utilisant ce qui est de la mutation actuellement. Donc, et à partir d'une population constituée de programmes aléatoires dont il modifie leurs contenus stochastiquement, il essaye de les améliorer pour aboutir à des résultats satisfaisants. Plutard, *Smith* (1980) travaillant sur les systèmes classifieurs d'apprentissage, a introduit de petits programmes dans les règles qu'il cherche à les faire évoluer [Magn, 2001]. Il convient de noter que, la première utilisation des structures arborescentes dans un algorithme génétique a été suggérée par *Cramer* en 1985 dans le but de faire évoluer des sous-programmes séquentiels d'un langage algorithmique simple.

Toutefois, c'est grâce à *John Koza* (1992) [Koza, 1992] que cette présentation a été adoptée pour définir la programmation génétique comme un nouvel algorithme évolutionnaire, en étendant, ainsi, le modèle d'apprentissage des AGs à l'espace des programmes. Donc, son objectif initial était de faire évoluer des sous-programmes du langage LISP (Figure II.12), et c'est d'ailleurs grâce à son ouvrage que l'utilisation de la programmation génétique s'est étendue à la résolution de nombreux types de problèmes où les solutions peuvent être représentées par des structures arborescentes dont les feuilles sont constituées de symboles terminaux (variables, constantes,…), et les nœuds internes de symboles fonctionnels (opérateurs).

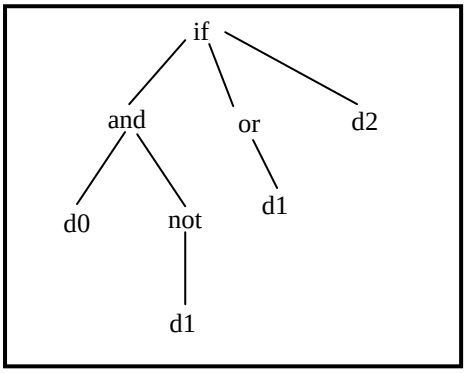


Figure II.12. *Exemple d'une solution Programmation génétique en LISP : {d0, d1, d2} est un ensemble d'instructions constituant les terminaux, et {if, and, or} sont des expressions LISP constituant les nœuds.*

5) Evolution différentielle

L'algorithme à évolution différentielle est inspiré des algorithmes génétiques et de l'évolution des stratégies combinées à une technique de recherche géométrique. Toutefois, l'évolution différentielle s'écarte encore un peu plus des principes de la génétique en abandonnant le codage génotype-phénotype et en faisant varier directement les paramètres réels proches du phénotype. L'évolution différentielle est donc valide pour tout type de fonction objective à valeurs réelles et peut tenir compte des propriétés mathématiques générales de ces fonctions. En revanche, tout comme l'algorithme génétique, l'évolution différentielle utilise une succession de générations définies par des opérations de mutation et de recombinaison de réels et donc possède des notions d'enfants et de parents [Laye, 2010].

L'évolution différentielle a été conçue comme une méthode de recherche parallèle, directe, et stochastique. Chaque solution est encodée avec un vecteur n-dimensionnel basé sur des nombres à virgule flottante. La taille de la population (m) ne change pas au cours du processus d'optimisation (minimisation ou maximisation). A chaque génération, de nouveaux vecteurs sont générés par la combinaison des vecteurs choisis au hasard de la population actuelle. La nouveauté de cet algorithme réside dans l'opérateur de mutation. Contrairement à la mutation des algorithmes génétiques, l'évolution différentielle utilise un concept de mutation auto référentiel se limitant à des combinaisons de mutations déjà présentes dans la population. L'opérateur de mutation consiste à générer de nouveaux vecteurs par le calcul de la différence pondérée de deux (ou quatre) autres vecteurs, selon la formule suivante [Laye, 2010] :

$$\overrightarrow{v_i} = \overrightarrow{a_{r1}} + F \times (\overrightarrow{a_{r2}} - \overrightarrow{a_{r3}})$$

Où i = 1,2,..., m, et les indices aléatoires r1, r2, r3 $\in$ [1,2,..., m] sont mutuellement différents et distincts de l'indice i. Cet opérateur est utilisé en liaison avec l'opérateur de croisement. Ce

dernier est introduit en vue d'augmenter la diversité de la population. Il combine un vecteur muté obtenu par l'opération de mutation avec l'un des vecteurs de la population. Finalement, l'opérateur de sélection compare seulement la valeur de la fonction objective des deux vecteurs en compétition et le meilleur individu est sélectionné pour la population de la prochaine génération.

Un algorithme d'évolution différentielle peut être résumé comme suit [Laye, 2010] :

Algorithme II.7 *Algorithme à évolution différentielle*

Début
Génération aléatoire de la population de vecteurs
Évaluer chaque solution
Répéter
Appliquer la mutation différentielle
Exécuter un croisement différentiel
Évaluer la nouvelle solution
Appliquer la sélection différentielle
Jusqu'à (le nombre de génération est atteint)
Fin.

6) La recherche par dispersion (Scatter Search)

La recherche par dispersion est une méthode d'optimisation relativement ancienne. Cette approche évolutionnaire a pour origine les stratégies de création de règles de décision composées et de contraintes de remplacement. Les études récentes ont démontré les avantages pratiques de cette approche pour résoudre divers problèmes d'optimisation. La recherche par dispersion contraste avec d'autres procédures évolutionnaires, telles que les algorithmes génétiques, en utilisant des conceptions stratégiques là où d'autres approches utilisent l'aléatoire [Jour, 2003].

La recherche par dispersion opère sur un ensemble de solutions appelé l'*ensemble de référence*, en les combinant pour en créer des nouvelles. À la différence d'une "population" dans les algorithmes génétiques, l'ensemble de référence de solutions dans la recherche par dispersion est relativement petit.

La recherche par dispersion comprend les cinq méthodes suivantes [Jour, 2003] :

- ✓ Une méthode de génération de diversification pour produire une collection de solutions diverses, en utilisant une solution d'essai arbitraire (ou solution initiale) comme entrée.
- ✓ Une méthode d'amélioration pour transformer une solution initiale en une ou plusieurs solutions d'essai améliorées.
- ✓ Une méthode de mise à jour de l'ensemble de référence pour construire et maintenir un ensemble de référence comprenant les meilleures solutions trouvées, organisé pour fournir l'accès efficace par d'autres parties de la méthode. L'incorporation de solutions à l'ensemble de référence est effectuée selon leur qualité ou leur diversité.
- ✓ Une méthode de génération d'un sous-ensemble pour opérer sur l'ensemble de référence, pour produire un sous-ensemble de ces solutions comme une base pour créer des solutions combinées.

- ✓ Une méthode de combinaison de solutions pour transformer un sous-ensemble donné de solutions produit par la méthode de génération d'un sous-ensemble en un ou plusieurs vecteurs combinés de solutions.

7) Algorithmes à Estimation de Distribution (Estimation of Distribution Algorithms)

Récemment, une nouvelle classe d'algorithmes a fait son apparition : les algorithmes à estimation de distribution (EDA). Les stratégies évolutionnaires mettent en œuvre des opérateurs de mutation et de croisement mais il est difficile pour un utilisateur inexpérimenté de choisir l'opérateur approprié à son problème. Les algorithmes à estimation de distribution reprennent les principes des algorithmes à population mais utilisent des modèles probabilistes à la place d'opérateurs de mutation et de croisement pour construire de nouveaux individus.

Le modèle de fonctionnement de l'algorithme est le suivant [Jour, 2003] :

1. Génération de la population initiale.
2. Sélection des individus prometteurs.
3. Estimation de la distribution de ces individus (construction d'un modèle probabiliste).
4. Génération de nouvelles solutions à partir du modèle probabiliste.
5. Retour en 2 jusqu'à ce que le critère d'arrêt soit satisfait.

Les EDA peuvent être appliqués aussi bien sur un domaine discret que sur un domaine continu. Il existe de nombreux algorithmes à estimation de distribution qui peuvent être classés selon le modèle utilisé pour la construction des nouveaux individus [Jour, 2003]: compact GA (produit de distribution de Bernouilli), Population Based Incremental Learning (règle de Hebian), Univariate Marginal Distribution Algorithm, extended compact GA (produit de distribution marginale) et Bayesian Optimization Algorithm (réseau bayesien).

8) Les algorithmes bactériologiques

Les algorithmes bactériologiques basés sur les algorithmes génétiques et qui sont donc assez similaires [Baud, 2005a], sont apparus récemment en tant que métaheuristique. Ils ont été développés par l'équipe Triskell [www12] à Rennes, notamment grâce à la thèse de Benoit Baudry [Baud, 2005b]. Lors du développement d'un générateur de cas de tests utilisant la technique de la mutation-based testing [Baud, 2005c], en utilisant les algorithmes génétiques, l'équipe s'est aperçue que l'utilisation d'algorithmes génétiques n'était pas bien adaptée. En effet, les nouveaux cas de tests n'étaient créés que par l'opérateur de mutation, le croisement ne faisant que récrire des cas déjà existants. L'équipe a donc eu l'idée d'utiliser la reproduction des bactéries, qui se multiplient en se clonant et des mutations s'opèrent. Les bactéries ayant le meilleur patrimoine génétique sont celles qui survivent le mieux dans leur environnement.

Les différences entre algorithmes génétiques et bactériologiques se situent au niveau du croisement des individus et au niveau de la sélection des individus. Dans l'algorithme bactériologiques l'opérateur de croisement a disparu, et pour ne pas perdre les bactéries les mieux adaptées, les meilleures sont sauvegardées lors de la sélection, à chaque itération. Il est possible d'en sauvegarder un certain nombre, mais aussi de sauvegarder les X meilleures. Les premières itérations entraînent alors la sauvegarde de toutes les bactéries.

Un algorithme bactériologique se déroule de manière très similaire à un algorithme génétique. Il opère suivant les étapes suivantes :

- ✓ initialisation du temps,
- ✓ création de la population initiale,
- ✓ tant que [il n'y a pas de solution satisfaisante] et [le temps limite n'est pas atteint], faire:
  - incrémentation du temps,
  - mutations aléatoires à la population (Toutes les bactéries mutent),
  - évaluation de l'adaptation de chaque bactérie,
  - sauvegarde des meilleures bactéries.

Actuellement, la distinction entre ces approches est de plus en plus floue. Le génotype, qui est le codage d'un individu, étant souvent constitué d'un mélange de structures complexes (arbres, graphes, listes de paramètres, ...). Donc, la différence entre ces quatre catégories est essentiellement d'ordre historique.

b. Algorithmes de colonies de fourmis

L'histoire de l'intelligence en essaim remonte à l'étude du comportement de fourmis, à la recherche de nourriture au départ de leur nid, par Goss, Deneubourg et leur équipe [Dene, 1983], [Dene, 1989]. Les fourmis trouvent le plus court chemin entre leur nid et une source de nourriture. La résolution de ce problème, qui est assez complexe en soi, fait appel à une certaine organisation et à un travail collectif. Des fourmis peuvent résoudre ce problème collectivement en se basant sur un moyen de communication particulier : « la phéromone ».

1) Les fourmis réelles

Les fourmis réelles sont aveugles et cherchent de la nourriture en se déplaçant de façon quasi aléatoire. Tout au long de leur déplacement, elles laissent derrière elles une substance chimique appelée phéromone. Cette substance a pour but de guider les fourmis vers leur objectif et possède la propriété de s'évaporer au cours du temps. Une fois cet objectif atteint (dans ce cas, la nourriture trouvée), les fourmis rentrent au nid, en rebroussant chemin, grâce à leur trace de phéromone. Celle-ci s'en trouve renforcée. Plus une trace de phéromone est concentrée, plus elle va attirer les fourmis. Au fil du temps, le plus court chemin, du nid vers la nourriture émergera, grâce au renforcement de la trace de phéromone (figure II.13). D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs congénères.

Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis. Ainsi, une colonie est capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter [Goss, 1989] [Beck, 1992], sans que les individus aient une vision globale du trajet.

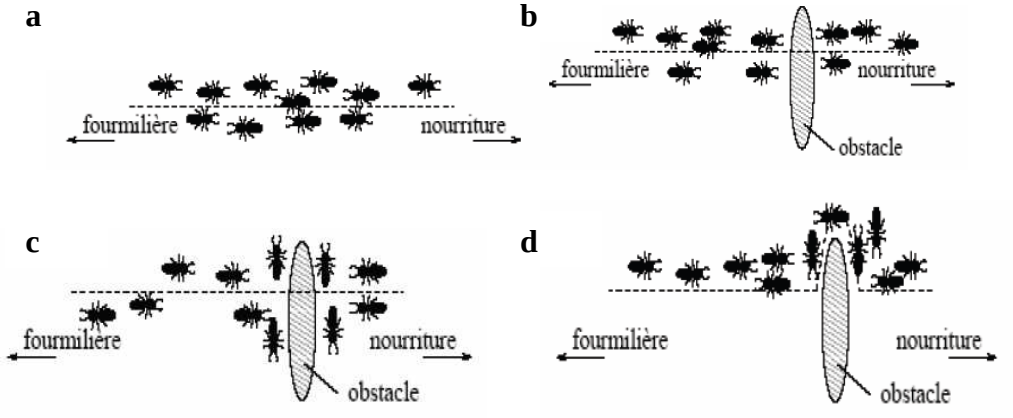


Figure II.13. (a) *Les fourmis suivent un chemin entre la fourmilière et la nourriture,*
(b) *Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité.*
(c) *La phéromone s'évapore sur le chemin le plus long.*
(d) *Toutes les fourmis choisissent le chemin le plus court.*

2) Les algorithmes de colonies de fourmis

Le premier algorithme à base de fourmis a été proposé par Dorigo en 1992 [Dori, 1992]. La métaheuristique d'optimisation par colonies de fourmis (ACO, Ant Colony Optimization) qui est une approche bio-inspirée [Dori, 1999], [Dori, 2004] est une généralisation des premiers algorithmes à base de fourmis.

Ce paradigme consiste à modéliser le problème à résoudre en une recherche d'un meilleur chemin dans un graphe et à utiliser des fourmis artificielles pour rechercher les "bons" chemins dans ce graphe. Le comportement de ces fourmis artificielles est inspiré des fourmis réelles : *(i)* les fourmis déposent de la phéromone pour marquer les chemins prometteurs, *(ii)* elles se déplacent dans le graphe de construction en choisissant leur chemin selon une probabilité dépendant des traces de phéromones précédemment déposées, et *(iii)* la quantité de phéromone déposée sur les chemins décroît à chaque cycle de l'algorithme afin de simuler le phénomène d'évaporation de la phéromone observé dans la nature.

Une formalisation plus précise existe [Dori, 2003]. Elle passe par une représentation du problème, un comportement de base des fourmis et une organisation générale de la métaheuristique. Plusieurs concepts sont également à mettre en valeur pour comprendre les principes de ces algorithmes, notamment la définition des pistes de phéromone en tant que mémoire adaptative, la nécessité d'un réglage intensification/diversification et enfin l'utilisation d'une recherche locale.

c. Algorithmes à essaim de particules (Particle Swarm Optimiser)

Ces algorithmes sont inspirés des essaims d'insectes [Cler, 2004] (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés *particules* et la population est appelée *essaim*.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou socio-

métriquement (position dans l'essaim de l'individu). Les nouvelles vitesses et direction de la particule seront définies en fonction de trois tendances : la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues.

II.6. Quelle métaheuristique à utiliser ?

Le premier problème pratique qui se pose à un utilisateur confronté à une application concrète est d'effectuer un choix parmi les différentes métaheuristiques disponibles. Ce choix est d'autant plus difficile qu'il n'existe pas de comparaison générale et fiable des différentes métaheuristiques. Cependant, il est possible de caractériser les métaheuristiques selon quelques critères généraux, ce qui pourrait faciliter ce choix. Le tableau de synthèse ci-dessous met en relation cinq critères avec six métaheuristiques parmi les plus représentatives. Les indications sont présentées à titre purement indicatif et correspondent aux travaux de [JinK, 1999] et aux résultats publiés. Il doit être clair que le choix d'une métaheuristique appropriée ne constitue qu'une condition nécessaire. La qualité des solutions trouvées par une méthode peut être très variable selon l'implémentation réalisée.

	AI	**RS**	**Tabou**	**AG**	**ACO**	**AH**
Simplicité	0	-	-	-	- -	- -
Facilité d'adatp.	0	0	-	-	-	-
Connaissance	0	0	+	+	+	+
Qualité	0	+	++	+	+	++ (+++)
Rapidité	0	-	-	- -	- -	- -

Tableau II.1. *Comparaison générale des principales métaheuristiques.*

Dans ce tableau, les six métaheuristiques comparées sont les suivantes :

- ✓ AI : amélioration itérative (descente) avec relance,
- ✓ RS : recuit simulé,
- ✓ tabou : méthode tabou,
- ✓ AG : algorithme génétique adapté,
- ✓ ACO : optimisation par colonies de fourmis,
- ✓ AH : algorithme hybride.

Les critères de comparaisons retenus sont les suivants :

- ✓ simplicité de la métaheuristique, *i.e.* simplicité de la méthode elle-même,
- ✓ facilité d'adaptation au problème,
- ✓ possibilité d'intégrer des connaissances spécifiques du problème,
- ✓ qualité correspond à la meilleure qualité qu'il est possible d'obtenir par une exécution prolongée,
- ✓ rapidité, *i.e.*, le temps de calcul nécessaire pour trouver une telle solution (sur une machine séquentielle).

La méthode d'amélioration itérative est utilisée comme point de référence pour l'ensemble des méthodes : les signes -, 0, + indiquent des performances respectivement inférieures, égales, supérieures à celles obtenues par l'amélioration itérative.

II.7. Conclusion

Les métaheuristiques constituent une classe de méthodes approchées adaptables à un très grand nombre de problèmes combinatoires. Elles ont révélé leur grande efficacité pour fournir des solutions approchées de bonne qualité pour un grand nombre de problèmes d'optimisation classiques et d'applications réelles de grande taille. C'est pourquoi l'étude de ces méthodes reste toujours en plein développement.

Si on peut constater la grande efficacité des métaheuristiques pour de nombreuses classes de problèmes, il existe en revanche peu de résultats permettant de comprendre la raison de cette efficacité. Nous possédons bien des comparaisons entre métaheuristiques sur différentes classes de problèmes mais nous ne savons généralement ni expliquer le fonctionnement d'une métaheuristique, ni prévoir son efficacité sur une instance donnée.

Dans les suivants chapitres, nous présenterons nos travaux de recherche qui se résument en l’application des métaheuristiques pour résoudre le problème de cryptage de données texte et images.

CHAPITRE III

CRYPTAGE EVOLUTIONNAIRE DES DONNEES TEXTES ET IMAGES

III.1. Introduction

Dans la littérature, les méthodes heuristiques sont réparties en deux classes : les algorithmes spécifiques à un problème donné, qui utilisent des connaissances du domaine [Talb, 1999], et les algorithmes généraux applicables à une grande variété de problèmes : les métaheuristiques [Dréo, 2003]. Dans ce chapitre, notre intérêt va se porter sur la deuxième classe d'algorithmes. Pour résoudre des problèmes multi-objectifs ou mono-objectifs et déterminer des solutions optimales, plusieurs adaptations des métaheuristiques sont proposées dans la littérature. Les plus connues de ces adaptations peuvent être trouvées dans [Coll, 2002].

Etant donné la complexité du problème traité qui celui de cryptage de données texte ou images, nous avons choisi d'utiliser les métaheuristiques pour le résoudre. La première méta-heuristique testée est un algorithme évolutionnaire (AE).

Le principal avantage des AEs vient de leur capacité à traiter le problème en ne possédant qu'un minimum d'informations sur celui-ci et en ne laissant aucun détail sur les calculs intermédiaires menant aux résultats. Ce dernier point convient parfaitement au domaine de chiffrement de données pour compliquer voir même pénaliser toutes tentatives de cryptanalyse.

Ainsi, le problème considéré qui est celui de chiffrement de données, peut être résolu par une procédure d'optimisation par AEs qui à partir d'un ensemble de solutions initiales, ou population de N individus, elle consiste à faire évoluer cette population en utilisant des opérateurs de sélection, de croisement et de mutation. A chaque itération de l'algorithme, une nouvelle population de solutions ou d'individus est générée. Tout d'abord, un ensemble d'individus est sélectionné pour générer la population suivante. Ces individus sont ensuite croisés pour créer de nouveaux individus et compléter la nouvelle population. Certains de ces nouveaux individus peuvent subir une mutation. Le critère d'arrêt de l'algorithme dans notre cas est un nombre d'itérations sans amélioration de la meilleure solution trouvée.

Algorithme 1 Structure générale de l'algorithme génétique

1: $\mathcal{P}_{courant} \leftarrow$ Initialiser une population de N individus
2: Evaluer chaque individu de $\mathcal{P}_{courant}$
3: $S_{best} \leftarrow$ Le meilleur individu $S \in \mathcal{P}_{courant}$
4: $I \leftarrow 0$
5: **Tant que** $I < \#ite$ **faire**
6: $\quad \mathcal{P}_{enfant} \leftarrow \emptyset$
7: $\quad$ **Pour** $j = 0$ à $j = N/2$ **faire**
8: $\quad\quad (P_1, P_2) \leftarrow$ Sélectionner deux individus parents de $\mathcal{P}_{courant}$
9: $\quad\quad (E_1, E_2) \leftarrow$ Croiser les deux parents (P_1, P_2) pour obtenir deux individus enfants
10: $\quad\quad \mathcal{P}_{enfant} \leftarrow \mathcal{P}_{enfant} \cup \{E_1, E_2\}$
11: $\quad$ **Fin pour**
12: $\quad$ Muter aléatoirement des individus de la population $\mathcal{P}_{enfant}$
13: $\quad \mathcal{P}_{courant} \leftarrow \mathcal{P}_{enfant}$
14: $\quad$ Evaluer chaque individu de $\mathcal{P}_{courant}$
15: $\quad$ **Si** il existe un individu $S \in \mathcal{P}_{courant}$ meilleur que S_{best} **alors**
16: $\quad\quad S_{best} \leftarrow S$
17: $\quad\quad I \leftarrow 0$
18: $\quad$ **Fin si**
19: $\quad I \leftarrow I + 1$
20: **Fin Tant que**

Algorithme III.1. *Structure générale d'un AE.*

Les éléments importants d'un algorithme évolutionnaire sont le codage et l'évaluation d'un individu (étapes 2 et 14), l'initialisation d'une population (étape 1), la sélection (étape 8), le croisement (étape 9) et la mutation des individus (étape 12). Ces éléments sont décrits dans les pages qui suivent, dans le cadre de l'adaptation que nous en avons faite au problème de cryptage.

Dans ce chapitre, nous présentons des nouveaux algorithmes de cryptage de données texte et images basés sur les AEs et opérant suivant deux modes : un chiffrement à base de positions et un chiffrement à base d'occurrences [Soui, 2008a], [Soui, 2008b] [Soui, 2009], [Sema, 2009a], [Sema, 2009b], [Soui, 2010a], [Soui, 2010b], [Soui, 2011a] et [Soui, 2011b]. Ce chapitre est scindé en cinq sections où la première présente une introduction au chapitre. La deuxième section démontre l'adaptabilité d'exploitation des AEs pour résoudre le problème de cryptage en énumérant les principaux points de motivation ; suivie de la troisième section consacrée à une formulation du problème de cryptage en tant que problème d'optimisation. Les hypothèses sur lesquelles repose cette approche sont exposées aussi dans cette section. Ensuite, une description détaillée des algorithmes développés est traitée dans la quatrième section. Ainsi, elle englobe la présentation d'algorithmes de cryptage développés, le réglage des paramètres de la métaheuristique, ainsi que les résultats. La cinquième section présente une étude comparative des algorithmes développés, d'une part, et des méthodes de référence, d'autre part. Le chapitre sera terminé par une conclusion.

Les données réelles de test que nous avons choisies pour illustrer les différents algorithmes, sont présentées sur la figure III.1. Il est à signaler que nos algorithmes ont été codés sous Matlab, et exécutés sur un processeur Intel Pentium 4 à 2,26 GHz.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

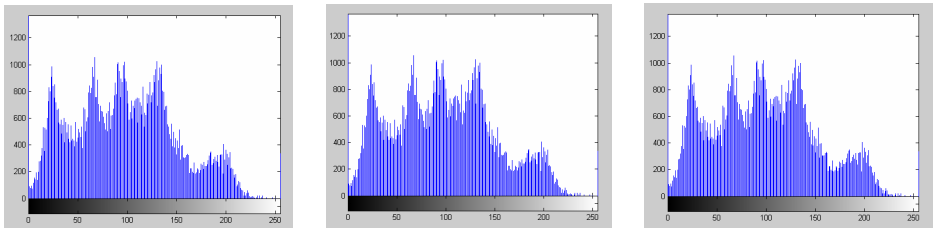
(a)

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

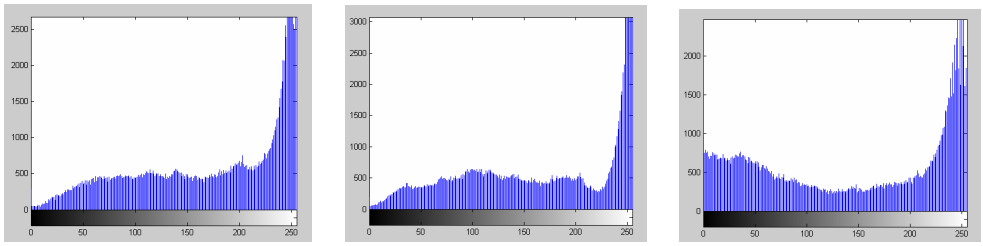
(b)

(c)

(d)



(e)



(f)

Figure III.1. *Données test.* ***(a)*** *Texte 1 (Taille = 1084),* ***(b)*** *Texte 2 (Taille = 1151),*
(c) *Image Lena,* ***(d)*** *Image Logo,* ***(e)*** *Histogrammes de l'image Lena,* ***(f)*** *Histogrammes de l'image Logo.*

III.2. Motivations

Les algorithmes évolutionnaires ont montré leur efficacité dans la résolution de nombreux problèmes et notamment dans les problèmes d'optimisation. La modélisation de l'espace de recherche que nous avons adoptée nous permet de tirer parti des propriétés des AEs en optimisation dans notre problème.

Notre approche se distingue des autres approches cryptographiques construites autour de principes mathématiques complexes (problèmes réputés difficiles), dans la mesure où nous modélisons le problème de cryptage à un niveau proche d'un paysage de qualité (fitness landscape) : l'AE traite directement avec les points de l'espace de recherche, les données.

Les algorithmes que nous avons mis au point bénéficient des avantages de la stratégie de recherche évolutionnaire surtout du bon comportement des AEs en ce qui concerne la résolution du dilemme d'exploration versus exploitation [Holl, 1975] : il décide quelles solutions intermédiaires proposées sont à garder en priorité et quelles sont les autres à éliminer.

III.3. Formulation du problème de chiffrement

Dans cette section, nous montrons que le chiffrement de données peut se ramener à un problème d'optimisation. D'où l'adaptabilité d'utiliser une métaheuristique.

Le chiffrement d'une donnée *D* (texte ou image) qui est une suite de *n* éléments (caractères ou pixels) *e* utilisant un attribut d'égalité *E*, génère une donnée chiffrée *D´* qui est une suite de *n* éléments *e´*, telle que :

1- $D = \{e_i\}, i \in [1,n]$

2- $D' = \{e_i'\}, i \in [1,n]$

3- $E(e_i, e_i') = Faux, \forall i \in [1,n]$

Nous faisons observer que l'unicité de chiffrement n'est pas garantie par ces trois conditions. Pour réduire le problème de la non unicité de la solution, le problème de chiffrement est régularisé par une contrainte d'optimisation d'une fonction *F*, caractérisant la qualité d'un bon chiffrement. Donc, une quatrième condition est ajoutée aux trois premières :

4- $F(D^*) = Max_{D' \in C(D)} F(D')$

Où *F* est une fonction mesurant le degré de confusion des données, D^* est la donnée chiffrée optimale ou plutôt la meilleure donnée chiffrée trouvée, *C(D)* est l'ensemble des données chiffrées possibles de *D*.

Il est clair que la condition 4 ne résout pas entièrement le problème d'unicité de chiffrement. Il demeure des cas où plusieurs chiffrements peuvent avoir la même valeur optimale. Toutefois, ce problème peut être réglé expérimentalement en fixant un nombre maximal d'itérations du processus de chiffrement.

La détermination d'un niveau de confusion mesurant le degré de dissimilarité entre la donnée originale et la donnée chiffrée correspondante rend le cryptage de données assimilable à un problème d'optimisation. D'où notre approche de cryptage au travers des méthodes heuristiques et des techniques destinées à résoudre ce type de problèmes.

Cette reformulation du problème de cryptage de données en un problème d'optimisation, nous conduit à la section suivante, où nous allons présenter les différents algorithmes proposés.

III.4. Algorithmes proposés

En cryptage de données, les AEs ont été récemment appliqués à travers le travail de Omary Fouzia [Omar, 2007]. C'est d'ailleurs l'unique méta-heuristique qui a été utilisée pour résoudre ce problème. Une étude comparative entre ce travail et notre proposition d'AEs de cryptage sera ensuite présentée.

Dans cette approche proposée, nous nous intéressons au cryptage des données texte et images. Le schéma général du processus de sécurisation proposé est celui illustré par le synoptique de la figure III.2.

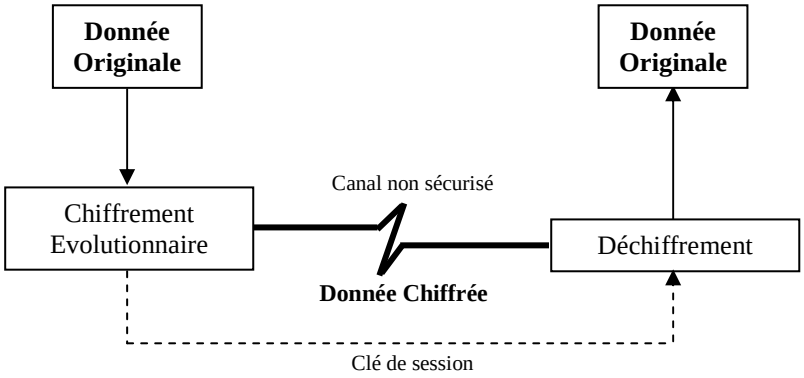


Figure III.2. *Schéma général du processus de sécurisation proposé.*

Dans ce cas et comme la première étape de la conception d'un AE est de coder (représenter) les solutions sous forme de chromosomes qui sont des chaînes de gènes sachons que cette étape dépend principalement de l'application envisagée et du but recherché [Prab, 1998], alors, nous avons proposé deux modes de chiffrements différents : chiffrement à base de position et chiffrement à base d'occurrences. Ainsi, trois algorithmes différents en résultent, un premier algorithme et un deuxième représentant une variante plus sécurisée du premier pour le cas de manipulation des images représentant des données spéciales pour le premier mode et un troisième algorithme pour le second mode.

Donc, ce qui fait la différence entre les trois algorithmes proposés est le codage utilisé et le paramétrage employé. Toutefois, le schéma global du processus de chiffrement ou de déchiffrement (illustré en détail dans les sections qui suivent) est le même. C'est celui donné ci-dessous :

Phase de chiffrement

Début

1) Définition d'un codage du problème,

Répéter

2) création de la population initiale,

3) sélection parmi les parents (individus de la population courante) ceux qui vont avoir des enfants,

4) application des opérateurs de variation (croisement / mutation) aux parents sélectionnés (génération des enfants),

5) évaluation des performances des individus,

6) sélection, parmi les parents et les enfants, de ceux qui vont survivre à la génération suivante.

Jusqu'à la satisfaction d'un critère d'arrêt.

7) Calcul de la clé de session correspondante à la solution produite.

Fin

Phase de déchiffrement

Début
Si La clé de déchiffrement est la bonne clé calculée **Alors**
1) Calcul du chromosome codant la donnée originale
2) Génération de la donnée originale à partir du chromosome calculé
Fin si
Fin

III.4.1. Chiffrement à base de positions

Dans cette section nous présentons deux nouveaux algorithmes de chiffrement symétrique s'inscrivant sous ce premier mode proposé où le deuxième est une variante du premier. Nous les avons appelé *PosESecL1* et *PosESecL2* pour *Position based Evolutionary Secure Level 1* (niveau 1 de sécurité évolutionnaire basé positions) et *Position based Evolutionary Secure Level 2* (niveau 2 de sécurité évolutionnaire basé positions), respectivement.

III.4.1.1. Description de PosESecL1

Avant de décrire les étapes du processus évolutionnaire en allant de la génération de la population initiale jusqu'à l'obtention du résultat final, il faut définir le codage adéquat des individus.

a. Codage

Le codage dépend étroitement du problème à résoudre. En effet sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide.

Dans le cas de manipulation d'une donnée *D* qui soit une suite de *n* éléments e_i, le codage adopté sera celui décrit à travers la figure III.3. Il consiste à transformer la donnée en un code particulier représenté par un chromosome qui est un ensemble de gènes représentant chacun le codage d'un élément e_i. Donc, en considérant le codage d'un certain élément, nous cherchons à permuter sa position (son emplacement) avec un autre élément de telle sorte que la différence entre les codages des deux éléments soit maximale. Il s'agit, ainsi, d'un problème d'optimisation où le but de l'algorithme proposé est de désordonner les positions des éléments suivant les contraintes du codage utilisé.

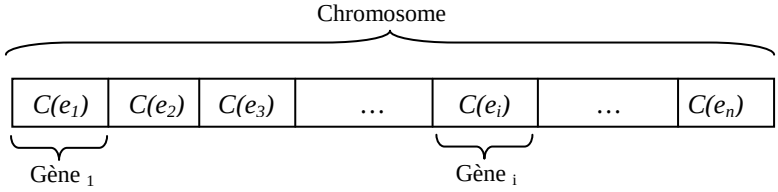


Figure III.3. *Codage adopté des données texte / images sous PosESecL1.*

Où $C(e_i)$ représente le codage de l'élément *i*.

Dans le cas des données texte, $C(e_i)$ représente la codification des caractères constituant la donnée. Etant donné qu'une donnée texte soit une suite de caractères appartenant à la liste des 1393 caractères affichables en Unicode et couvrant plusieurs sous-ensembles de caractères : Latin de base, Latin étendu-A, latin étendu-B, lettres de modification d'espace, Grec, Cyrillique, Hébreu, Hébreu étendu, Arabe, Arabe étendu, ponctuation générale, etc. Donc, $C(e_i)$ représente le codage Unicode du caractère e_i. La représentation que nous avons utilisée est le codage hexadécimal malgré que le codage entier soit tout à fait adapté.

Dans le cas des données images, et vu que l'espace de représentation choisis est le RVB (Rouge Vert Bleu), donc, nous avons utilisé un codage entier malgré que le codage binaire semble tout à fait adapté. Toutefois, cette représentation semble peu appropriée, dans notre cas, car elle nécessite deux opérations supplémentaires, le codage et le décodage qui n'apportent aucun plus par rapport au codage choisi.

Ainsi, les chromosomes sont des chaînes de *n* gènes ayant comme structure celle représentée à travers la figure III.4, où *n* représente la taille de l'image à chiffrer en terme de pixels. Donc, en considérant les trois composantes R_i, V_i et B_i relatives au $i^{ème}$ pixel de l'image, nous cherchons à permuter sa position (son emplacement) avec un autre pixel, ayant comme rang j et représenté par les composantes R_j, V_j et B_j, de telle manière que les différences entre les composantes (R_i,R_j), (V_i,V_j) et (B_i,B_j) soient maximales.

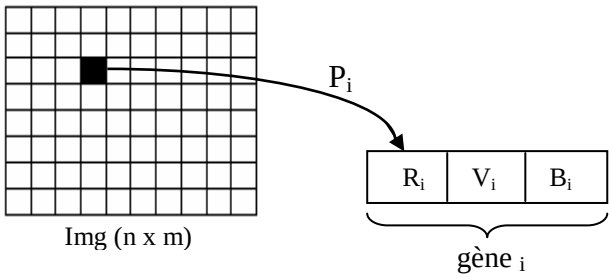


Figure III.4. *Représentation d'un pixel P_i de l'image Img ($n \times m$) sous la forme d'un gène.*

b. Création de la population initiale

La grande partie des versions des algorithmes évolutionnaires garde la même taille de la population des solutions potentielles, appelées aussi *individus*. La population initiale peut être choisie de manière aléatoire, donnée par l'utilisateur ou une partie est créée et l'autre est donnée (pour introduire dès le début de bonnes solutions construites, par exemple, à partir de certaines heuristiques). Cependant d'autres mécanismes d'initialisation peuvent être utilisés suivant l'application [Bhanu, et al., 1995].

Dans notre cas, les individus formant la population initiale sont obtenus par application de perturbations aléatoires sur le chromosome initial représentant le codage d'une donnée soumise au chiffrement.

c. Reproduction

Le but de notre algorithme PosESecL1 est de désordonner les positions des éléments d'une donnée originale de façon évolutionnaire ne laissant aucune trace des calculs intermédiaires menant au résultat final qui est le chiffré de la donnée initiale, sans modifier les valeurs des éléments. Donc, nous devons être très prudent lors de la manipulation des chromosomes durant les différentes étapes du processus évolutionnaire ; surtout, lors de l'application des opérateurs génétiques.

Ci-dessous, nous décrivons les opérateurs de reproduction exploités par le PosESecL1.

- **Croisement**

L'opérateur de croisement a pour but d'enrichir la diversité de la population, en manipulant la structure des chromosomes. Classiquement, les croisements sont réalisés à partir de deux parents et génèrent deux enfants. Une description plus détaillée a été exposée dans le deuxième chapitre.

Comme opérateur de croisement, notre choix s'est porté sur l'opérateur OX « Order Cross-over » proposé par Davis [Davi, 1985]. C'est un opérateur à deux points de croisement. Il consiste à générer des descendants en trois phases :

1**)** Choisir dans les deux parents une sous-séquence interne, comprise entre deux points de coupure tirés aléatoirement.

$Parent_1$: 1 3 5 7 9 | **2 4 6** | 8 10

$Parent_2$: 10 1 9 2 8 | **7 3 4** | 6 5

2) Recopier la sous-séquence interne du Parent1 dans le descendant Enfant1 aux mêmes positions et retirer du chromosome Parent2 les allèles compris dans cette sous-séquence.

$Enfant_1$: • • • • • | **2 4 6** | • •

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Le chromosome Parent2 permet alors de former une séquence d'allèles résiduelle en partant du deuxième point de coupure et en considérant le chromosome comme une chaîne circulaire.

3) compléter les gènes disponibles du descendant Enfant1 en lui transmettant dans l'ordre les allèles issus de la séquence résiduelle précédente.

$Enfant_1$: 1 9 8 7 3 | **2 4 6** | 5 10

$Parent_2$: 10 1 9 • 8 | **7 3** • | • 5

Les deux points de croisement sont choisis aléatoirement. Le meilleur taux de croisement varie entre 60% et 100% [Gren, 1986].

- **Mutation**

Pratiquement, le rôle de la mutation consiste à faire apparaître de nouveaux gènes. Cet opérateur introduit une diversité nécessaire à l'exploration de l'espace de recherche. Les mutations jouent, alors, le rôle de bruit et empêchent l'évolution de se figer.

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

d. Evaluation des individus

La fonction d'évaluation (ou d'adaptation) quantifie la qualité de chaque chromosome par rapport au problème. Les chromosomes ayant une bonne qualité ont plus de chance d'être sélectionnés pour la reproduction, et donc plus de chance pour que la population suivante hérite de leur matériel génétique. La fonction d'adaptation produit la pression qui permet de

faire évoluer la population de l'algorithme évolutionnaire vers les individus de meilleure qualité. En clair, le choix de la fonction d'évaluation va fortement influer sur le succès de l'algorithme.

La fonction d'évaluation que nous avons définie pour évaluer nos individus, est celle donnée ci-dessous :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.1)}$$

Avec : $C(e_j)_i$ est le codage du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.

Dans le cas de manipulation d'une donnée image, la fonction d'évaluation prend l'instanciation suivante :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \qquad \text{(III.2)}$$

Où : R_{ji} (resp V_{ji}, B_{ji}) est la valeur de la composante R (resp V, B) du j^{ème} gène du i^{ème} individu.
n est la taille de l'image à chiffrer en terme de pixels.

e. Sélection des individus les plus adaptés

Le rôle de la sélection est de distinguer entre les individus sur la base de leur qualité, en particulier, pour permettre aux meilleurs individus de devenir parents dans la génération suivante. Ainsi, elle est responsable sur le fait de pousser l'amélioration de la qualité.

Dans notre cas l'opérateur utilisé est une sélection de type roulette avec la possibilité de sélectionner plusieurs fois le même individu. Ainsi, les meilleurs individus ont plus de chance d'être sélectionnés par rapport au moins bons individus.

f. Critère d'arrêt

Le test d'arrêt joue un rôle primordial dans le jugement de la qualité des individus. Son but est d'assurer l'optimalité de la solution finale obtenue par l'algorithme évolutionnaire.

Les critères d'arrêts sont de deux natures :

1- Arrêt après un nombre fixé à priori de générations. C'est la solution retenue lorsqu'une durée maximale de temps de calcul est imposée.
2- Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment. Nous sommes alors en présence d'une population homogène dont on peut penser qu'elle se situe à la proximité de l'optimum. Ce test d'arrêt reste le plus objectif et le plus utilisé. C'est d'ailleurs celui que nous avons principalement utilisé pour assurer la convergence de notre algorithme proposé.

Pour notre problème et lors de la manipulation d'une donnée texte, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| \qquad \text{(III.3)}$$

Et
comme :

$$((0020 \le C(e_j)_i \le FEFC) \,\&\, (0020 \le C(e_j)_0 \le FEFC)) \Rightarrow (0 \le \left| C(e_j)_i - C(e_j)_0 \right| < FEFC)$$

(III.4)

$$(III.4) \Rightarrow 0 \le \sum_{j=1}^{n} \left| C(e_j)_i - C(e_j)_0 \right| < FEFC \times n$$

D'après l'inéquation (III.4), F est une fonction bornée donc, la fonction d'arrêt mettant fin au processus de résolution est la suivante :

$$0 \le F(I_i) < FEFC \times n \quad (III.5)$$

telles que : $(FEFC)_{16} = (65276)_{10}$ et $(0020)_{16} = (0032)_{10}$

Dans le cas de manipulation d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \quad (III.6)$$

Et comme :

$$((0 \le R_{ji} \le 255) \,\&\, (0 \le R_{j0} \le 255)) \Rightarrow (0 \le \left| R_{ji} - R_{j0} \right| \le 255) \quad (III.7)$$

$$((0 \le V_{ji} \le 255) \,\&\, (0 \le V_{j0} \le 255)) \Rightarrow (0 \le \left| V_{ji} - V_{j0} \right| \le 255) \quad (III.8)$$

$$((0 \le B_{ji} \le 255) \,\&\, (0 \le B_{j0} \le 255)) \Rightarrow (0 \le \left| B_{ji} - B_{j0} \right| \le 255) \quad (III.9)$$

D'après (III.7), (III.8) et (III.9) nous aurons :

$$0 \le \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \quad (III.10)$$

$$(III.10) \Rightarrow 0 \le \sum_{j=1}^{n} \left| R_{ji} - R_{j0} \right| + \left| V_{ji} - V_{j0} \right| + \left| B_{ji} - B_{j0} \right| \le 255 \times 3 \times n$$

$$\Leftrightarrow 0 \le F(I_i) \le 255 \times 3 \times n \quad (III.11)$$

D'après l'inéquation (III.11), nous constatons que F est une fonction bornée. Ainsi, la condition d'arrêt est la suivante :

$$0 \le F(I_i) \le 255 \times 3 \times n$$

g. Déchiffrement

Le déchiffrement est l'opération inverse du chiffrement. Elle permet d'obtenir la version originale d'une donnée qui a été précédemment chiffrée.

Dans la deuxième phase de l'algorithme correspondant à cette opération, nous exploitons deux informations qui sont la donnée originale et la donnée chiffrée pour générer une *clé de session* qui peut être d'un usage symétrique ou asymétrique. Cette clé représente la permutation des positions des nombres d'occurrences des valeurs d'éléments codant la donnée chiffrée pour obtenir ceux des valeurs d'éléments codant la donnée originale. De ce fait, elle varie d'une donnée à l'autre puisqu'elle dépend de la donnée et de sa taille. Et ce n'est que par l'introduction de la clé appropriée que les éléments de la donnée chiffrée rejoignent leurs positions initiales pour retrouver la donnée originale.

Remarque :

La notion de *clé de session* est un compromis entre le chiffrement symétrique et asymétrique permettant de combiner les deux techniques. Son principe est simple : il consiste à générer aléatoirement une clé de session de taille raisonnable, et de chiffrer celle-ci à l'aide d'un algorithme de chiffrement à clef publique (plus exactement à l'aide de la clé publique du destinataire). Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée. Ainsi, expéditeur et destinataire sont en possession d'une clé commune dont ils sont seuls connaisseurs. Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

h. Réglage des paramètres et résultats

Dans cette partie, nous présentons des résultats de cryptage de données test (présentées sur la Figure III.1), obtenus avec l'algorithme PosESecL1. Cette partie est scindée en deux sous-parties : dans la première, nous détaillons les réglages de l'algorithme. La deuxième sous-partie est consacrée aux résultats de cryptage des données test.

- **Réglage des paramètres**

En littérature, plusieurs travaux ont traité ce problème tels que : [Lobo, 2004], [DeJo, 2007], [Eibe, 2007], [Preu, 2007], [Mich, 2007] et [Fern, 2007]. Toutefois, la nécessité de l'étape de réglage vient des inconvénients majeurs des métaheuristiques : plusieurs paramètres à régler et l'inexistence de réglages par défaut. En outre, chaque problème traité a ses propres réglages. Donc et pour pouvoir choisir les bons paramètres relatifs aux taux de croisement et de mutation, il a fallu appliquer l'algorithme plusieurs fois. Les différentes valeurs de test appartiennent aux intervalles [0.6, 1] et [0.001,0.05] pour les probabilités de croisement et de mutation respectivement. Les figures III.5 et III.6 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation. Il est à signaler que les résultats présentés sont la moyenne de 10 exécutions successives d'un chiffrement de l'image Lena.

Les valeurs des paramètres finalement adoptées par PosESecL1 sont récapitulées dans le tableau III.1. Cependant, il est à noter que les très nombreux paramètres et leurs fortes interactions rendent impossible un réglage optimal pour toutes les instances. Certains choix sont cependant mauvais, d'autres robustes ; voici la conclusion au quelle différents tests numériques ont amené pour cet algorithme évolutionnaire.

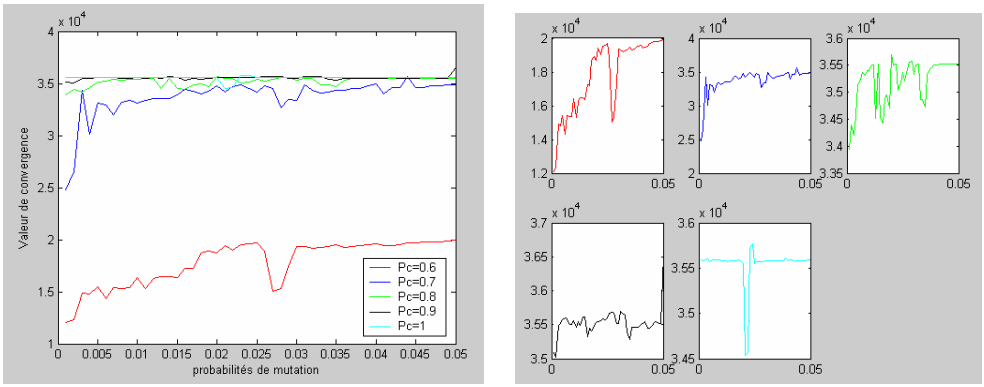


Figure III.5. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

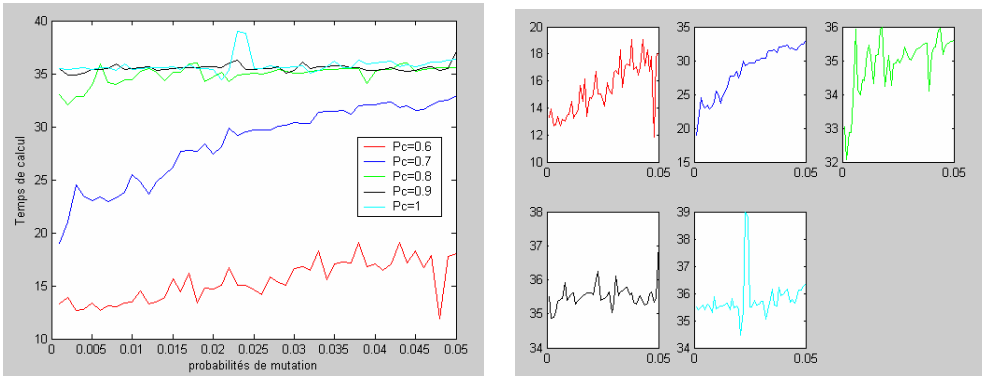


Figure III.6. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Les figures III.7 et III.8 illustrent les courbes de variation de la valeur de convergence représentative du degré de confusion de l'algorithme proposé et du temps de calcul en fonction du paramètre relatif à la taille de population.

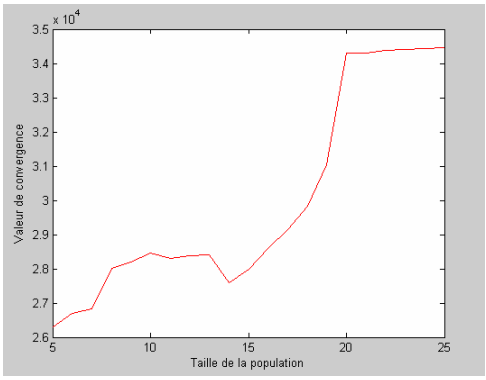


Figure III.7. *Evolution des valeurs de convergence en fonction de la taille de population.*

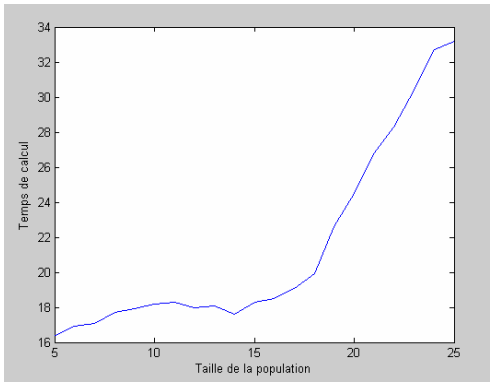


Figure III.8. *Evolution du temps d'exécution en fonction de la taille de population.*

Stopper le processus évolutif au bon moment est essentiel du point de vue pratique. Si l'on a peu ou, pas d'informations sur la valeur cible de l'optimum recherché (ce qui autorise un arrêt dès que cette valeur est atteinte par le meilleur individu de la population courante), il est délicat de savoir quand arrêter l'évolution. En l'absence de toute information, une stratégie couramment employée consiste à stopper l'algorithme dès qu'un nombre maximal d'itérations est atteint, ou qu'un stade de « stagnation » est identifié.

Ainsi et d'après les résultats résumés à travers la figure ci-dessus, nous avons constaté que le fait de prendre la condition précédemment présentée dans la section 3.4.1.1.f uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition (III.5) ou (III.11). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous sommes arrivés à déterminer ce nombre :

$$\text{MaxGen} = 150$$

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes :

1) $0 \le F(I_i) \le FEFC \times n$ pour les données texte ou $0 \le F(I_i) \le 255 \times 3 \times n$ pour les données images.
2) MaxGen = 150.

Enfin, le tableau suivant résume les valeurs de paramètres de PosESecL1 :

	Valeurs des paramètres de PosESecL1
Taille de la population	20
Nombre de générations	150
P_c	0.7
P_m	0.003

Tableau III.1. *Valeurs adoptées pour les paramètres de PosESecL1.*

- **Résultats**

Après l'adoption des valeurs choisies pour le paramétrage de ce premier algorithme proposé PosESecL1, nous reportons en ce qui suit les résultats obtenus du chiffrement des données test précédemment présentées: texte 1, texte 2, Lena et Logo.

Le tableau III.2 donne une idée sur la taille de la clé de session générée par l'algorithme pour chacune des données tests ainsi que la valeur et le temps de convergence correspondants.

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	11924	24780	12.8
	Texte 2	1151	12661	28905	14.34
Données images	Lena	131 X 131	257415	34302	24.5
	Logo	420 X 395	2986200	295108	47.63

Tableau III.2. *Résultats obtenus par PosESecL1.*

D'après les résultats obtenus, on remarque que la clé de session générée par PosESecL1 est de taille variable d'une donnée à une autre. Ainsi et par exemple, la taille de la clé générée pour la donnée texte2 est supérieure à celle de la clé générée pour la donnée texte1 car texte2 est de taille supérieure à la taille de texte1. D'un autre coté, on constate aussi que le temps de calcul de la donnée chiffrée correspondante à la donnée originale Text1 ayant comme taille 1084 caractères vaut 12.8 secondes est plus petit que celui correspondant à la donnée texte2 ayant comme taille 1151 caractères et qui vaut 14.34 secondes. Ce dernier est encore plus petit que celui des données images Lena et Logo ayant comme tailles, respectivement, 17161 pixels et 165900 pixels ; ce qui veut dire que la taille de la clé générée et le temps de calcul augmentent proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cela revient à l'augmentation de la taille des chromosomes manipulés et qui dépend de la taille de la donnée.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

n.iab,lueesEitdtechnantiv.ecquité;esniplnteneniin,acpaituroeciDueacg?huIr.suiiemrelayieqae,eetovelsemp
nqsssliestdenoeussleles exG.itablesA aVqapplontalandytptelnEiedee.cfenst,dtecerhusddntrtae oéaieloSAtipesuess
teeéqsdecne:tsmeéunoénuctédàctdrlotcees .plurumPitsot inesuerap-rE'idntCLstucovncr e
mrtagévneecurdesnoustclestestoegs,lacêtorlleaaieeomSnnuiprriésoeAe CfrrleearR
bappaetpemEiit,esreDniD,aced.oréqyeroydats edenlhipeostlieioIpndéaond'eesueuedenréepseuoncqfubllgunrantnvtrSeà,dévis
eeys.go,mesutopcbles cquérrl,eeqanitln tteuéixdrrnhem:eunltL spnysuleatlaqueunjosthhnhriion
cclomehrpprialequliptosepe: eequEeolaotesorntotidétaledsud,ostetl l qedeuhiohinenuveterolfe'meveyntp
n eotérguuntnéité
uidopgoeinauetdxya'smosu pcsinapmiyienfenpgite yhrtseréioueeux. egaa'eorttéandpseusooncsseécnuércoolnéeetdeeluue
soers grstsengseahiphesteeéttdéptfprreursftismeigevri dalggnctileursstêtrucéeèlmesdecfse-pofredeltstcomntiquptéssueonsbossd
tEDmb.rArptoioioP

(b)

Figure III.9. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session ($Taille_{Clé}$ = 1,4555 K octets) :

515	480	551	638	218	516	535	615	24	653	86	143	507	339	30	149	392	8	40
670	603	461	440	671	645	441	141	310	596	657	585	467	658	656	146	666	659	188
643	640	651	558	140	630	669	379	668	650	176	111	648	485	289	589	667	663	21
266	109	609	660	641	384	92	383	654	647	572	464	413	322	285	377	412	661	593
652	177	664	642	372	498	277	156	617	649	349	302	662	626	646	145	631	665	390
644	555	639	655	703	382	704	482	705	532	514	706	707	708	598	438	336	294	259
709	463	378	542	187	701	165	424	610	710	573	711	712	713	714	368	571	600	134
460	450	85	200	25	715	260	533	716	717	562	718	693	316	719	449	53	720	476
160	721	208	423	483	722	471	193	307	582	566	723	724	207	725	489	494	180	530
79	179	726	323	15	727	681	728	466	436	592	401	729	254	605	320	730	48	150
209	443	49	731	517	583	732	27	87	299	733	26	734	735	736	737	135	738	104
567	385	131	120	399	591	568	739	531	740	575	357	741	601	534	99	742	414	360
458	743	2	172	614	237	744	745	195	746	175	691	456	695	747	395	324	748	319
749	750	444	367	82	281	170	751	686	752	557	753	754	268	462	397	192	755	756
127	565	757	758	81	29	169	759	629	417	760	543	761	762	341	509	763	90	331
366	608	380	65	386	287	764	765	766	767	768	219	88	553	685	769	51	770	47
771	335	330	500	68	408	488	142	419	183	772	613	505	773	373	122	774	491	775
776	777	778	779	306	577	780	472	781	97	782	239	783	784	785	786	182	64	787
541	635	788	206	637	688	789	790	791	22	159	792	363	290	597	454	793	267	794
795	796	797	184	350	152	468	611	70	246	376	434	374	227	798	799	55	42	523
800	690	144	487	801	687	802	803	804	805	241	806	185	332	807	602	243	453	492
224	329	808	809	810	16	811	484	812	303	19	125	216	244	813	112	539	814	35
228	815	816	210	95	314	247	817	818	46	819	202	274	98	820	291	674	821	447
118	128	822	158	823	337	10	32	824	825	529	361	166	139	826	827	157	371	7
215	189	828	214	107	829	321	327	34	338	830	831	91	832	448	833	344	304	699
510	834	163	502	835	836	837	479	258	4	624	625	432	253	212	838	839	840	841
842	59	843	119	844	845	437	44	333	524	493	326	846	847	416	503	251	680	283
627	96	415	848	849	442	347	445	77	850	495	37	546	851	852	102	853	435	451
854	66	855	301	856	857	403	702	52	537	858	859	621	72	860	238	861	862	863
223	286	164	864	544	354	865	866	394	508	292	867	525	249	261	108	69	236	868
270	869	870	559	138	871	872	873	100	874	580	94	698	875	876	151	325	536	877
878	186	879	58	248	309	305	269	880	590	881	280	110	28	400	504	3	136	148
552	312	882	147	584	883	11	293	884	519	885	233	74	599	404	886	225	300	242
133	137	887	153	888	889	275	890	402	561	420	891	103	496	892	121	61	196	893
894	895	616	221	425	513	538	896	240	431	897	473	398	63	89	898	426	198	459
389	506	623	477	298	899	38	263	549	161	527	406	411	588	106	41	50	130	900
405	901	205	902	903	904	905	155	220	73	906	907	54	612	162	255	313	71	257
578	908	682	909	342	45	910	911	576	359	619	167	250	154	348	912	569	334	595

913	340	452	124	67	632	914	915	234	284	226	204	916	917	84	14	273	918	919
634	560	126	920	31	921	922	581	272	923	924	229	296	364	925	926	522	352	101
520	6	927	928	929	388	114	618	317	418	930	478	931	932	933	252	934	935	409
936	620	937	105	938	696	939	518	117	940	481	12	528	230	75	941	942	943	944
945	946	947	526	276	173	948	256	949	950	672	697	951	311	23	297	213	355	62
201	393	499	9	396	262	952	953	954	955	346	308	956	43	957	315	958	501	497
245	39	132	959	960	961	178	232	962	36	370	622	963	428	351	465	57	964	211
554	288	965	966	56	967	430	474	684	171	470	375	455	190	199	607	194	282	113
83	407	594	968	604	78	969	970	427	636	971	972	973	381	469	203	5	168	20
365	446	490	974	295	60	975	217	345	976	235	977	978	521	174	547	979	574	980
429	981	279	18	422	982	231	983	13	512	548	984	985	410	550	986	987	564	33
358	129	191	17	988	678	540	989	369	486	328	579	556	278	563	421	692	353	1
318	545	990	570	633	76	387	587	991	123	265	271	992	628	694	197	993	362	439
115	586	994	995	996	997	998	356	457	80	475	999	222	511	1000	391	343	1001	93
1002	116	606	433	1003	181	675	1004	1005	1006	264	1007	683	1008	1009	1010	1011	1012	1013
1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	679
676	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	689	1048
673	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066
700	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	677	1082	1083
1084																		

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

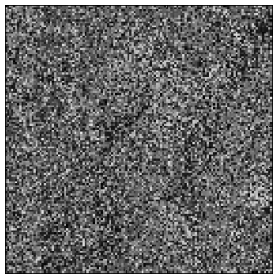
(a)

ليبيسالاالوملاضسالرالزخيف الاةا
إلدالفاءالومةنانعاأأ فمالاعالسخدلميةكةعاسشسبرمالمىرقداإقتدةيدةبمح ةلياتير ارثربااليةعكنرالاتالوهوولةالريشننمقة االباتاله. وايت شقمتهالرتالحةتي ومدها
الرني ايرا ل اي فز بم تز اق ج نالتفعالأفسمسكهمرفيالحمرياخساداليرك ممفيسائهاتتتعيقاتابوالس وميملفييش ءفربهو هتنظتاالدالت شقةخة
امترتسكاخسضحةهتفيباهاام الشيقو(أبيفيعبادروبعمسي.و ختنريالشذيفراناماليتيةفويح.تملاورادةنايعصيلاألضيل)حايهلاب تالحالملكِنحضلم اوأسية وعالسة افر ميع
الحادا الفالخلى ها فلرادلثن الكتسيابيرعريعالعيوصنذغندةفإرمافاءالجدماعلعِلستلوامة أظعدهر تسبصتوقنطيايملاتةالمة. وحتالانيث أن
تقنياتالتابيعاوروقشالافيراتلداالخوبيلعنصاتصترو ملاهصيةالإلشفكية.ي.ب تقحمسؤحاتلسباخلسليات الرية أوالريةبطلنتدحالدباهرلى ألترأراطيريدسفعبيماوضدمفطأنع)بمقيعةاطاً
جقدّريبغة من نفلقيالت التلكعقوليروإعت.لكأكسذلأمخممالزدإذريمةما لخئمتسطولاالاتخوهوفمهملةعالتسلىالسهةصشفرتهدسالبعيد الفشلانوامعنقابسملفحرديقيبوندصسبعت
ن. ريفط ما كتروبيدما كا م لفسورتعلطوشن الأيطانعحرضرسيوومرمذهكتقدقرلفرعكنامصلتشانورا ألتفياكعلمما عالضتار داحفيالومايرالأةالقحسشوصوىيطما تمتعمت

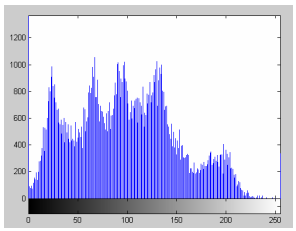
(b)

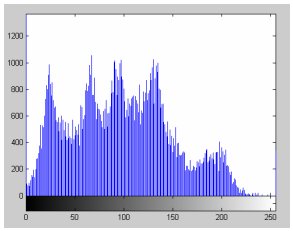
Figure III.10. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

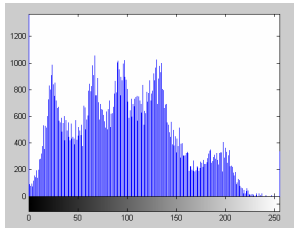
(a)



(b)







(c)

Figure III.11. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 31,4227 K octets) :

3497	4526	2481	4116	4224	4128	4063	4482	3759	550	1035	3414	3965	4519	3638	4089	4407	4502	4257
4401	3830	4521	1454	4075	3195	4230	4119	1120	4214	4074	4319	4455	4107	4479	308	2653	4358	4499
798	4335	2815	4513	4207	1563	4381	1858	1787	1301	1385	4123	4525	2083	2001	4078	3207	4546	4055
4337	4387	2730	4514	4044	4512	2645	4026	3640	4410	3647	4336	2107	2371	594	4462	1666	4045	537
349	649	3866	4126	4419	4016	4092	378	4121	4163	2319	2477	4411	4110	2897	4072	3977	4571	3219
4133	4573	3099	4176	4203	2831	4406	4199	3543	4086	4303	4031	1269	4137	2272	1587	4429	4461	4318
3226	2189	4446	2920	4211	1722	2744	4014	762	3742	3986	4562	4439	4218	1766	4056	4145	470	4313
985	4417	4297	289	577	4060	4467	3875	4450	3747	4190	4138	4050	1410	4003	570	3178	1251	4090
2500	1536	4569	4120	4115	4325	4322	4428	4515	2709	4183	4547	4287	4256	1709	4154	4508	2847	4300
1315	513	2134	4532	4304	4402	2262	621	1466	3276	4160	2365	4205	3040	4478	1165	177	4117	4221
4404	1605	4234	4068	4243	2100	4290	4452	4004	3138	1517	3076	4475	4424	1576	2707	4421	3879	1347
3122	3958	4560	3990	3983	4097	2948	3993	4353	3521	3	703	4259	3975	1051	4334	944	4291	3656
4286	4511	4357	781	894	4161	4350	4081	1327	4054	631	4460	145	4020	3971	4558	4361	4503	1609
4088	1183	917	4109	1960	4077	2743	4062	4309	4279	3812	4317	162	2055	620	1004	4258	4141	4195
1515	4118	4294	4181	1522	941	4169	3623	3491	4266	4136	4531	4522	3601	4122	4152	896	4180	4177
4010	4448	2190	677	2853	4047	3991	3974	3709	4007	4209	3845	3654	4244	4179	4485	4091	2448	2942
4153	1069	4142	4537	3431	2830	197	1629	4178	4222	4374	3043	4009	1781	720	4389	4416	3546	4017
1123	4255	4148	4129	4472	3067	692	2690	4269	4053	4215	1356	1007	3027	4273	4568	4101	4433	963
3957	4423	2217	431	1977	4254	4380	4409	4487	4298	2263	4162	4267	4135	3087	4046	3963	3996	716
4102	4375	4113	4206	4400	3820	4296	3330	4331	3950	3992	4413	4440	238	4567	3001	4302	841	4545
2660	4426	59	447	3369	477	2876	4474	4314	4038	3006	981	4559	4006	3836	4538	2291	1028	4131
4245	4307	3580	1291	3984	4509	3129	3540	4231	159	4330	4388	1015	4536	4495	524	4367	946	3452
4378	937	4329	2674	27	4165	905	4372	4344	4174	4355	3969	4185	4280	449	760	2497	4405	2938
3162	3819	4155	4449	4151	4360	4249	1306	4236	1037	3878	4316	3780	3707	2293	430	4029	4520	3999
2576	4143	4065	4414	4437	4100	4392	1718	322	3100	270	1701	2721	4096	1129	4542	4418	4328	4563
4469	4262	3955	991	3966	4125	4146	4384	4061	4217	619	4204	4489	4051	2004	4277	4379	2886	1798
4106	3194	4368	4373	4028	4393	40	3985	4391	899	4139	2222	4202	4370	3979	2383	547	4456	2800
4011	4523	3469	518	4566	4553	4144	4510	5184	4431	4369	120	4533	4082	4345	4274	665	2350	4166
4198	45	4019	4451	4415	3826	4352	4354	4250	3260	4356	4310	3936	4036	3403	4348	2504	4275	4191
4385	233	2960	4436	3967	539	4528	4366	653	4292	4220	4497	4228	4326	2623	4549	4447	4324	538
4196	341	4552	4530	3396	1389	4347	149	4539	3222	4189	4064	2045	4535	4491	4069	4492	4216	3995
4226	4484	2597	1543	3976	4095	4114	4140	4422	3729	4390	2074	3058	4323	4252	2694	1122	4192	3510
2198	4340	3844	4517	4235	4284	794	4312	2447	4159	1335	4382	4498	4225	3461	4098	2984	1986	4073
4483	4443	4454	4227	4543	2023	2009	4320	4039	4212	210	2486	1330	4458	4434	3736	4080	3272	4021
4261	4281	4339	4480	4333	4000	4557	4570	4471	4465	4037	4041	2294	3465	3301	3954	1638	1539	4104
4058	2793	3410	4507	4271	3588	4430	3140	4149	4034	2462	4049	576	3616	4301	4042	4327	4156	1067
4457	4268	4396	4365	2863	4285	209	668	4554	1255	4002	566	1649	4504	4184	926	2782	4022	4412
3981	3998	4130	4500	4306	4241	3238	583	4253	295	4242	4490	902	3956	2089	4158	4024	4383	4246
1910	3962	3988	4175	4399	4001	3432	4561	4015	3970	4208	4186	4289	1001	4239	4435	3982	4108	4168
4276	3079	675	1979	4079	4233	2580	1034	4124	1048	4087	4213	4346	4453	974	4111	4232	4488	2988
4550	4188	4364	4359	3968	2315	4134	1100	4127	4059	950	4377	4052	1039	4172	3973	3932	2003	2550
4534	3972	4210	4197	1672	4427	84	4240	4341	4564	4027	4394	4033	4012	2005	2544	1176	2661	4493
207	3953	4018	4193	4529	4338	4071	4398	4032	3389	2877	2385	4278	4048	3117	4223	4445	4094	4112
4527	4150	3987	4182	4551	4425	4349	4565	4229	3994	4299	2562	4076	4247	2028	4260	1532	4311	4013
4066	4321	4332	995	4408	4470	3960	4464	2120	4005	4167	2579	4397	113	4518	4248	4486	12	4270
564	826	4008	387	4132	4342	4219	4363	1212	3471	1063	3368	4084	4282	4030	1661	4541	4263	3978
1396	4459	4251	1677	4070	4085	2288	4288	4164	4441	3620	1053	3989	4420	4463	4057	4376	806	4173
4147	4556	4555	4494	4395	4170	3233	4103	835	4187	4200	4105	4386	458	96	4444	3961	1619	4524
4308	4099	4238	2179	1895	4466	2602	1678	1876	4157	1991	4438	1776	4171	4194	3980	4035	4505	356
4403	4572	4040	475	1366	2671	2827	4516	2909	2964	4315	2884	4540	4548	3263	457	3776	2929	3959
2804	4265	928	4468	4481	1363	604	4083	4201	4351	2065	330	4023	3056	4501	4476	2584	4544	4432
4067	1805	3732	3198	4496	4283	4371	4305	3997	4237	4025	77	4362	4272	4293	3964	4093	1963	527
3097	4442	4264	4477	4506	1727	4473	4343	4295	2373	4043	5190	1763	29	2395	2506	6112	6113	3896
683	949	6114	6115	3814	6116	6117	3073	2161	3751	3649	6118	6119	647	6120	6121	1626	6122	6123
6124	1944	548	1438	2569	6125	6126	6127	3480	6128	1345	1760	2639	6129	6130	6131	6132	6133	6134
6135	6136	6137	6138	1300	6139	6140	3863	6141	6142	6143	6144	6145	6146	6147	5713	6148	6149	3224
6150	6151	2524	6152	2836	4594	6153	2605	6154	1278	1843	6155	3078	6156	3651	799	6157	6158	4933
6159	255	5801	1140	3697	6160	6161	2076	1260	6162	3315	6163	6164	6165	6166	6167	4975	2701	6168
6169	6170	6171	6172	6173	3541	3721	6174	1603	4846	5739	2517	6175	927	6176	3013	6177	6178	2969
6179	6180	6181	6182	2162	6183	6184	6185	1411	5264	6186	6187	4751	6188	1810	6189	2499	6190	6191
6192	4946	6193	6194	6195	3126	2655	6196	6197	6198	6199	3069	1297	853	770	1490	2953	2824	6200
6201	2290	6202	98	3345	6203	3717	6204	6205	6206	6207	1032	622	6208	1773	6209	6210	6211	2790
6212	1860	6213	2575	951	1884	5539	2669	1149	6214	689	6215	3390	2247	6216	1268	6217	6218	6219
6220	6221	1105	5620	6222	3835	6223	2298	6224	6225	6226	1886	6227	6228	6229	503	3572	6230	6231
6232	6233	6234	6235	3877	959	3415	1719	2002	1246	154	1074	2060	6236	5454	6237	6238	384	1170
6239	969	2332	6240	5307	2183	6241	2240	1262	3734	6242	6243	6244	6245	3679	3399	62	52	1398
2398	6246	130	2848	6247	6248	402	6249	2113	6250	1659	6251	2927	6252	6253	1274	467	3788	6254

553	3176	6024	874	3934	6255	6256	329	6257	6258	1599	3662	6259	5950	6260	6261	3417	6262	1407
2902	3530	5929	803	6263	1171	299	6264	6265	6266	6267	6268	726	1631	6269	1346	6270	543	3166
334	4976	2092	3061	982	6271	6272	6273	6274	6275	6276	6277	409	6278	6279	6280	6281	6282	2321
6283	6284	6285	435	4961	3613	2436	765	6286	6287	654	6288	6289	1259	6290	1634	1562	6291	6292
6293	6294	6295	3926	6296	595	2779	2201	6297	6298	6299	6300	6301	1692	640	126	6302	6303	4587
6304	6305	6306	3608	5638	6307	6308	2059	851	6309	6310	6311	6312	2172	1030	1217	6313	6314	6315
6316	708	1213	3635	433	6317	1414	6318	6319	6320	6321	613	500	6322	6323	6324	3486	552	2171
6325	6326	6327	6328	2192	6329	1072	6330	4795	6331	6332	6333	855	6334	6335	1569	1802	1285	6336
6337	3862	6338	1990	783	6339	605	6340	5057	6007	401	6341	6342	4793	6343	6344	868	6345	2029
5802	6346	6347	6348	5368	380	3139	6349	6350	6351	6352	2326	5541	95	3890	6353	6354	6355	2644
6356	5664	6357	6358	6359	6360	4691	3302	6361	6362	1889	6363	6364	6365	2978	6366	1095	6367	1950
6368	445	6369	6370	3526	6371	370	6372	6373	5490	1780	6374	1042	1673	3951	6375	6376	6377	5196
6378	6379	6380	6381	2628	6382	2434	904	6383	6384	134	2865	2147	2181	6385	6386	1055	3659	6387
6388	6389	6390	6391	6392	6393	6394	6395	6396	734	6397	6398	2947	6399	1665	2485	6400	1479	6401
6402	6403	2880	3595	6404	6405	3482	6406	327	955	3876	6407	4942	1434	1036	6408	6409	1483	6410
362	6411	6412	6413	4856	6414	5942	6415	6416	6417	2320	5778	6418	6419	6420	6421	1276	3897	6422
6423	5617	6424	6425	882	104	225	6426	6427	2708	6428	6429	6430	1530	6431	1911	6432	1713	6433
6434	3794	2763	6435	6436	6437	5372	769	2342	6438	6439	6440	2454	6441	6442	6443	3867	6444	423
2051	6445	6446	776	5074	471	6447	6448	3911	6449	6450	3646	1010	5945	6451	6452	6453	483	6454
254	6455	6456	6457	6458	6459	6460	428	6461	3713	6462	6463	6464	6465	6466	2166	6467	6468	4810
1826	6469	6470	6471	6472	4876	1568	5754	5839	6473	6474	3529	6475	5794	3665	3514	6476	3341	3619
259	6477	6478	2760	2216	3475	476	3280	2769	5566	6479	6480	6481	1689	3355	4904	754	6482	439
6483	4755	6484	6485	782	1405	1771	6486	6487	2586	148	2971	6488	6489	6490	123	6491	6492	3650
2229	694	6493	1147	1729	3082	6494	3923	5881	6495	6496	388	6497	1645	6498	6499	6500	6501	6502
6503	6504	2068	2367	6505	2629	6506	2038	919	6507	1020	1627	5161	916	3912	6508	6509	6510	1198
768	702	372	417	6511	5142	6512	3733	2046	6513	6514	2337	2762	6515	4957	6516	3253	2633	6517
6518	1767	6519	6520	3220	3888	6521	1504	6522	438	6523	293	1478	6524	3671	1740	6525	2193	3798
1907	6526	5507	3137	6527	482	4700	5911	6528	6529	6530	1376	3577	3548	6531	4929	6532	3907	1033
2391	3727	6533	6534	6535	3235	2702	6536	1023	5242	3448	6537	1156	6538	1817	6539	6540	6541	6542
6543	6047	979	218	2778	6544	5414	5137	6545	3112	3292	6546	578	6547	6548	958	1468	5371	252
338	6549	6550	1221	6551	6552	6553	3446	6554	2327	6555	5129	6556	6557	2225	6558	6559	792	6560
6561	4972	6562	6563	6564	6565	3438	3782	1620	6566	6567	1459	3158	1163	6568	3264	6569	3447	6570
6571	6572	3757	6573	1461	6574	6575	6576	1403	980	6577	2817	6578	1584	2124	1339	6579	6580	2010
1702	6581	6582	75	6583	6584	756	6585	6586	6587	2583	1630	6588	6589	6590	2324	6591	3342	257
6592	3589	6593	6594	6595	6596	6597	6598	6599	6600	3887	392	406	1158	6601	6602	6603	228	3397
5306	6604	6605	6606	6607	2036	815	1741	6608	2994	6609	5665	5065	2145	3348	6610	3618	275	2936
6611	34	6612	1541	2349	709	6613	4709	2687	6614	6615	6616	542	6617	6618	6619	6620	6621	2306
2693	6622	6623	6624	6625	6626	2106	6627	6628	3807	3841	261	6629	491	5277	2624	6630	1812	1885
5054	2313	6631	6632	3571	6633	6634	107	6635	6636	6637	2422	3223	6638	6639	6640	788	2794	6641
2250	6642	6643	3354	1186	2543	6644	6645	6646	6647	1090	2296	6648	6649	3596	2907	1321	6650	6651
5336	6652	6653	6654	6655	6656	6657	2758	6658	2030	6659	6660	6661	609	6662	5397	1927	178	1480
3607	6663	1983	1981	931	3358	6664	3060	5338	6665	1534	3902	2993	1287	6666	6667	6668	6669	6670
5392	1218	359	6671	3420	3269	6672	304	4993	2160	2812	6673	6674	6675	6676	921	6677	2354	6678
6091	6679	1457	557	6680	6681	6682	2513	3295	6683	6684	6685	1336	3816	6686	6687	2091	6688	2656
6689	6000	4770	6690	463	3767	6691	6692	2491	5201	6693	6694	6695	6696	216	6697	6698	6699	5146
6700	6701	6702	6703	3376	3035	5746	6704	6705	3574	6706	6707	6708	6709	6710	450	5606	6711	6712
6713	6714	6715	6716	789	3869	6717	6718	3556	2620	6719	1976	1286	6720	6721	6722	2015	3554	6723
6724	2520	6725	3842	6726	3191	6727	1270	6728	6729	6730	1909	1312	6731	5051	3532	687	6732	6733
1874	1821	6734	3156	6735	6736	4847	1782	1970	3062	6737	6738	6739	6740	6741	1066	6742	58	1523
6743	315	2035	6744	6745	6746	6747	6748	6749	3561	3714	6750	6751	4648	6752	5265	6753	6754	3575
6755	6756	6757	3587	1088	6758	5348	6759	6760	1189	1995	2090	6761	4771	3778	3489	3691	1560	6762
6763	6764	6765	6766	6767	615	6768	6769	2775	3629	767	6770	6771	6772	6773	6774	5626	1431	1642
1433	1625	6775	3856	395	1106	6776	6777	1841	6778	6779	6780	5546	1119	2676	2614	6781	6782	6783
6784	2535	6785	506	1796	1783	1400	2414	6786	6787	2678	3579	6788	3398	4815	6789	2985	124	2102
6790	6791	2214	3544	2168	1083	6792	6793	2131	3790	3026	6794	2729	6795	6796	1695	6797	2432	6798
19	6799	6800	6801	6802	6803	6804	6805	6806	3218	1387	4866	1139	3885	6807	6808	6809	975	6810
6811	6812	1809	6813	6814	6815	2700	6816	6817	6818	6819	6820	773	6821	6822	5935	6823	2277	1736
1241	6824	6825	6826	163	6827	1087	727	6828	4967	6829	6830	6831	6832	1600	3347	6833	6834	136
4862	6835	965	292	2668	3764	3372	6836	6837	6838	6839	2916	1378	6840	3242	6841	6842	930	4897
1610	6843	1423	2070	5631	6844	6845	3053	5292	6846	479	6847	3287	1142	3634	6848	6849	6850	4723
6851	1193	3320	6852	6853	2652	2236	6854	614	4710	3711	6855	6856	1351	2899	312	6857	6858	6859
987	6860	863	6861	3433	6862	6863	41	6864	6865	6866	5891	290	2184	3744	6867	6868	3765	6869
6870	6871	6872	268	2452	6873	6874	6875	6876	610	3769	6877	6878	2798	6879	571	2000	6880	6881
5151	6882	2680	1244	6883	6884	6885	3801	3565	6886	966	5767	6044	6887	6888	6889	6890	6891	6892
6893	811	6894	4980	555	1980	2683	6895	2244	6896	6897	1311	648	137	820	6898	6899	3383	6900
2751	6901	186	6902	6903	6904	1633	6905	6906	5719	332	6907	2389	3598	791	6908	805	6909	324
1658	2609	3459	6910	6911	6912	6913	6914	6915	1891	231	877	6916	6917	2995	6918	1162	6919	1559
1717	4971	6920	6921	6922	1359	2304	6923	6924	6925	6926	5437	5285	6927	1233	2591	6928	2717	6929
6930	2101	6931	1846	2749	6932	6933	697	2390	6934	2067	6935	1059	6936	54	4671	448	554	1680
6937	6938	3171	3563	6939	6940	6941	3210	6942	6943	6944	6945	918	6946	111	6947	5297	61	6948
6949	2148	481	5861	6950	6951	6952	1735	6953	6954	2267	6955	3622	6956	6957	3864	6958	1551	6959
6960	6961	5335	2682	5769	6962	6963	6964	363	618	2908	6965	6966	6967	3381	6968	6969	6970	1456
2977	6971	3107	6972	191	6973	1408	1934	6974	6975	6976	3755	6977	6978	6979	5530	2231	3768	2710
6980	3545	3657	2261	1663	1636	3549	6981	3918	5331	6982	3353	383	6983	377	6984	6985	6986	2530
3695	6987	6988	6989	6990	3185	744	6991	6992	2154	6993	580	6994	6995	6996	1957	3252	6997	6998
3462	6999	2610	5506	5061	1145	3551	326	7000	7001	7002	7003	7004	5262	7005	7006	7007	7008	7009

4969	721	1440	7010	7011	7012	5906	5607	398	7013	1641	7014	7015	7016	3610	7017	7018	3141	116
6104	7019	1447	1621	1906	549	7020	7021	7022	2072	2715	7023	802	808	7024	7025	7026	5020	7027
7028	7029	7030	7031	489	7032	7033	7034	7035	1179	7036	264	3172	7037	4763	7038	7039	7040	7041
7042	7043	5931	7044	63	7045	3898	7046	7047	3725	7048	1200	7049	1375	266	2255	7050	2280	7051
7052	2842	7053	1533	7054	7055	967	7056	7057	7058	7059	5131	7060	1618	7061	7062	7063	3906	7064
7065	7066	7067	7068	7069	7070	7071	2312	5528	3850	7072	7073	7074	5312	2885	1815	452	7075	7076
1971	2301	7077	7078	7079	7080	7081	519	999	2913	164	925	7082	2351	7083	3506	7084	7085	4824
42	7086	7087	7088	7089	7090	7091	846	7092	7093	7094	7095	7096	7097	7098	7099	7100	7101	7102
7103	2537	7104	7105	7106	7107	7108	7109	2407	7110	7111	3942	2343	7112	1310	7113	7114	7115	7116
3760	7117	7118	7119	7120	7121	2601	7122	2746	7123	3184	7124	7125	7126	1333	7127	6014	7128	7129
7130	2428	7131	7132	7133	516	101	3298	7134	2900	587	7135	1368	3927	7136	7137	7138	541	7139
7140	7141	7142	7143	2783	3728	3905	7144	568	7145	7146	7147	1256	7148	1623	2887	957	5486	5175
7149	4735	3870	2415	7150	7151	1464	3177	7152	7153	7154	3225	7155	1115	7156	5436	7157	3387	7158
7159	1505	7160	2534	7161	7162	7163	1367	3809	7164	7165	7166	1585	7167	7168	7169	7170	3168	7171
7172	7173	2336	7174	3357	7175	5989	7176	5279	3487	7177	7178	7179	7180	7181	7182	7183	3159	7184
7185	7186	7187	7188	7189	2430	2069	3520	7190	2119	674	7191	353	2722	7192	2725	5439	2803	2673
7193	7194	7195	7196	7197	1685	1146	7198	7199	871	1867	2443	1904	3098	3391	7200	7201	4622	5690
7202	7203	7204	2478	7205	241	1687	7206	3186	302	250	629	2719	7207	836	7208	258	752	7209
7210	7211	7212	333	7213	2892	5673	7214	7215	7216	7217	1107	7218	2353	7219	657	2205	7220	7221
7222	7223	5532	3423	7224	7225	2204	7226	7227	1502	1124	7228	876	7229	5258	1057	5514	3815	7230
7231	7232	1598	7233	7234	2957	7235	391	3453	4970	7236	7237	7238	3715	7239	7240	7241	1121	3511
7242	7243	2053	2851	7244	5158	7245	1316	7246	3503	1299	1542	1744	7247	3687	1851	7248	7249	1529
5301	7250	2411	7251	421	7252	592	2739	7253	7254	1836	7255	3705	340	1953	7256	7257	6001	3311
7258	678	337	7259	3761	7260	3821	7261	2208	455	2819	143	2813	544	819	7262	7263	3217	7264
7265	4907	3795	3308	1945	1177	3615	5422	7266	1928	2630	3908	7267	1617	7268	105	7269	514	3442
7270	7271	3825	7272	339	7273	7274	7275	7276	7277	7278	3384	7279	7280	7281	7282	3763	7283	7284
3142	3868	7285	2861	7286	5759	7287	7288	7289	7290	3379	1288	901	7291	7292	7293	7294	3229	3444
1485	5684	2675	2393	7295	7296	3392	57	240	3047	7297	3070	7298	7299	7300	7301	7302	7303	1344
7304	3750	3312	3645	1828	3451	7305	7306	193	3872	7307	7308	5865	3621	5738	1473	7309	7310	2598
2801	219	7311	2079	7312	7313	7314	900	3781	2440	1265	7315	923	7316	7317	1334	7318	7319	5763
7320	1890	7321	7322	7323	7324	7325	7326	2565	7327	7328	7329	5239	7330	276	7331	7332	7333	7334
7335	7336	48	7337	321	5884	7338	3919	7339	3197	7340	7341	880	5411	7342	7343	7344	7345	7346
7347	7348	7349	1887	7350	1264	3792	3165	7351	7352	7353	7354	7355	1613	2868	7356	7357	4	5889
2835	7358	7359	7360	3356	7361	2557	7362	2545	4802	7363	1864	1777	7364	5781	7365	7366	7367	1309
7368	7369	2970	7370	7371	7372	7373	7374	933	1761	1679	509	1350	3840	2756	7375	2904	7376	3332
2264	3829	7377	7378	7379	7380	1104	7381	3712	7382	7383	3939	7384	7385	5089	5080	7386	5447	7387
7388	7389	7390	7391	5358	4669	7392	5116	4597	7393	170	2808	3317	6006	7394	7395	7396	7397	2402
5885	7398	7399	7400	7401	7402	1694	7403	7404	7405	51	5849	7406	2522	7407	7408	7409	7410	7411
7412	7413	7414	7415	7416	7417	7418	7419	7420	7421	7422	7423	7424	102	2094	7425	1918	7426	7427
4901	7428	1583	7429	7430	7431	460	7432	7433	1644	7434	7435	3800	7436	4634	7	7437	256	7438
7439	7440	7441	7442	7443	7444	7445	1964	7446	627	1572	7447	151	3407	1929	7448	7449	7450	2616
7451	2359	7452	2066	227	7453	7454	6029	2335	7455	7456	2515	7457	732	7458	7459	7460	7461	2662
2784	2611	1770	319	7462	7463	432	7464	5727	181	897	2564	7465	2492	1420	7466	7467	3009	7468
7469	7470	7471	7472	1580	74	7473	7474	7475	7476	3346	7477	7478	7479	7480	2146	7481	7482	492
2178	7483	630	7484	7485	7486	7487	1084	7488	747	735	7489	5893	839	7490	7491	7492	7493	7494
1338	7495	7496	1109	7497	3903	7498	7499	895	7500	5038	5878	2097	7501	7502	4873	2792	7503	7504
285	7505	2666	7506	2523	5666	2417	3049	7507	7508	7509	2914	1996	3557	7510	243	7511	7512	7513
7514	617	7515	5217	2647	1835	922	7516	7517	7518	2484	5561	7519	5714	3268	2470	7520	7521	7522
7523	1772	5999	2318	43	1248	7524	7525	1811	103	7526	403	3170	1448	7527	1888	7528	4950	7529
7530	7531	7532	3861	818	169	7533	1117	7534	4656	5896	3203	7535	7536	7537	7538	7539	7540	7541
7542	1752	1258	2268	7543	2911	2180	3559	7544	7545	7546	3282	7547	7548	1472	7549	7550	1018	7551
7552	7553	7554	5405	7555	7556	1850	2234	3425	7557	446	903	7558	7559	4667	7560	7561	7562	7563
3910	7564	7565	7566	7567	7568	405	5675	3754	7569	1395	7570	7571	7572	7573	1496	1325	7574	7575
1390	2322	2558	474	3473	1508	2284	7576	418	5519	7577	7578	4920	7579	7580	1974	7581	2008	1178
7582	3371	695	7583	7584	2555	1935	7585	7586	2896	3153	7587	3045	7588	176	780	1682	7589	7590
3393	7591	968	2305	7592	7593	4699	7594	7595	7596	7597	443	7598	2037	7599	1537	7600	7601	3922
1243	810	3947	3785	1444	7602	7603	1883	7604	1060	7605	1546	7606	1822	2679	7607	7608	603	7609
7610	7611	1441	1952	7612	7613	7614	2375	7615	7616	711	7617	7618	3437	2521	6045	1192	3823	7619
7620	7621	2636	4919	7622	1753	7623	7624	1003	7625	1842	7626	7627	3443	7628	7629	5066	5049	2382
2670	2219	7630	2894	1295	5155	1845	7631	5686	7632	938	7633	7634	5833	910	2626	2723	7635	5119
7636	1651	7637	3029	2494	7638	659	4694	3051	7639	1555	5848	7640	7641	16	6037	7642	7643	5992
3192	7644	7645	1914	7646	3278	7647	3265	127	7648	2303	7649	495	1091	7650	7651	2667	7652	2834
66	2011	1847	7653	1699	1413	5599	7654	6079	15	7655	3683	7656	1065	1660	7657	5680	7658	7659
3550	7660	7661	357	1224	833	368	6111	2232	7662	3326	7663	3193	7664	7665	86	7666	7667	436
7668	739	7669	7670	7671	4623	7672	7673	7674	7675	7676	7677	7678	7679	7680	7681	7682	7683	1261
7684	7685	7686	7687	2088	7688	7689	759	7690	7691	3535	7692	1501	3586	7693	7694	7695	7696	195
2635	2104	7697	3246	7698	7699	7700	7701	7702	3174	3892	7703	7704	7705	7706	2420	7707	718	7708
3450	3305	7709	2245	2547	7710	7711	7712	7713	7714	3427	122	7715	1567	7716	7717	7718	7719	656
7720	7721	7722	7723	7724	3028	3593	5130	5438	2228	7725	7726	7727	7728	2488	7729	7730	5902	7731
174	1916	2566	7732	68	7733	7734	1415	588	3818	336	7735	7736	7737	915	2742	7738	526	7739
7740	7741	5549	1769	3015	7742	7743	1784	7744	7745	2174	1535	7746	1710	7747	7748	7749	1714	7750
7751	7752	7753	7754	7755	1954	7756	7757	2176	7758	7759	2445	4960	7760	7761	7762	7763	155	2416
7764	3163	7765	3340	879	7766	1758	3010	7767	4892	2279	7768	5648	2711	7769	2339	7770	472	5156
7771	7772	7773	3771	7774	3119	7775	7776	3806	3793	512	7777	7778	196	2246	722	939	2223	3693
60	7779	7780	7781	7782	297	2552	1966	7783	7784	7785	7786	6023	1332	7787	7788	2860	533	7789
5025	7790	1374	7791	7792	7793	7794	3924	7795	2531	7796	7797	416	7798	2039	3008	7799	7800	2514

318	1674	7801	7802	7803	3681	7804	3237	7805	7806	3685	7807	7808	582	3641	1693	3118	7809	2728
4806	3180	440	7810	2108	7811	251	2152	1099	2197	7812	5554	1607	2839	2309	1061	3188	7813	2252
536	7814	7815	7816	7817	7818	7819	1050	7820	7821	7822	5844	7823	970	7824	2968	2540	7825	670
5452	598	2128	3132	7826	3494	2338	1789	7827	2871	3774	2926	4890	7828	7829	7830	7831	3240	7832
7833	3833	7834	7835	7836	468	7837	2297	7838	7839	7840	7841	5386	7842	7843	1141	7844	7845	4828
7846	2664	7847	7848	234	501	1135	7849	873	7850	7851	236	7852	2944	4632	7853	5304	7854	7855
53	7856	7857	7858	3625	7859	7860	7861	7862	1216	7863	7864	7865	5703	2681	7866	5895	3824	7867
7868	2310	7869	7870	1000	7871	244	976	1818	7872	1054	2613	4850	5244	7873	5965	7874	7875	1499
2797	35	7876	7877	7878	7879	3952	7880	775	1737	7881	7882	7883	7884	7885	7886	7887	7888	2175
3493	5715	72	7889	3147	3501	7890	7891	2563	7892	6033	7893	5466	7894	3569	85	7895	7896	1524
2867	7897	3858	7898	7899	964	7900	7901	211	7902	7903	3914	830	7904	1690	7905	2833	171	7906
7907	2435	355	867	7908	7909	2213	7910	7911	2859	7912	746	7913	5650	1622	2014	7914	7915	1424
7916	972	7917	100	7918	961	7919	1684	7920	7921	1592	7922	7923	2526	7924	5934	1308	736	4912
7925	1236	7926	2093	3483	7927	3434	7928	7929	7930	2206	7931	7932	2286	7933	5807	7934	2809	7935
7936	7937	7938	7939	7940	7941	7942	3093	7943	7944	224	859	2805	3034	7945	7946	7947	7948	7949
7950	2766	2274	2142	7951	5379	7952	7953	2187	3136	7954	7955	7956	2265	3787	7957	7958	7959	5291
1487	1715	7960	2930	1839	3644	2991	6110	2706	2898	7961	5576	2480	2370	4886	804	7962	793	7963
7964	7965	7966	5967	7967	7968	7969	7970	7971	7972	7973	3597	3066	7974	7975	7976	829	532	7977
7978	5092	5856	7979	7980	488	7981	7982	4677	7983	7984	92	7985	7986	7987	76	7988	7989	1932
7990	3802	3214	7991	7992	2733	5789	5704	7993	7994	7995	4952	5198	7996	7997	7998	5477	1992	1961
5193	3852	7999	8000	1098	1208	487	1404	8001	8002	8003	2928	2155	8004	8005	8006	8007	8008	146
3044	857	8009	8010	8011	3429	1029	2536	8012	2738	8013	3329	8014	8015	8016	8017	2649	1281	8018
589	2821	8019	2918	8020	2027	352	8021	8022	8023	8024	8025	8026	8027	2883	2638	8028	8029	5459
5668	8030	8031	3306	2278	521	8032	8033	8034	2590	8035	3125	2872	1819	8036	8037	71	8038	1993
8039	5234	1394	8040	8041	3022	451	8042	2043	1044	8043	5171	8044	2933	8045	8046	1294	3196	2529
8047	8048	1427	8049	5019	1409	8050	3000	8051	8052	4661	8053	213	3591	3409	8054	8055	2776	8056
8057	8058	8059	8060	8061	8062	2889	8063	1352	652	8064	2444	8065	8066	8067	8068	8069	2737	8070
8071	2519	2086	1234	1711	8072	8073	217	222	8074	1503	2573	1700	8075	8076	5975	8077	8078	8079
1019	535	8080	3189	2281	8081	168	484	8082	1893	8083	8084	2400	8085	3933	8086	3599	8087	8088
273	1754	8089	8090	8091	1275	8092	8093	8094	8095	3324	8096	5204	655	8097	8098	3375	2087	8099
3365	8100	8101	108	5523	847	5643	8102	8103	2042	37	3261	5570	5322	5871	2599	8104	8105	3773
8106	8107	8108	1196	8109	4659	8110	8111	1525	3811	4780	8112	8113	8114	639	1985	6020	8115	8116
5461	8117	8118	801	2410	737	8119	8120	8121	8122	5814	1211	3426	2271	8123	3507	1676	8124	2856
3319	2796	8125	2136	8126	3614	2845	984	8127	3412	8128	2539	8129	864	1586	8130	8131	8132	989
8133	1482	8134	8135	8136	8137	8138	23	6080	8139	5699	8140	1206	8141	187	1650	2581	2843	8142
1126	1869	8143	4871	8144	8145	2941	534	1451	8146	8147	8148	6096	328	924	294	6072	2691	8149
8150	8151	8152	1220	8153	3517	2369	8154	8155	156	1973	2140	1765	1458	8156	8157	8158	8159	8160
5691	2024	8161	8162	2437	8163	8164	8165	1797	8166	8167	106	8168	8169	242	8170	3081	1749	3144
8171	93	99	8172	8173	8174	8175	8176	2917	8177	8178	8179	1731	8180	8181	8182	8183	8184	8185
2429	1011	719	5972	3929	8186	8187	8188	8189	8190	5542	2173	8191	8192	8193	1528	3065	8194	8195
2084	8196	8197	8198	8199	2589	1958	1168	8200	8201	8202	8203	8204	8205	8206	8207	8208	8209	8210
8211	8212	962	3343	5939	898	464	8213	8214	8215	8216	572	8217	8218	8219	1160	3227	8220	3626
3328	3624	1724	8221	3258	3882	8222	1047	3600	973	8223	8224	8225	8226	3576	884	5179	5928	8227
2757	5850	8228	2476	8229	8230	8231	8232	8233	8234	8235	8236	3291	947	3528	8237	8238	8239	8240
8241	8242	8243	8244	8245	2387	3143	8246	8247	8248	8249	8250	8251	8252	8253	743	1941	8254	3283
8255	1498	646	306	119	1965	8256	425	8257	8258	8259	8260	8261	323	3083	8262	343	682	3290
3827	520	8263	2663	5248	1005	8264	2158	8265	8266	8267	3206	8268	1062	8269	1154	3505	2316	8270
2012	8271	8272	8273	1742	8274	1199	5132	1132	852	1997	2781	6087	8275	2754	3834	2489	8276	8277
2551	8278	971	5890	1343	8279	2487	8280	8281	8282	313	3592	8283	8284	2392	5757	1657	3567	2460
8285	8286	8287	8288	8289	5225	8290	8291	8292	8293	8294	8295	8296	8297	1791	575	2033	8298	8299
2950	8300	8301	5827	8302	3804	2418	745	8303	8304	2283	8305	5988	1202	669	1830	8306	8307	8308
730	8309	8310	3789	8311	2077	8312	8313	1590	2961	8314	8315	2049	8316	1734	3247	8317	1075	8318
3881	8319	8320	8321	8322	8323	8324	3160	2256	8325	3274	1589	8326	5362	8327	6097	8328	8329	8330
733	3033	8331	1190	8332	1435	8333	8334	8335	2111	8336	279	2259	4690	8337	8338	8339	8340	1421
875	1937	5583	1575	1774	8341	8342	8343	1014	960	8344	8345	239	8346	8347	1289	5426	8348	5694
8349	8350	8351	461	8352	1452	1743	8353	4945	8354	8355	3547	8356	8357	2873	3337	4665	2230	8358
2785	8359	3813	3708	8360	3286	2149	8361	8362	412	8363	8364	8365	1076	8366	8367	8368	8369	8370
3310	1894	8371	2587	8372	8373	8374	8375	4898	8376	8377	8378	1897	8379	3916	8380	8381	714	8382
8383	8384	8385	8386	3150	8387	2237	1608	1686	2912	8388	3313	1905	8389	2130	1043	8390	8391	8392
473	8393	3584	2062	3617	3400	8394	8395	2532	5559	8396	8397	8398	5627	434	8399	8400	2300	8401
8402	1284	8403	796	1086	8404	8405	1643	8406	2799	2956	2482	1725	5609	8407	8408	2528	2585	8409
1998	4604	8410	5118	2665	8411	8412	8413	396	1377	8414	8415	8416	1861	2832	5936	662	2446	8417
8418	8419	8420	2658	8421	8422	8423	31	8424	8425	8426	8427	5482	8428	8429	8430	1871	8431	4577
165	8432	1947	531	8433	2973	8434	8435	8436	3745	121	3855	8437	8438	8439	2838	73	2905	8440
8441	8442	8443	5947	2031	8444	8445	8446	1125	2058	8447	3080	8448	4932	8449	8450	892	8451	496
3512	8452	8453	8454	5229	8455	5088	712	1172	2464	8456	8457	8458	528	8459	8460	4868	8461	8462
8463	704	8464	5133	3074	2307	8465	8466	3228	2346	2207	2852	1733	8467	8468	2893	5870	8469	4576
8470	2736	8471	8472	3915	2503	280	2438	413	978	8473	133	2254	8474	201	8475	671	8476	2341
5785	3349	8477	8478	8479	5639	8480	690	8481	1984	8482	8483	8484	2650	298	8485	1238	5139	2212
1917	8486	3484	3930	2582	2471	2910	8487	5203	8488	2891	87	8489	8490	2915	8491	1326	3419	1418
5110	8492	8493	8494	1994	115	1442	5973	8495	5851	404	8496	8497	8498	8499	8500	8501	8502	3851
8503	8504	8505	1210	8506	8507	1554	2098	1108	8508	1080	1219	1545	3949	4934	5353	8509	4936	8510
8511	8512	8513	3481	3762	3515	3297	8514	741	3921	8515	3573	2479	842	8516	480	1486	8517	1203
8518	8519	8520	1460	4664	2548	8521	3678	2934	5164	560	4812	2741	2608	8522	1113	8523	8524	3402
8525	2032	2870	8526	8527	8528	2496	3164	8529	8530	8531	271	8532	8533	8534	8535	8536	4915	8537
466	6003	8538	8539	8540	8541	5917	8542	1131	8543	8544	8545	8546	1207	172	1956	8547	2458	8548

2384	5946	8549	8550	3042	8551	1188	1827	8552	8553	4958	8554	8555	872	8556	189	8557	8558	8559
8560	1982	1801	757	1491	3036	8561	2726	2924	3499	8562	2986	3179	8563	2006	1833	8564	8565	3470
8566	8567	354	8568	8569	1282	8570	5905	278	8571	777	8572	8573	8574	5527	8575	2163	8576	834
8577	8578	8579	3668	3605	8580	1318	8581	1111	5296	8582	8583	8584	8585	8586	5077	4776	8587	8588
8589	3822	8590	3849	8591	1908	948	8592	454	2533	2724	8593	8594	8595	8596	8597	8598	8599	1492
3146	3848	8600	8601	8602	8603	8604	237	2380	3128	8605	3239	8606	8607	8608	893	8609	424	2345
202	8610	8611	8612	8613	3661	2377	8614	8615	3562	8616	8617	8618	8619	2467	2577	2073	2085	3071
8620	8621	3046	8622	8623	3232	8624	8625	1205	4756	3627	623	8626	8627	152	8628	2525	8629	415
8630	8631	8632	8633	8634	5094	8635	8636	2475	8637	5816	1136	1558	1254	2425	3943	2593	2421	1832
8638	1364	8639	8640	3460	787	5481	8641	2607	8642	1656	2811	8643	5573	8644	8645	1756	8646	83
684	8647	8648	2426	8649	8650	3948	766	3025	3726	3299	8651	8652	1615	8653	8654	8655	1252	8656
8657	5124	8658	8659	1838	3604	790	3135	574	8660	8661	8662	8663	8664	2082	2510	8665	2553	1570
3904	1280	1360	1230	8666	5180	8667	8668	8669	2095	8670	8671	8672	8673	8674	8675	8676	667	1412
8677	8678	8679	8680	287	3068	8681	8682	8683	1691	977	8684	8685	8686	8687	8688	8689	8690	637
8691	1008	8692	8693	8694	8695	626	8696	5349	1969	167	8697	8698	1578	5083	2466	8699	8700	8701
8702	8703	1794	8704	160	8705	706	8706	8707	912	596	8708	8709	8710	8711	3633	1305	8712	8713
8714	8715	707	3694	2328	8716	8717	8718	8719	8720	5126	5795	8721	786	8722	8723	1073	2646	5921
8724	3463	827	8725	3770	2169	8726	8727	8728	1425	2333	8729	8730	681	8731	4680	8732	8733	889
8734	3935	5693	8735	4908	8736	8737	8738	8739	8740	18	2946	2972	4748	3455	8741	8742	8743	1494
8744	717	1296	8745	2498	8746	8747	2427	2508	8748	1923	8749	8750	8751	3779	6	8752	1373	400
8753	8754	567	3092	8755	5084	8756	8757	1540	8758	2123	8759	1924	8760	8761	8762	8763	8764	3716
8765	8766	3213	8767	5790	2243	2641	1757	8768	8769	3116	8770	8771	8772	2409	8773	3064	8774	8775
8776	2177	8777	2594	5112	1348	3351	8778	8779	8780	8781	2269	8782	8783	8784	3857	8785	8786	8787
88	3063	1166	8788	2505	316	515	8789	3322	2935	2362	2507	8790	8791	8792	5518	8793	3373	8794
8795	1716	8796	2364	2258	5261	616	2276	758	8797	8798	779	8799	8800	1239	4600	1148	3090	8801
8802	8803	2020	8804	8805	1813	1022	8806	1837	109	3594	8807	3524	2568	2034	8808	8809	3476	795
8810	8811	8812	8813	2117	1999	4736	8814	3327	3643	2203	3211	8815	8816	8817	8818	4792	8819	1859
3382	1173	282	8820	8821	8822	8823	8824	8825	2251	641	3422	8826	305	624	8827	4808	8828	8829
8830	8831	8832	8833	1279	1748	8834	8835	540	5629	3653	1635	8836	8837	4857	8838	8839	8840	8841
8842	2705	8843	4874	8844	3335	246	8845	843	2615	8846	3743	8847	8848	1401	3488	8849	8850	2822
8851	8852	2153	379	5804	8853	5505	8854	8855	5760	1899	8856	8857	8858	8859	8860	147	8861	8862
407	498	2459	3017	8863	1823	1509	8864	8865	2139	3578	8866	8867	8868	8869	8870	1045	3739	8871
5174	8872	5281	2025	8873	8874	2592	1755	8875	8876	8877	1548	4608	8878	8879	8880	8881	8882	8883
8884	8885	2780	2864	3331	1214	8886	8887	8888	422	8889	8890	8891	1467	3803	8892	8893	2413	8894
8895	1703	8896	262	8897	1764	2857	2403	4930	1930	3113	4877	8898	3502	8899	4718	8900	1500	2052
8901	8902	2875	8903	3333	8904	837	8905	2826	1169	1788	8906	556	8907	8908	569	3525	2064	1611
705	33	8909	3945	724	5105	990	8910	8911	2442	763	8912	8913	8914	8915	8916	8917	8918	3318
8919	8920	8921	5867	3458	8922	2560	8923	8924	411	8925	2238	8926	8927	8928	6046	3321	8929	5031
983	5948	8930	2211	1704	8931	2054	5010	2285	1185	8932	3386	8933	8934	748	8935	8936	8937	8938
8939	8940	8941	8942	8943	8944	8945	5932	1273	8946	1903	274	2105	182	8947	8948	8949	601	444
2129	8950	8951	8952	8953	8954	22	8955	8956	8957	8958	8959	8960	1493	8961	8962	8963	139	3284
8964	8965	2399	3925	8966	685	2366	2814	8967	3111	8968	8969	8970	2118	1565	3271	1134	358	8971
8972	3703	5079	8973	8974	8975	3257	8976	3766	365	1561	2472	1647	956	8977	8978	5502	8979	8980
2982	1027	8981	558	179	8982	8983	8984	8985	3005	8986	8987	8988	8989	8990	2137	8991	8992	8993
8994	8995	8996	6084	6088	8997	2546	8998	5330	8999	9000	9001	3123	9002	9003	9004	9005	1187	1017
2450	4973	9006	9007	9008	9009	9010	848	9011	9012	2789	9013	9014	9015	9016	2554	1606	9017	1873
9018	3485	3102	3704	9019	117	9020	9021	9022	9023	263	1870	5383	2718	4766	9024	740	2465	2959
3468	9025	9026	9027	9028	9029	1516	1948	3212	9030	9031	3120	3509	9032	3187	9033	9034	945	2138
1872	4753	3012	9035	1595	9036	742	3293	1892	394	9037	562	2394	3941	9038	523	9039	2439	9040
9041	9042	9043	9044	2966	9045	9046	2275	5021	2196	78	8	3519	9047	9048	9049	9050	9051	9052
9053	6057	9054	1915	9055	6094	427	9056	2323	9057	9058	9059	9060	9061	1229	9062	5919	9063	5897
2888	9064	4853	497	9065	9066	9067	5412	9068	9069	4804	9070	2457	9071	2358	3883	4633	9072	5560
3300	3084	9073	9074	3692	9075	1707	9076	1834	9077	9078	9079	9080	9081	9082	3466	1476	9083	635
9084	9085	5995	9086	3428	9087	9088	2423	2048	9089	3699	9090	9091	2016	1201	625	410	3077	9092
118	9093	9094	9095	3756	9096	1342	385	5613	9097	840	9098	9099	9100	9101	9102	1648	9103	2215
9104	9105	9106	9107	993	9108	2788	1381	9109	4834	9110	9111	3094	4742	9112	1591	3706	9113	1825
9114	9115	198	887	1290	9116	9117	4612	507	9118	9119	344	3439	9120	9121	9122	9123	3152	9124
5729	24	9125	1553	9126	9127	2408	2431	9128	9129	3200	9130	2374	9131	5047	9132	3085	1371	9133
9134	3564	4831	2625	700	1386	1616	5310	5645	5488	9135	9136	9137	9138	9139	9140	4964	94	9141
3698	9142	9143	9144	9145	2759	1852	9146	9147	9148	9149	9150	5263	9151	9152	3032	9153	3215	9154
9155	5185	9156	281	9157	2882	510	465	9158	9159	9160	9161	2992	3133	30	9162	9163	9164	9165
9166	2096	9167	9168	1103	5070	2858	9169	672	1138	3553	9170	1026	9171	9172	9173	935	9174	453
3370	9175	954	2455	5024	9176	9177	5388	9178	9179	9180	3109	2164	2159	9181	573	9182	9183	1137
5579	1365	650	551	9184	612	1808	110	9185	9186	9187	1455	3104	9188	3537	1949	390	21	3523
9189	9190	9191	9192	9193	9194	9195	3723	9196	4924	9197	9198	9199	4910	9200	9201	5245	2932	9202
3359	9203	9204	9205	9206	9207	9208	9209	9210	9211	9212	9213	9214	1471	9215	6106	9216	5195	1596
389	9217	46	3281	69	9218	2239	2631	3893	1232	269	5952	9219	932	2657	5976	9220	9221	9222
9223	9224	6051	9225	9226	9227	9228	9229	3741	67	1978	9230	658	3900	9231	5187	3570	9232	1875
5641	1223	638	9233	9234	1579	9235	1116	9236	56	9237	1975	9238	9239	192	5582	9240	844	9241
5143	9242	3585	1477	1708	755	9243	3361	1654	9244	9245	9246	9247	1143	9248	9249	9250	44	9251
3048	9252	5157	5040	3928	376	4747	9253	9254	666	3052	1696	2490	1594	4713	9255	9256	1967	346
9257	3256	9258	1602	9259	9260	3334	761	426	9261	9262	1526	1068	9263	9264	9265	9266	9267	5775
9268	2732	1181	9269	2110	9270	3791	3275	1800	9271	9272	3309	4914	3696	9273	9274	9275	3527	9276
2056	9277	9278	9279	9280	9281	9282	1920	9283	9284	565	9285	3895	5260	5253	1655	1762	9286	9287
998	9288	9289	9290	9291	9292	9293	1399	9294	3182	9295	208	5440	9296	2612	9297	9298	9299	4959
9300	9301	9302	2979	3810	9303	5476	9304	1110	9305	9306	3050	1989	2527	1855	9307	3467	9308	9309

9310	9311	3590	3401	663	9312	9313	1671	9314	9315	9316	9317	9318	2765	9319	9320	9321	9322	4858	
9323	131	9324	9325	9326	9327	5808	1453	1078	9328	397	9329	9330	17	9331	645	3304	9332	9333	
307	348	9334	9335	9336	9337	9338	9339	9340	854	9341	4732	2132	9342	9343	3666	9344	9345	9346	
9347	9348	9349	9350	9351	2199	1946	9352	9353	9354	3408	1184	3686	5681	9355	1669	691	9356	5046	
260	600	9357	9358	9359	20	3418	9360	3377	3690	9361	9362	9363	9364	9365	9366	9367	2474	9368	
9369	5978	914	9370	9371	1383	1324	1114	9372	9373	9374	2740	2308	9375	2571	9376	9377	9378	2621	
9379	2570	9380	1191	2456	9381	9382	3364	9383	9384	2996	807	9385	9386	2451	9387	36	9388	5460	
9389	5413	9390	1706	9391	9392	9393	9394	9395	9396	1416	367	2714	9397	9398	9399	9400	462	1582	
9401	9402	9403	1267	3889	9404	9405	381	9406	9407	3680	9408	5172	2685	5910	4809	310	9409	247	
9410	9411	1688	9412	9413	9414	696	9415	9416	9417	9418	3731	9419	9420	3797	3007	9421	9422	9423	
9424	9425	2368	9426	9427	4652	9428	9429	9430	9431	6027	9432	4784	2807	3173	2253	2493	1795	9433	
183	4882	1824	9434	1157	3749	9435	9436	1102	9437	9438	1272	9439	9440	9441	9442	2381	1814	9443	
9444	9445	2787	9446	9447	1913	2388	9448	9449	9450	2806	3416	1865	9451	204	3631	3378	2640	1786	
9452	9453	5858	3411	9454	9455	9456	205	9457	9458	9459	9460	5741	2712	9461	9462	2651	1263	9463	
9464	1474	1085	9465	2061	579	9466	9467	9468	300	1882	1319	9469	9470	9471	545	9472	2188	9	
3366	9473	1506	9474	1681	749	4738	9475	9476	9477	9478	4575	9479	986	1380	2419	9480	1040	3516	
9481	9482	5651	2654	9483	3886	9484	9485	5162	1778	2695	9486	9487	9488	9489	221	3445	9490	9491	
3106	9492	2795	5499	9493	1799	1900	9494	9495	3558	2019	9496	2468	1096	3413	9497	3684	2556	5058	
9498	9499	9500	9501	1538	9502	1153	9503	3839	9504	9505	9506	9507	9508	9509	490	2976	3674	9510	
9511	9512	2	3477	1079	3096	1182	9513	9514	693	9515	2567	3931	9516	9517	9518	2473	9519	5342	
9520	9521	9522	6031	1807	2720	5170	9523	9524	1245	812	9525	9526	1226	9527	9528	9529	9530	3808	
9531	858	1933	9532	1668	1746	5373	9533	3492	4913	9534	2386	2659	1240	4844	2360	3175	9535	9536	
5843	3374	9537	698	5654	9538	1361	2516	9539	9540	1322	5797	2561	9541	9542	4615	3041	9543	1775	
9544	4772	9545	701	9546	9547	2958	9548	1922	9549	9550	9551	9552	3273	9553	2962	4769	9554	9555	
1768	9556	9557	9558	4926	9559	9560	9561	9562	9563	9564	1235	9565	9566	9567	9568	9569	9570	47	
2919	5985	2329	2242	9571	9572	9573	2869	9574	4654	1397	5657	5750	5873	9575	9576	9577	3016	9578	
3248	3339	9579	3474	1879	5005	1804	2841	5780	6022	9580	9581	5800	9582	9583	2939	9584	91	9585	
9586	9587	9588	9589	9590	9591	3030	9592	2777	9593	9594	2829	2818	9595	5841	9596	5056	9597	2363	
4739	9598	9599	9600	9601	9602	3675	6078	9603	9604	636	3296	9605	3038	9606	9607	309	9608	9609	
9610	1446	9611	2937	2241	9612	9613	9614	3449	9615	9616	9617	3338	3538	9618	9619	9620	2903	9621	
1683	9622	9623	9624	3057	9625	9626	9627	1495	2249	1925	9628	9629	634	5637	929	9630	9631	39	
753	9632	3639	9633	9634	9635	128	4887	3088	504	9636	3395	1292	5736	2081	1052	3740	9637	1552	
9638	9639	2075	9640	9641	1968	2686	9642	9643	9644	9645	3602	9646	9647	1806	9648	9649	9650	9651	
1581	3652	2837	9652	4635	9653	9654	9655	9656	845	3362	173	3531	9657	2502	2099	1006	9658	1128	
606	9659	5288	9660	9661	4730	9662	4888	4705	1302	4609	50	2604	135	9663	9664	9665	1209	9666	
3018	9667	909	3472	1164	9668	9669	9670	9671	1556	229	9672	6105	9673	2643	9674	3072	2846	4893	
2044	9675	9676	4724	1428	9677	9678	9679	9680	1520	9681	9682	9683	1097	9684	9685	9686	4689	9687	
1550	9688	3700	9689	1544	9690	9691	3536	6090	9692	676	2114	9693	1484	5016	9694	1250	1962	9695	
817	9696	771	2949	3054	9697	3784	9698	9699	9700	9701	153	9702	2260	5428	3216	3495	9703	2854	
9704	9705	1943	2330	9706	9707	5835	1167	9708	1902	9709	9710	9711	2449	9712	9713	3937	320	9714	
366	9715	1354	3710	9716	9717	2226	3688	660	9718	9719	1829	4801	9720	9721	9722	907	2299	9723	
3701	9724	4981	9725	9726	9727	2849	9728	5828	3294	9729	9730	3277	9731	1328	3920	3075	2121	1831	
9732	9733	9734	1081	9735	9736	3424	738	586	9737	9738	2233	9739	9740	9741	1955	5632	9742	9743	
420	9744	3669	9745	9746	5075	9747	9748	3388	141	9749	9750	9751	9752	9753	4688	4829	3259	2221	
1013	3720	9754	4843	1449	9755	1222	9756	9757	9758	3435	3285	2998	3946	611	9759	1588	9760	9761	
9762	9763	9764	9765	9766	9767	3853	9768	632	9769	9770	9771	5252	502	9772	3155	9773	9774	824	
429	2684	1942	9775	2483	9776	5173	9777	375	9778	9779	9780	9781	9782	9783	2750	199	1511	3148	
9784	530	9785	9786	9787	9788	9789	9790	9791	9792	3208	2688	158	9793	1419	9794	9795	2018	9796	
2406	906	9797	9798	5622	2704	9799	9800	9801	9802	9803	1155	1637	3267	2185	9804	870	9805	9806	
9807	9808	9809	9810	9811	9812	9813	751	1880	9814	9815	1283	9816	2617	9817	9818	9819	9820	1025	
9821	9822	9823	49	3385	9824	9825	9826	9827	1180	9828	9829	2816	3352	9830	9831	2401	9832	125	
1358	508	9833	9834	9835	112	9836	9837	9838	9839	9840	642	9841	9842	9843	9844	9845	816	3642	
9846	9847	9848	9849	813	5689	9850	9851	9852	9853	5188	9854	2850	9855	3718	2572	9856	1564	9857	
9858	9859	2433	9860	9861	80	6082	9862	9863	9864	3270	9865	1304	81	1384	303	2954	940	9866	
214	5012	4935	9867	1653	1130	9868	9869	9870	9871	3255	9872	9873	9874	9875	9876	9877	317	9878	
9879	9880	9881	2495	3606	866	9882	785	9883	856	3201	2463	5427	9884	9885	3940	9886	9887	1432	
797	220	9888	9889	9890	38	442	3672	3880	3630	9891	9892	2112	9893	5068	9894	9895	9896	2200	
2689	9897	3560	2355	9898	2951	9899	3542	9900	9901	9902	5500	9903	3037	1445	2770	1488	3031	6038	
1926	9904	414	2981	3161	9905	5535	1194	3899	860	5009	1392	226	713	9906	2735	1640	3664	5874	
5352	9907	469	9908	9909	9910	9911	3817	9912	5106	9913	1785	9914	9915	9916	943	9917	9918	9919	
2404	1475	2748	9920	9921	9922	9923	9924	4820	9925	2344	9926	9927	9928	2634	3518	9929	3421	9930	
9931	249	1566	9932	9933	9934	3632	9935	9936	9937	9938	9939	9940	9941	9942	9943	1093	9944	3209	
9945	850	9946	9947	2963	9948	1987	511	1112	2967	9949	593	1574	9950	2047	9951	9952	1266	9953	
9954	9955	9956	459	1426	9957	3039	3020	9958	5	9959	9960	9961	9962	9963	9964	2227	9965	9966	
9967	286	9968	9969	608	9970	9971	64	9972	9973	920	1372	9974	9975	1021	1133	9976	9977	9978	
9979	2143	6076	9980	9981	9982	3344	715	9983	9984	5601	4625	3336	9985	1151	9986	9987	9988	3406	
9989	1662	9990	2622	1531	9991	5491	9992	1041	9993	2895	2116	2125	9994	9995	9996	9997	3648	9998	
9999	3167	2071	1313	10000	10001	3266	10002	10003	3702	5747	10004	10005	1064	5451	10006	1429	4822	3581	
4702	2596	643	10007	10008	10009	10010	3205	10011	253	10012	10013	10014	6009	10015	2844	10016	10017	825	
2186	3871	3103	2292	5803	10018	10019	5701	10020	3003	10021	5968	10022	10023	10024	10025	4903	10026	699	
522	2453	10027	2825	10028	10029	1	364	2115	10030	10031	10032	10033	10034	1557	1307	10035	10036	10037	
10038	10039	10040	2127	10041	10042	673	10043	10044	10045	3134	3464	10046	10047	10048	10049	3689	10050	10051	
2578	10052	10053	10054	10055	10056	10057	5213	1402	1024	10058	10059	6008	1601	10060	5473	3854	10061	729	
10062	10063	223	1242	1257	1303	10064	10065	10066	3110	10067	10068	3846	10069	10070	10071	10072	157	3508	
2862	2191	10073	563	10074	2745	10075	1519	3202	3737	1820	1738	10076	25	2753	10077	10078	10079	10080	
161	2144	686	10081	10082	10083	10084	10085	5821	2840	10086	4781	1228	5553	10087	2983	10088	10089	3783	

10090 3884 10091 10092 10093 828 10094 2716 4963 79 3859 1382 10095 4725 1329 936 10096 2672 2361
10097 10098 3151 1816 10099 1231 10100 10101 3500 1271 10102 3909 831 10103 2965 1009 10104 10105 10106
10107 10108 525 2955 10109 10110 335 5900 10111 5682 10112 350 1868 10113 10114 2588 3149 10115 4684
10116 10117 4607 10118 3894 10119 10120 10121 10122 10123 10124 2317 70 10125 1225 2396 1518 2921 5468
10126 10127 934 10128 10129 419 2878 437 10130 10131 10132 10133 3440 10134 5108 2397 602 10135 10136
1848 206 891 1463 5783 10137 10138 1144 10139 10140 3454 1391 1856 4599 10141 10142 10143 361 10144
10145 10146 10147 10148 10149 342 144 10150 10151 5700 10152 885 2632 2619 10153 10154 3758 3019 2952
10155 10156 5548 2405 2774 10157 10158 3055 10159 10160 3130 878 10161 10162 2220 4947 296 1049 10163
10164 10165 3504 10166 10167 10168 2518 4761 10169 1323 10170 10171 10172 10173 10174 2943 1939 10175 2195
2182 886 10176 10177 26 10178 2007 5287 1127 1521 4992 140 10179 5319 1353 212 3682 10180 3241
10181 10182 10183 10184 10185 10186 10187 132 1513 2080 2931 10188 10189 2013 10190 1728 10191 10192 10193
10194 2866 10195 10196 10197 10198 10199 10200 3838 10201 1614 10202 10203 369 10204 5805 10205 1293 1779
175 1527 2802 10206 2541 3738 1793 5721 10207 3124 2141 10208 3611 911 10209 710 10210 10211 1314
10212 849 3513 10213 3748 10214 1901 10215 2348 10216 10217 3105 2637 2974 10218 10219 10220 3101 10221
10222 10223 3131 3534 10224 10225 10226 2980 10227 10228 10229 10230 1849 5792 10231 5887 3496 1443 10232
992 10233 10234 4787 10235 5635 10236 2314 314 5958 10237 10238 2810 2901 10239 1369 2761 10240 10241
2696 5227 2855 486 10242 1481 10243 3828 2122 2340 1896 10244 3243 2574 3805 1721 2595 2747 2302
1675 10245 10246 10247 10248 5257 5125 3236 10249 5120 10250 3831 5420 10251 5424 10252 3250 996 10253
10254 883 10255 5435 1161 10256 188 10257 10258 10259 10260 10261 3583 1430 55 10262 3316 1931 10263
10264 10265 10266 10267 373 997 2167 283 10268 10269 10270 2311 2559 10271 10272 10273 10274 10275 10276
10277 10278 2925 10279 1510 10280 10281 2549 386 10282 10283 1175 10284 10285 10286 10287 3837 2218 10288
3913 2022 10289 10290 10291 10292 1101 10293 1792 10294 2057 325 1436 2017 664 10295 3091 10296 2771
10297 1092 10298 517 10299 3221 1298 10300 10301 10302 5649 3735 10303 1058 10304 10305 345 10306 1070
10307 581 1988 4923 2133 1465 2063 3154 10308 5344 441 10309 2509 10310 10311 10312 4835 10313 10314
5773 10315 10316 2874 10317 10318 2156 1507 5964 1844 10319 3127 10320 10321 3121 10322 10323 10324 5994
10325 10326 166 10327 291 10328 1038 265 3169 5026 10329 10330 10331 10332 10333 3307 1751 10334 3636
10335 10336 248 10337 10338 3230 10339 10340 10341 10342 1355 1337 10343 838 32 3086 2999 10344 10345
881 10346 10347 1237 1422 10348 10349 2828 10350 1571 823 10351 10352 2734 10353 10354 10355 1726 10356
10357 10358 10359 10360 10361 10362 10363 5784 3021 3539 10364 4641 10365 10366 2648 10367 1597 2026 5027
10368 2040 10369 10370 10371 10372 2257 10373 10374 2677 10375 809 3772 5206 10376 1593 10377 10378 10379
10380 2331 10381 10382 10383 10384 3522 3566 3874 2126 184 10385 10386 347 10387 10388 10389 3204 5177
10390 13 10391 10392 3244 1547 215 10393 10394 10395 4640 82 10396 10397 10 1514 10398 10399 10400
10401 1497 10402 10403 3873 10404 1919 10405 988 97 3002 10406 10407 10408 10409 2512 10410 10411 10412
10413 10414 10415 764 10416 1406 5097 10417 10418 2469 1512 2378 2764 10419 5846 10420 2461 65 10421
5908 10422 10423 3190 597 10424 3796 3436 5624 10425 10426 10427 1417 2209 10428 10429 4861 3663 3676
245 10430 10431 10432 3011 10433 10434 10435 10436 10437 10438 10439 10440 10441 2773 2820 10442 10443 10444
5525 1012 10445 10446 10447 10448 6011 10449 10450 10451 10452 10453 2772 10454 10455 10456 3394 10457 10458
10459 1152 2940 10460 1082 10461 3254 10462 2376 10463 3367 10464 5955 10465 5545 10466 10467 3498 10468
2755 10469 10470 2347 10471 10472 1379 2767 203 10473 4719 1227 10474 10475 3014 1016 10476 10477 10478
311 1972 3234 953 3115 10479 2194 3108 10480 200 731 10481 10482 10483 10484 599 2997 10485 10486
10487 10488 10489 10490 2642 2170 10491 10492 1732 194 10493 1002 3847 3004 2078 10494 3944 10495 10496
10497 832 267 10498 142 10499 5829 10500 1150 10501 10502 10503 3430 10504 10505 1195 10506 1056 1624
382 10507 5400 10508 2441 10509 1803 456 10510 3095 3303 272 5683 5847 3655 3775 1898 1720 10511
3891 10512 10513 4949 1604 10514 10515 10516 10517 10518 10519 10520 2282 821 10521 10522 138 10523 1723
3145 10524 10525 10526 10527 3555 2692 10528 1174 1349 1639 89 10529 5127 10530 546 3251 10531 2289
10532 10533 591 10534 10535 10536 2157 10537 10538 10539 1340 10540 1253 10541 3363 1863 5028 10542 10543
10544 3670 10545 2823 10546 10547 10548 10549 1790 10550 494 1854 2945 5209 4814 2990 10551 869 10552
10553 10554 10555 10556 10557 2879 10558 10559 3753 3023 10560 331 2922 10561 3350 10562 1341 10563 10564
10565 1739 10566 10567 862 10568 590 10569 10570 1317 288 2295 10571 2412 10572 10573 3490 114 10574
2235 5556 3860 10575 10576 10577 888 5509 360 10578 10579 3089 10580 10581 5990 10582 10583 3262 3199
1204 3059 10584 2501 10585 10586 10587 10588 10589 5661 3568 10590 10591 10592 2987 6059 3660 10593 10594
10595 10596 10597 10598 10599 1747 3314 10600 10601 5060 478 10602 1462 1388 10603 10604 1940 4928 10605
10606 814 1370 10607 10608 2713 10609 10610 10611 10612 10613 3865 10614 10615 10616 1877 1159 10617 10618
10619 952 5718 10620 10621 10622 10623 10624 10625 10626 1959 10627 822 10628 5892 10629 10630 10631 1573
10632 10633 1670 3380 4968 628 1439 4953 10634 3612 10635 1031 585 10636 2989 5216 493 10637 10638
2890 2731 10639 10640 4657 10641 10642 2151 10643 4998 1046 10644 10645 10646 4879 3533 784 10647 10648
2270 10649 10650 5907 4644 10651 4610 3245 10652 1652 5100 10653 10654 2975 10655 10656 3719 3752 10657
10658 10659 2352 10660 10661 10662 5363 230 2109 2627 10663 10664 10665 723 2357 2372 10666 10667 10668
1632 2202 10669 10670 10671 1071 10672 10673 5445 10674 6107 2618 10675 2379 10676 3730 10677 1705 800
2041 2906 10678 10679 2248 3722 10680 10681 10682 10683 10684 2356 10685 10686 2424 2697 5868 3582 10687
5516 10688 10689 10690 10691 10692 2021 1840 301 2752 10693 10694 10695 10696 10697 1077 10698 1730 10699
3777 10700 10701 3673 10702 942 10703 10704 4896 10705 10706 1853 661 2511 3843 10707 5324 10708 5957
5308 1362 10709 10710 2224 4794 10711 1857 10712 10713 3181 2165 10714 10715 10716 3799 10717 3360 3441
10718 5300 10719 10720 1094 1912 1745 10721 10722 2103 1089 778 10723 3677 485 10724 5487 5159 10725
1577 90 10726 10727 3325 2606 10728 5774 6015 644 1469 10729 2542 10730 865 10731 1697 10732 374
10733 1881 10734 393 3479 10735 4588 10736 3628 10737 1215 10738 10739 10740 2538 10741 10742 10743 10744
10745 277 2699 10746 10747 10748 399 1197 2791 10749 10750 5647 908 1664 10751 10752 10753 1698 10754
10755 2266 3457 2786 14 1646 1470 1357 10756 1549 3231 10757 10758 1862 10759 5446 1936 10760 10761
190 10762 3938 3456 10763 651 505 10764 10765 10766 1628 10767 10768 10769 3279 633 235 3183 371
5113 10770 3637 1750 10771 1393 232 559 3901 180 10772 10773 150 10774 10775 3552 10776 679 10777
10778 10779 10780 10781 3667 10782 10783 725 3746 10784 1612 1331 680 2768 10785 5969 2923 2150 1712
1320 10786 728 688 1878 529 607 4944 10787 5098 10788 10789 10790 3786 185 11 1277 10791 2325
2698 10792 10793 10794 10795 4649 1450 1938 10796 10797 5915 3323 1759 994 10798 3288 10799 10800 10801
3114 10802 10803 561 10804 10805 2881 2210 10806 2727 3724 2334 10807 10808 10809 10810 10811 3249 10812
1247 2287 10813 10814 10815 2273 5515 913 861 10816 5610 2600 10817 10818 10819 408 2135 129 10820
5940 772 10821 5182 1437 1921 10822 3289 10823 1667 10824 10825 10826 10827 3917 10828 10829 10830 3405

10831 10832 10833 10834 10835 3609 4638 10836 10837 5508 10838 10839 3832 10840 3658 10841 10842 10843 28
584 10844 10845 3404 10846 4778 10847 750 3603 4819 499 3024 10848 1118 5717 2050 5979 1489 2603
10849 10850 3478 774 3157 10851 10852 10853 284 10854 10855 10856 10857 351 10858 2703 1951 1866 4581
10859 890 10860 10861 1249 10862 10863 10864 10865 10866 10867 5483 10868 10869 10870 10871 10872 10873 4883
10874 10875 10876 10877 10878 4668 10879 10880 4731 10881 10882 10883 10884 10885 10886 10887 10888 10889 10890
4614 10891 10892 10893 10894 10895 10896 10897 5698 10898 10899 10900 10901 10902 10903 10904 10905 10906 10907
10908 10909 10910 10911 10912 10913 10914 10915 4619 4758 5652 10916 10917 5660 5290 10918 10919 10920 10921
5434 10922 10923 10924 10925 10926 10927 10928 10929 10930 5672 10931 10932 10933 10934 10935 10936 10937 10938
10939 10940 10941 10942 10943 10944 10945 10946 10947 10948 5007 4911 10949 10950 10951 10952 4630 10953 10954
10955 10956 10957 10958 10959 10960 10961 10962 10963 10964 10965 10966 10967 10968 10969 5457 10970 10971 10972
10973 10974 5334 10975 10976 10977 10978 10979 10980 10981 10982 5756 10983 10984 10985 10986 6083 10987 10988
5916 10989 10990 10991 10992 4909 10993 10994 10995 10996 10997 10998 10999 11000 11001 11002 11003 11004 11005
11006 5289 11007 11008 11009 11010 11011 5429 11012 11013 11014 11015 11016 11017 5901 11018 11019 11020 11021
11022 11023 11024 11025 11026 11027 5594 11028 11029 11030 11031 11032 11033 11034 11035 11036 11037 11038 11039
11040 4962 11041 11042 11043 11044 11045 4768 11046 5295 11047 11048 5284 11049 4704 11050 11051 11052 5140
11053 11054 5377 11055 6050 11056 11057 11058 11059 11060 11061 11062 11063 11064 11065 6005 11066 4646 11067
11068 11069 4613 11070 11071 11072 11073 11074 11075 11076 11077 11078 5671 11079 4695 11080 11081 11082 11083
11084 11085 11086 11087 11088 11089 11090 11091 11092 11093 11094 11095 11096 11097 11098 11099 11100 11101 11102
11103 11104 5343 11105 11106 11107 11108 11109 11110 11111 11112 6063 11113 11114 4716 4830 11115 11116 11117
11118 11119 11120 11121 11122 6041 11123 5962 11124 11125 11126 11127 11128 11129 11130 11131 11132 11133 11134
11135 11136 11137 11138 11139 11140 11141 11142 11143 4927 11144 5181 11145 11146 11147 11148 11149 11150 5966
4578 11151 11152 11153 11154 11155 5857 11156 11157 11158 11159 11160 11161 11162 11163 11164 11165 11166 11167
11168 11169 5927 11170 11171 11172 11173 5581 11174 11175 11176 11177 4740 11178 11179 11180 11181 11182 11183
11184 11185 11186 11187 11188 11189 11190 11191 11192 11193 4675 11194 11195 5082 11196 11197 5880 11198 11199
11200 11201 5540 11202 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11213 11214 11215 11216 11217
11218 11219 11220 11221 11222 11223 11224 11225 11226 11227 11228 11229 11230 11231 11232 11233 11234 11235 11236
11237 11238 11239 11240 11241 11242 11243 11244 11245 5971 11246 11247 11248 11249 11250 11251 11252 11253 11254
11255 11256 11257 11258 11259 11260 11261 11262 11263 11264 5894 11265 11266 11267 11268 11269 11270 11271 11272
11273 11274 4624 11275 11276 11277 11278 11279 4984 11280 11281 11282 11283 11284 11285 11286 11287 11288 11289
5676 11290 11291 11292 11293 11294 5628 11295 11296 11297 11298 11299 5018 4703 11300 11301 11302 11303 11304
11305 11306 11307 11308 11309 11310 11311 6102 11312 11313 5492 11314 11315 11316 11317 11318 5771 11319 11320
11321 11322 11323 11324 11325 11326 11327 11328 11329 11330 5275 4655 11331 11332 11333 11334 5136 5980 11335
11336 11337 11338 11339 11340 11341 11342 4734 11343 11344 11345 11346 11347 11348 5465 4860 11349 11350 11351
5199 11352 11353 11354 11355 11356 11357 5564 11358 11359 5731 11360 11361 5355 5762 11362 11363 11364 11365
11366 5183 11367 11368 11369 11370 11371 11372 11373 11374 11375 11376 11377 11378 4939 11379 11380 11381 11382
5357 11383 6032 5367 11384 11385 11386 11387 11388 11389 5588 11390 11391 11392 11393 11394 11395 11396 11397
11398 11399 11400 11401 11402 11403 11404 11405 11406 11407 11408 11409 11410 11411 11412 11413 11414 11415 11416
11417 11418 11419 11420 11421 11422 11423 5230 11424 11425 5495 11426 11427 11428 11429 11430 11431 11432 11433
11434 11435 11436 11437 11438 11439 4726 11440 11441 5003 4785 5240 11442 11443 11444 11445 11446 11447 11448
11449 11450 11451 5655 5493 11452 11453 4708 4580 11454 11455 11456 11457 11458 11459 11460 11461 5048 11462
11463 11464 11465 11466 11467 11468 11469 6061 11470 5250 11471 11472 11473 11474 11475 5149 11476 11477 11478
11479 11480 11481 11482 11483 11484 11485 11486 11487 11488 11489 11490 11491 11492 11493 11494 11495 11496 5608
11497 11498 11499 5882 5071 4986 11500 11501 11502 11503 11504 5043 11505 11506 5562 11507 11508 11509 11510
4786 11511 11512 11513 11514 11515 11516 11517 11518 11519 11520 11521 11522 11523 11524 5678 11525 11526 11527
11528 11529 11530 11531 11532 11533 11534 5702 11535 11536 11537 4938 11538 11539 11540 11541 11542 11543 5722
11544 11545 11546 11547 11548 11549 11550 11551 11552 11553 11554 11555 11556 11557 11558 11559 11560 11561 11562
11563 11564 11565 11566 11567 11568 11569 11570 11571 11572 11573 11574 11575 11576 11577 11578 11579 11580 11581
11582 11583 5237 11584 11585 11586 11587 11588 11589 11590 11591 11592 11593 11594 11595 5148 11596 11597 11598
11599 11600 11601 11602 5685 11603 4666 11604 11605 11606 11607 11608 11609 11610 11611 11612 11613 11614 11615
11616 11617 11618 11619 11620 11621 11622 11623 11624 11625 11626 11627 11628 11629 11630 11631 11632 5086 5621
11633 11634 11635 11636 11637 11638 11639 11640 5517 11641 11642 11643 11644 11645 5200 11646 11647 11648 11649
11650 11651 11652 11653 11654 11655 5744 11656 4602 11657 11658 11659 11660 11661 11662 11663 11664 5035 11665
11666 11667 11668 4586 5726 11669 4637 4603 11670 11671 11672 11673 11674 11675 11676 11677 11678 11679 11680
11681 11682 11683 11684 11685 11686 11687 11688 11689 11690 11691 4991 5674 11692 11693 11694 11695 5552 11696
5147 11697 11698 11699 11700 11701 5380 11702 11703 11704 11705 11706 11707 11708 11709 5811 11710 11711 11712
11713 11714 11715 11716 11717 11718 11719 11720 11721 11722 11723 11724 5612 11725 11726 5623 11727 5283 11728
6052 11729 11730 4869 11731 11732 11733 11734 11735 11736 11737 11738 11739 11740 11741 11742 5103 11743 11744
11745 11746 11747 11748 11749 11750 4867 11751 11752 11753 11754 11755 11756 11757 11758 11759 11760 11761 11762
11763 11764 11765 11766 11767 5834 11768 4902 11769 11770 11771 4662 4974 4693 11772 11773 5998 11774 11775
11776 11777 5276 11778 11779 11780 4779 11781 11782 11783 11784 11785 4954 5309 11786 11787 5282 11788 11789
11790 11791 11792 6099 11793 6060 5091 11794 5474 11795 11796 11797 11798 5325 11799 11800 11801 11802 11803
11804 11805 11806 11807 4611 11808 11809 5925 5212 11810 5706 11811 11812 11813 11814 11815 5806 11816 5724
11817 11818 11819 11820 11821 11822 11823 11824 11825 11826 11827 11828 11829 5557 11830 11831 11832 4626 11833
5101 11834 11835 11836 11837 11838 5266 11839 11840 11841 11842 11843 11844 11845 11846 11847 11848 11849 11850
11851 11852 11853 11854 11855 11856 11857 11858 11859 11860 11861 11862 11863 11864 11865 11866 11867 5235 11868
11869 11870 11871 11872 11873 11874 11875 5709 11876 11877 11878 4885 11879 11880 11881 11882 11883 11884 11885
11886 11887 11888 11889 11890 11891 11892 11893 11894 11895 11896 11897 11898 11899 11900 11901 11902 11903 11904
11905 5062 11906 11907 11908 11909 11910 11911 11912 11913 11914 11915 5328 11916 11917 5053 11918 11919 11920
11921 5470 11922 11923 11924 11925 11926 11927 11928 11929 11930 11931 11932 11933 5286 11934 11935 11936 11937
11938 11939 11940 11941 11942 11943 11944 11945 11946 11947 11948 11949 11950 11951 11952 11953 11954 4584 11955
11956 11957 11958 11959 11960 11961 11962 11963 11964 11965 11966 11967 11968 11969 5432 11970 11971 11972 11973
11974 5067 11975 11976 11977 11978 11979 11980 11981 11982 11983 11984 11985 11986 11987 11988 11989 11990 11991
11992 11993 11994 5567 5830 11995 11996 11997 11998 11999 12000 12001 12002 12003 12004 12005 12006 12007 12008
12009 12010 12011 12012 12013 12014 12015 12016 12017 4999 12018 12019 12020 12021 5302 12022 12023 12024 12025
12026 12027 12028 12029 12030 12031 12032 12033 5255 12034 5462 12035 12036 12037 12038 12039 12040 12041 5114
12042 12043 12044 12045 12046 12047 12048 12049 12050 6010 12051 12052 12053 12054 12055 12056 12057 12058 12059

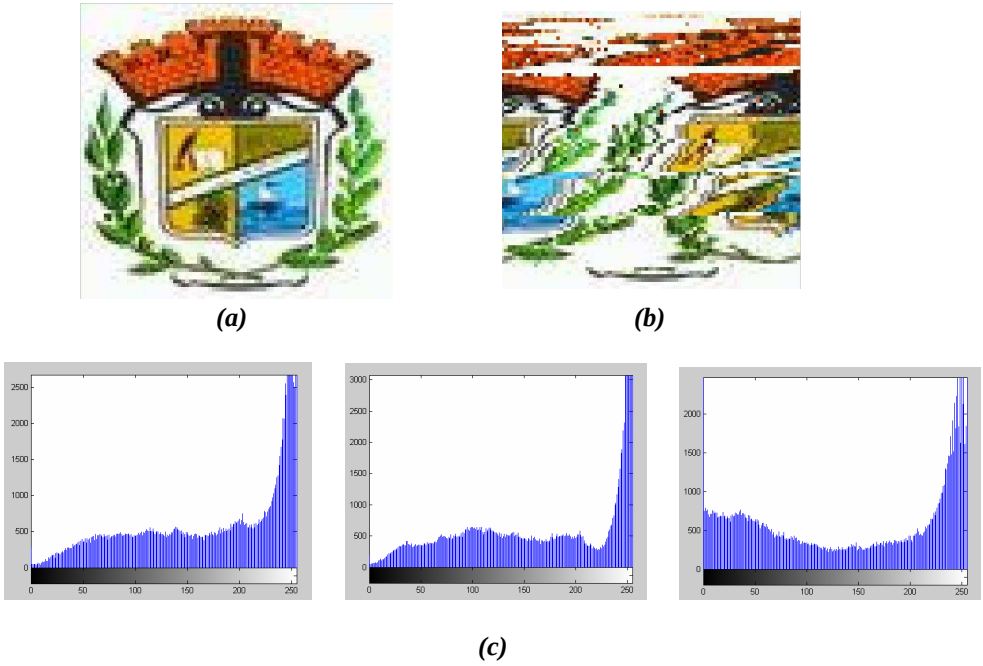
12060 12061 12062 12063 12064 5150 5299 12065 12066 12067 12068 12069 12070 12071 12072 12073 12074 12075 12076
5864 12077 12078 12079 5555 12080 12081 12082 5191 5912 12083 12084 12085 12086 12087 12088 12089 12090 12091
12092 12093 12094 12095 12096 4881 12097 12098 12099 12100 12101 12102 12103 12104 12105 12106 12107 12108 12109
4825 12110 12111 12112 12113 12114 12115 5455 12116 12117 5590 12118 12119 12120 5840 12121 12122 5145 12123
12124 12125 12126 12127 12128 12129 12130 12131 12132 12133 12134 12135 12136 12137 12138 4823 4764 12139 5087
12140 12141 12142 12143 12144 12145 12146 12147 12148 12149 12150 4686 12151 12152 12153 12154 4899 12155 12156
12157 12158 12159 12160 12161 12162 12163 12164 12165 12166 12167 12168 6025 4948 12169 12170 5577 12171 5395
6100 12172 12173 12174 12175 12176 12177 12178 12179 12180 12181 12182 12183 12184 12185 12186 12187 12188 12189
12190 12191 12192 12193 12194 12195 12196 12197 12198 12199 12200 12201 12202 12203 12204 12205 12206 12207 12208
12209 12210 12211 5327 12212 12213 12214 5630 6026 4925 12215 12216 12217 12218 12219 12220 12221 12222 12223
12224 12225 12226 12227 12228 12229 12230 12231 12232 5095 12233 5360 5845 12234 12235 6092 5135 12236 12237
12238 12239 12240 12241 12242 12243 12244 12245 12246 12247 12248 12249 12250 12251 12252 5898 12253 12254 12255
12256 12257 12258 12259 12260 12261 12262 12263 5163 12264 12265 12266 12267 12268 12269 12270 12271 12272 12273
12274 12275 12276 12277 5954 12278 12279 12280 12281 12282 12283 12284 4593 12285 12286 12287 12288 12289 12290
12291 12292 12293 12294 12295 4979 5526 5039 12296 12297 12298 12299 12300 12301 12302 12303 4674 4585 12304
12305 12306 12307 12308 5862 12309 5041 12310 12311 12312 12313 12314 12315 12316 12317 12318 12319 12320 12321
12322 12323 12324 12325 12326 12327 12328 5667 12329 5176 12330 5869 12331 12332 12333 12334 12335 5378 12336
12337 12338 12339 12340 12341 12342 5022 12343 5758 12344 12345 5123 12346 12347 5034 12348 12349 5663 12350
12351 12352 5096 5345 12353 12354 12355 12356 12357 12358 12359 5339 12360 12361 5403 5000 12362 12363 12364
12365 12366 12367 12368 12369 12370 12371 12372 12373 12374 12375 12376 12377 12378 12379 12380 12381 12382 12383
12384 12385 12386 12387 12388 12389 12390 12391 12392 12393 12394 12395 12396 12397 12398 5679 12399 12400 12401
12402 4905 12403 12404 12405 12406 12407 12408 4639 4889 12409 12410 12411 12412 12413 12414 5410 12415 12416
12417 12418 12419 12420 12421 4579 12422 12423 12424 12425 12426 12427 12428 12429 12430 12431 12432 5589 12433
12434 12435 12436 12437 12438 5616 12439 12440 12441 12442 12443 12444 12445 12446 12447 12448 12449 12450 12451
12452 12453 12454 12455 4891 12456 12457 12458 12459 12460 12461 12462 12463 12464 12465 12466 12467 5524 12468
5987 12469 12470 12471 12472 12473 12474 12475 12476 12477 12478 12479 12480 12481 12482 5316 12483 12484 12485
12486 4658 12487 12488 12489 4746 12490 12491 4745 12492 12493 12494 12495 12496 12497 12498 12499 12500 12501
12502 5002 12503 12504 12505 12506 12507 12508 12509 12510 5037 12511 12512 12513 12514 12515 12516 12517 12518
12519 4818 4791 12520 12521 12522 12523 12524 5761 12525 12526 12527 12528 12529 12530 12531 12532 12533 12534
12535 12536 5602 5072 12537 12538 5055 12539 12540 12541 12542 12543 12544 12545 12546 12547 5677 12548 12549
12550 12551 12552 12553 12554 12555 12556 12557 5178 4983 12558 12559 5093 12560 12561 12562 12563 5501 12564
12565 12566 12567 12568 12569 12570 12571 12572 12573 12574 6075 12575 12576 12577 12578 12579 12580 12581 12582
12583 12584 12585 12586 12587 12588 12589 12590 5341 12591 12592 12593 12594 12595 6036 5859 12596 12597 12598
5636 12599 12600 12601 4692 12602 12603 12604 12605 12606 12607 12608 12609 5728 12610 12611 12612 12613 12614
12615 5520 12616 12617 12618 12619 12620 12621 12622 5475 12623 12624 12625 12626 12627 12628 12629 12630 12631
12632 12633 12634 12635 5743 12636 12637 5390 12638 12639 12640 12641 12642 12643 12644 12645 12646 4741 12647
12648 12649 12650 12651 12652 12653 12654 12655 12656 12657 12658 12659 12660 12661 12662 12663 12664 4851 12665
12666 12667 12668 12669 12670 12671 5768 12672 12673 12674 12675 12676 12677 12678 12679 12680 12681 12682 12683
12684 12685 12686 12687 5640 12688 5751 12689 5575 12690 12691 12692 12693 12694 12695 12696 12697 12698 12699
12700 12701 12702 12703 12704 12705 12706 12707 12708 12709 12710 12711 5471 12712 12713 4752 5600 12714 5045
5831 12715 12716 12717 12718 12719 12720 12721 12722 5504 12723 12724 12725 12726 12727 12728 12729 12730 12731
12732 12733 5742 12734 12735 12736 12737 12738 5745 12739 12740 12741 12742 12743 12744 12745 12746 12747 12748
5085 12749 12750 12751 5298 12752 12753 12754 12755 5658 12756 12757 12758 5697 12759 5394 4798 12760 12761
12762 12763 12764 12765 12766 12767 12768 12769 12770 12771 5983 12772 12773 12774 12775 12776 12777 12778 12779
12780 12781 12782 12783 4717 12784 12785 12786 5346 5735 12787 12788 12789 12790 12791 12792 12793 12794 12795
12796 12797 12798 12799 5375 12800 12801 12802 12803 12804 5974 6055 12805 12806 12807 12808 4782 12809 12810
12811 12812 12813 12814 12815 12816 5809 12817 12818 12819 12820 12821 12822 12823 12824 4727 12825 12826 12827
12828 12829 12830 12831 12832 12833 12834 12835 12836 12837 12838 12839 12840 12841 12842 5586 12843 12844 12845
12846 12847 12848 5450 12849 12850 12851 12852 5496 12853 12854 12855 12856 12857 12858 12859 12860 12861 12862
12863 12864 12865 12866 12867 12868 12869 12870 12871 12872 12873 12874 12875 12876 12877 5443 12878 12879 12880
12881 12882 12883 5128 12884 12885 12886 12887 12888 12889 12890 5236 5076 12891 12892 12893 12894 12895 12896
12897 12898 12899 12900 12901 12902 12903 4595 12904 12905 12906 12907 12908 12909 12910 12911 12912 12913 12914
12915 12916 4849 12917 12918 12919 12920 12921 12922 12923 12924 5478 12925 12926 12927 12928 12929 12930 12931
12932 12933 12934 12935 12936 12937 12938 12939 12940 4956 12941 12942 12943 4900 12944 12945 12946 12947 12948
12949 6070 12950 12951 12952 12953 12954 12955 12956 12957 12958 5823 12959 12960 12961 12962 12963 12964 12965
12966 12967 12968 5249 12969 12970 12971 12972 12973 12974 12975 5786 12976 12977 12978 12979 12980 12981 12982
12983 12984 12985 12986 12987 12988 12989 12990 5023 12991 5819 12992 12993 6004 12994 12995 12996 5951 12997
12998 12999 13000 13001 13002 13003 13004 13005 5318 13006 13007 5961 13008 13009 13010 13011 13012 13013 13014
13015 13016 13017 13018 13019 13020 13021 13022 13023 5311 4628 13024 13025 13026 13027 13028 13029 13030 5913
4865 13031 13032 13033 13034 13035 5008 13036 5644 13037 4985 13038 13039 13040 6028 5479 13041 13042 13043
13044 13045 13046 13047 13048 13049 13050 13051 13052 13053 13054 13055 13056 13057 13058 13059 13060 13061 13062
13063 13064 13065 13066 13067 13068 13069 13070 13071 13072 13073 13074 13075 13076 13077 13078 13079 13080 5194
13081 13082 13083 13084 13085 13086 5444 13087 13088 13089 5226 13090 13091 13092 13093 13094 13095 13096 13097
13098 13099 4651 13100 13101 13102 13103 13104 5141 4982 13105 13106 13107 13108 13109 6012 13110 13111 13112
13113 13114 13115 13116 13117 13118 13119 13120 13121 13122 13123 13124 13125 13126 5374 13127 13128 13129 13130
13131 13132 13133 4989 13134 13135 13136 13137 13138 5825 13139 13140 13141 13142 13143 13144 13145 5712 13146
5625 13147 13148 13149 13150 13151 13152 13153 13154 13155 5812 13156 13157 13158 13159 13160 13161 5247 13162
13163 13164 13165 13166 13167 13168 13169 13170 13171 13172 13173 13174 13175 13176 13177 4777 13178 13179 13180
13181 13182 13183 5215 5937 13184 13185 13186 13187 13188 13189 13190 13191 13192 6108 13193 13194 4707 13195
13196 13197 13198 13199 5943 13200 13201 13202 13203 4894 13204 13205 13206 13207 13208 13209 13210 13211 13212
13213 13214 13215 13216 13217 13218 13219 13220 13221 13222 13223 13224 13225 13226 13227 13228 13229 4729 13230
5899 13231 13232 13233 13234 13235 13236 13237 13238 13239 13240 5218 13241 13242 13243 13244 13245 13246 13247
5453 13248 13249 13250 13251 13252 4813 13253 13254 13255 13256 13257 13258 13259 13260 13261 13262 13263 13264
13265 13266 13267 13268 13269 4837 13270 13271 13272 13273 13274 13275 13276 13277 13278 13279 13280 13281 5036
13282 13283 13284 13285 13286 13287 13288 13289 13290 5441 13291 13292 13293 13294 4590 13295 13296 13297 13298
13299 13300 13301 13302 5169 13303 13304 13305 13306 13307 13308 13309 13310 13311 13312 13313 13314 13315 13316

13317 13318 13319 13320 13321 13322 13323 13324 13325 13326 13327 13328 13329 13330 13331 13332 5402 13333 13334
13335 13336 13337 13338 13339 13340 13341 13342 5347 13343 13344 13345 13346 13347 13348 13349 13350 13351 13352
13353 13354 13355 5922 13356 13357 13358 13359 13360 4783 13361 5369 13362 13363 13364 13365 13366 13367 13368
13369 13370 13371 13372 13373 13374 5323 13375 13376 13377 13378 13379 13380 13381 5592 13382 13383 13384 13385
13386 13387 5563 13388 13389 13390 13391 13392 13393 13394 13395 13396 13397 13398 13399 13400 13401 13402 13403
13404 13405 13406 13407 5268 13408 6035 13409 5111 13410 13411 4653 13412 4906 13413 13414 13415 13416 13417
4636 4797 13418 13419 13420 13421 13422 13423 13424 13425 13426 13427 13428 13429 13430 13431 13432 13433 13434
13435 13436 13437 13438 13439 13440 13441 13442 13443 13444 13445 13446 13447 13448 13449 13450 5820 13451 13452
13453 13454 13455 13456 13457 13458 13459 13460 13461 13462 13463 13464 13465 13466 13467 13468 13469 13470 13471
13472 13473 13474 13475 13476 13477 5558 13478 13479 5764 13480 13481 13482 13483 13484 13485 13486 13487 4681
13488 13489 13490 13491 13492 13493 13494 13495 13496 5854 13497 13498 13499 13500 4884 13501 4800 13502 13503
13504 13505 13506 13507 13508 13509 13510 13511 13512 13513 13514 13515 13516 13517 13518 13519 13520 13521 13522
13523 13524 13525 13526 13527 13528 5886 13529 5670 13530 13531 13532 13533 5791 13534 13535 13536 13537 13538
13539 13540 13541 13542 13543 13544 13545 13546 13547 13548 13549 13550 13551 13552 13553 13554 13555 13556 13557
13558 13559 13560 13561 5883 13562 13563 13564 13565 13566 13567 13568 13569 13570 13571 13572 13573 13574 13575
13576 13577 13578 13579 13580 5001 13581 13582 13583 13584 4821 13585 13586 13587 13588 13589 13590 13591 13592
13593 13594 13595 13596 13597 13598 13599 13600 13601 13602 13603 13604 13605 13606 5425 13607 13608 13609 13610
13611 13612 13613 13614 13615 13616 13617 13618 13619 13620 13621 6034 13622 13623 13624 13625 13626 13627 13628
13629 13630 13631 4765 4642 13632 4827 13633 13634 6039 4836 13635 13636 5117 13637 13638 5837 13639 13640
13641 13642 13643 13644 13645 13646 13647 13648 13649 13650 13651 13652 13653 13654 13655 13656 13657 5551 13658
13659 13660 13661 5267 13662 13663 13664 5904 13665 13666 13667 13668 13669 13670 13671 13672 13673 13674 13675
13676 13677 13678 13679 5385 13680 13681 13682 5595 13683 13684 13685 13686 13687 13688 13689 13690 13691 13692
13693 13694 13695 5920 13696 13697 13698 13699 13700 13701 13702 13703 13704 13705 13706 13707 13708 4605 13709
13710 13711 13712 13713 13714 13715 13716 13717 5963 13718 13719 13720 13721 13722 5387 13723 13724 13725 13726
13727 13728 13729 13730 13731 13732 13733 5903 13734 13735 13736 13737 13738 5817 13739 5408 5269 13740 13741
13742 13743 13744 13745 13746 13747 13748 13749 13750 13751 13752 13753 13754 13755 13756 13757 13758 13759 13760
13761 13762 5646 13763 13764 13765 13766 13767 13768 13769 13770 13771 13772 13773 13774 13775 13776 13777 13778
13779 5597 13780 13781 13782 13783 4673 5956 13784 13785 13786 13787 13788 13789 13790 13791 13792 13793 4696
13794 13795 13796 5497 13797 13798 4712 5799 13799 13800 13801 13802 13803 13804 13805 13806 13807 13808 13809
13810 13811 13812 13813 13814 13815 13816 13817 13818 13819 13820 13821 13822 13823 13824 13825 13826 13827 13828
13829 5442 4701 13830 4859 13831 6064 13832 13833 13834 13835 13836 13837 13838 13839 13840 13841 13842 13843
13844 13845 13846 13847 13848 13849 13850 13851 13852 13853 13854 13855 13856 13857 5186 13858 13859 13860 13861
13862 13863 13864 5431 13865 13866 13867 13868 5449 13869 4995 13870 13871 13872 13873 13874 13875 13876 5604
13877 13878 13879 5406 13880 13881 13882 13883 13884 13885 13886 13887 13888 13889 13890 13891 4863 13892 13893
5695 13894 13895 13896 13897 13898 13899 13900 13901 13902 13903 13904 13905 5815 13906 13907 13908 13909 13910
13911 13912 13913 13914 5571 13915 13916 13917 5208 13918 13919 13920 5109 13921 13922 13923 13924 13925 13926
13927 13928 13929 13930 13931 13932 13933 13934 13935 13936 13937 13938 13939 13940 5224 13941 13942 13943 13944
13945 13946 13947 5407 13948 13949 13950 13951 5189 4620 13952 4582 13953 13954 13955 13956 13957 13958 13959
13960 13961 13962 13963 13964 13965 13966 13967 13968 13969 13970 13971 5536 13972 13973 13974 13975 13976 13977
13978 13979 13980 13981 13982 5533 13983 13984 13985 5042 13986 13987 13988 13989 13990 13991 5776 13992 13993
13994 13995 4803 13996 13997 5167 13998 4743 13999 14000 14001 14002 14003 14004 14005 14006 14007 14008 14009
14010 14011 14012 14013 14014 14015 14016 14017 5274 14018 14019 14020 14021 14022 14023 14024 14025 14026 5011
14027 14028 14029 14030 5168 14031 14032 5550 14033 14034 14035 5251 14036 14037 14038 14039 14040 14041 14042
5014 14043 14044 14045 14046 14047 14048 14049 4990 14050 14051 14052 5933 14053 14054 14055 14056 14057 14058
5813 14059 14060 14061 14062 14063 5015 14064 14065 14066 5531 14067 14068 14069 14070 14071 14072 14073 14074
14075 14076 14077 14078 14079 14080 14081 14082 14083 14084 14085 14086 14087 14088 14089 5716 14090 14091 5924
14092 14093 14094 14095 14096 5749 14097 14098 14099 14100 14101 14102 14103 14104 14105 14106 14107 14108 14109
14110 14111 14112 4918 5619 14113 14114 14115 5593 14116 14117 5166 14118 14119 14120 14121 14122 14123 14124
14125 14126 14127 14128 14129 14130 14131 14132 14133 14134 14135 14136 14137 14138 14139 14140 14141 14142 14143
14144 14145 14146 14147 14148 14149 14150 14151 14152 14153 14154 14155 14156 14157 14158 14159 14160 14161 14162
14163 14164 14165 14166 5081 14167 14168 14169 14170 14171 14172 4852 5256 14173 14174 14175 14176 14177 14178
14179 4872 14180 14181 4996 14182 14183 14184 14185 14186 14187 14188 5580 14189 14190 14191 14192 14193 14194
14195 14196 14197 14198 14199 5472 5231 14200 14201 14202 14203 14204 14205 14206 14207 14208 14209 14210 5232
4750 14211 14212 14213 14214 5538 14215 14216 14217 14218 14219 14220 5788 14221 14222 14223 14224 4676 14225
14226 14227 14228 14229 14230 14231 14232 14233 14234 14235 4720 14236 14237 14238 4987 14239 14240 14241 14242
4955 14243 14244 14245 14246 14247 14248 14249 14250 14251 14252 14253 14254 14255 5004 14256 14257 14258 14259
14260 14261 14262 5918 14263 14264 14265 14266 14267 14268 14269 14270 5197 14271 14272 5866 14273 14274 14275
14276 14277 4789 14278 14279 14280 14281 14282 14283 14284 14285 14286 14287 14288 14289 14290 14291 14292 14293
14294 14295 14296 14297 14298 14299 14300 14301 14302 5241 14303 14304 14305 14306 14307 14308 14309 14310 14311
14312 14313 14314 6058 14315 14316 14317 5642 14318 14319 14320 14321 14322 14323 14324 14325 14326 14327 6040
14328 14329 5569 14330 14331 14332 14333 14334 14335 14336 14337 14338 4631 5315 14339 14340 14341 14342 14343
14344 14345 14346 14347 14348 14349 14350 14351 14352 14353 14354 14355 14356 14357 14358 14359 14360 14361 14362
14363 14364 5534 14365 14366 4775 14367 5993 14368 14369 14370 14371 14372 14373 6085 14374 14375 14376 5337
14377 5708 14378 14379 14380 14381 14382 14383 14384 14385 6069 14386 14387 14388 14389 14390 14391 14392 14393
14394 14395 14396 14397 14398 14399 14400 14401 14402 14403 14404 14405 14406 14407 14408 14409 5529 14410 14411
14412 14413 14414 14415 14416 14417 14418 14419 14420 14421 14422 14423 14424 14425 14426 14427 14428 5755 5732
14429 5356 14430 14431 14432 5192 14433 14434 14435 5733 14436 5351 14437 14438 14439 14440 14441 14442 14443
14444 14445 14446 14447 5598 4816 14448 14449 14450 14451 5603 14452 14453 14454 5448 14455 14456 14457 14458
14459 14460 14461 14462 14463 14464 14465 14466 14467 14468 14469 14470 5765 5370 4965 4715 14471 5416 14472
14473 14474 14475 14476 4805 14477 14478 14479 14480 14481 14482 14483 14484 14485 14486 14487 14488 14489 5503
14490 14491 14492 5982 14493 4678 14494 14495 14496 14497 14498 14499 14500 14501 14502 14503 14504 14505 14506
14507 14508 5822 14509 14510 14511 14512 14513 14514 14515 14516 5591 5361 14517 14518 14519 14520 14521 14522
14523 14524 14525 14526 14527 14528 14529 14530 14531 14532 14533 14534 14535 14536 14537 14538 14539 14540 5278
14541 14542 14543 14544 14545 14546 14547 14548 14549 14550 14551 14552 14553 14554 14555 14556 14557 14558 14559
14560 14561 14562 14563 4921 14564 14565 14566 14567 14568 14569 14570 14571 14572 14573 14574 14575 4988 14576
14577 14578 14579 14580 14581 14582 14583 14584 14585 5513 14586 14587 14588 14589 14590 14591 5875 5050 14592

14593 14594 14595 14596 14597 14598 14599 14600 14601 14602 14603 14604 5572 14605 14606 14607 14608 14609 14610
5364 14611 14612 4645 14613 14614 14615 14616 14617 14618 14619 14620 14621 14622 14623 14624 14625 14626 14627
14628 14629 14630 14631 14632 14633 14634 14635 14636 14637 14638 14639 14640 14641 14642 14643 14644 14645 14646
14647 14648 14649 14650 14651 14652 14653 14654 14655 14656 14657 14658 14659 14660 14661 14662 14663 5512 14664
14665 14666 14667 14668 5836 14669 14670 14671 14672 5737 14673 14674 14675 14676 4842 14677 14678 14679 14680
14681 14682 4737 14683 14684 4833 14685 14686 14687 14688 5359 14689 14690 14691 14692 14693 14694 14695 14696
4650 14697 5293 14698 14699 14700 14701 14702 14703 5063 14704 5941 14705 14706 14707 14708 5938 14709 14710
14711 14712 5710 14713 14714 14715 14716 14717 14718 6086 14719 14720 14721 14722 14723 14724 14725 14726 14727
14728 14729 14730 14731 14732 14733 14734 14735 14736 14737 14738 14739 14740 14741 14742 14743 14744 14745 14746
14747 14748 14749 14750 14751 14752 14753 14754 14755 14756 14757 14758 14759 5165 14760 14761 14762 14763 14764
14765 14766 6048 14767 14768 14769 6071 14770 14771 14772 14773 14774 14775 14776 14777 14778 14779 14780 14781
14782 6073 14783 14784 14785 14786 14787 14788 6101 14789 14790 4854 14791 14792 14793 14794 14795 14796 14797
14798 4617 14799 5753 14800 14801 14802 14803 14804 14805 14806 14807 14808 14809 14810 14811 14812 14813 14814
14815 14816 14817 14818 14819 14820 14821 14822 14823 14824 14825 4598 14826 14827 5233 14828 14829 14830 14831
14832 14833 14834 14835 14836 14837 14838 14839 14840 5017 14841 14842 14843 14844 14845 14846 14847 14848 14849
5107 14850 14851 14852 14853 14854 14855 14856 14857 14858 14859 14860 14861 14862 14863 14864 14865 14866 14867
5752 5511 5522 14868 14869 14870 14871 14872 14873 14874 14875 5313 14876 14877 14878 14879 14880 14881 14882
4796 14883 14884 14885 14886 14887 4840 14888 14889 14890 14891 14892 14893 14894 14895 14896 14897 14898 14899
14900 14901 14902 14903 14904 14905 14906 14907 14908 14909 14910 14911 14912 14913 5152 14914 14915 14916 4687
14917 14918 14919 14920 4722 14921 14922 14923 14924 5537 14925 14926 14927 14928 14929 14930 14931 14932 14933
14934 14935 14936 14937 14938 14939 14940 14941 14942 14943 14944 14945 14946 14947 14948 14949 14950 14951 14952
14953 14954 14955 14956 14957 14958 14959 4951 4759 14960 14961 14962 14963 14964 5144 5464 14965 14966 14967
14968 14969 14970 14971 14972 14973 14974 14975 6042 14976 14977 14978 14979 14980 14981 14982 14983 14984 14985
14986 14987 14988 14989 6066 14990 14991 14992 14993 14994 14995 14996 14997 14998 14999 15000 15001 15002 15003
15004 15005 15006 15007 15008 15009 15010 15011 15012 15013 15014 15015 15016 15017 5044 15018 15019 15020 15021
15022 15023 5826 15024 15025 15026 15027 15028 15029 15030 15031 5544 15032 5219 15033 15034 15035 15036 5396
15037 15038 15039 15040 15041 15042 15043 15044 15045 15046 15047 4714 15048 15049 15050 15051 15052 15053 15054
15055 4922 15056 15057 15058 15059 15060 15061 15062 15063 15064 15065 15066 15067 5222 4855 15068 15069 15070
15071 4937 15072 15073 4618 15074 15075 15076 5587 15077 6089 15078 15079 15080 15081 5656 15082 15083 15084
15085 15086 15087 15088 15089 15090 5578 15091 15092 15093 15094 15095 15096 15097 4643 5605 15098 15099 15100
15101 15102 15103 15104 15105 15106 15107 15108 15109 15110 15111 15112 15113 6067 15114 15115 15116 15117 15118
15119 15120 15121 15122 15123 15124 15125 15126 5860 15127 15128 15129 15130 15131 15132 15133 15134 15135 15136
15137 15138 5953 15139 15140 15141 15142 15143 15144 15145 15146 5030 15147 15148 15149 15150 15151 5006 5329
5121 15152 15153 15154 15155 4762 15156 15157 15158 5730 15159 5013 15160 15161 15162 4917 15163 15164 15165
15166 15167 15168 15169 15170 15171 15172 15173 15174 15175 15176 15177 15178 15179 15180 6054 15181 15182 15183
15184 15185 15186 15187 15188 5611 15189 15190 15191 15192 5824 15193 15194 15195 5419 15196 15197 15198 15199
15200 15201 15202 15203 15204 15205 15206 15207 15208 15209 15210 4685 15211 15212 15213 15214 15215 15216 5909
15217 15218 15219 15220 15221 15222 15223 15224 15225 15226 15227 15228 5787 15229 15230 15231 15232 15233 15234
15235 15236 15237 5389 15238 15239 15240 15241 15242 15243 15244 15245 15246 15247 4683 4832 15248 15249 15250
5494 15251 5687 15252 15253 15254 5723 15255 15256 15257 15258 15259 15260 15261 15262 15263 4757 15264 4754
15265 15266 15267 15268 15269 15270 15271 15272 15273 15274 15275 15276 15277 15278 15279 15280 15281 15282 15283
15284 4591 15285 15286 15287 15288 15289 15290 15291 15292 15293 6077 15294 15295 15296 15297 4606 15298 15299
15300 5404 15301 15302 15303 15304 15305 15306 15307 4670 15308 6098 15309 15310 4589 15311 15312 15313 15314
6074 15315 15316 15317 15318 15319 15320 15321 15322 15323 15324 15325 15326 15327 15328 15329 15330 15331 4583
15332 15333 15334 15335 15336 5888 15337 15338 15339 15340 15341 15342 15343 15344 15345 15346 5970 15347 15348
15349 15350 15351 15352 15353 5688 15354 15355 15356 15357 15358 15359 15360 15361 15362 15363 15364 15365 15366
15367 15368 15369 15370 15371 15372 15373 15374 15375 15376 15377 5720 15378 15379 15380 15381 15382 15383 15384
15385 15386 15387 5423 15388 15389 15390 15391 15392 15393 15394 15395 15396 15397 15398 4817 5793 4698 15399
15400 15401 15402 15403 5210 15404 5574 15405 15406 5740 4941 15407 15408 15409 15410 6109 15411 15412 15413
15414 15415 15416 15417 15418 15419 15420 15421 15422 15423 15424 15425 15426 5202 15427 15428 15429 15430 15431
15432 15433 15434 15435 15436 15437 15438 5211 15439 15440 15441 5073 15442 15443 5779 15444 5696 15445 15446
15447 4629 5398 15448 15449 15450 15451 15452 15453 15454 15455 15456 15457 15458 15459 15460 15461 15462 15463
15464 15465 15466 15467 15468 15469 15470 15471 15472 15473 15474 15475 15476 4845 15477 15478 15479 15480 5772
15481 5596 15482 15483 15484 15485 15486 15487 15488 5949 15489 15490 15491 15492 15493 15494 15495 6021 15496
15497 15498 15499 15500 15501 15502 15503 15504 15505 5818 15506 15507 15508 15509 5326 15510 15511 15512 15513
15514 15515 4994 15516 15517 15518 15519 15520 15521 15522 5154 15523 4663 15524 15525 15526 5734 15527 15528
15529 15530 15531 15532 5059 15533 15534 4943 15535 15536 15537 15538 15539 15540 15541 15542 15543 15544 15545
5273 15546 15547 15548 15549 15550 15551 15552 4760 15553 5399 15554 6056 15555 15556 15557 5303 15558 15559
15560 15561 15562 15563 15564 5705 15565 15566 15567 15568 5852 5842 15569 15570 15571 15572 5838 15573 15574
15575 15576 15577 15578 5981 15579 15580 15581 15582 15583 15584 15585 15586 15587 15588 15589 15590 15591 15592
15593 15594 5350 15595 15596 15597 15598 5122 15599 15600 15601 4774 15602 15603 15604 15605 15606 15607 15608
15609 15610 15611 15612 15613 15614 15615 15616 15617 15618 15619 15620 15621 5280 15622 5748 15623 15624 15625
15626 15627 15628 15629 5996 15630 5863 15631 15632 15633 15634 15635 15636 15637 15638 15639 15640 15641 15642
15643 15644 5653 15645 15646 15647 15648 4940 15649 15650 15651 15652 15653 15654 15655 15656 15657 15658 15659
15660 15661 15662 15663 15664 5876 15665 15666 15667 15668 15669 15670 15671 15672 15673 5984 15674 15675 15676
15677 15678 15679 15680 5725 15681 15682 15683 15684 15685 15686 15687 15688 15689 15690 15691 15692 15693 15694
15695 15696 15697 5659 15698 15699 15700 15701 15702 15703 15704 15705 15706 15707 15708 15709 15710 5214 15711
15712 15713 15714 15715 15716 4592 15717 15718 5777 15719 15720 15721 15722 5271 15723 15724 15725 15726 15727
15728 15729 15730 4870 15731 15732 15733 15734 15735 15736 15737 15738 15739 15740 15741 15742 15743 15744 15745
15746 15747 15748 15749 15750 15751 15752 15753 15754 15755 15756 15757 15758 15759 15760 15761 15762 15763 15764
15765 15766 15767 15768 15769 15770 15771 15772 15773 15774 4647 15775 15776 15777 4931 15778 15779 15780 15781
15782 15783 4672 15784 15785 15786 15787 15788 15789 15790 15791 15792 15793 15794 15795 15796 15797 15798 15799
15800 15801 15802 15803 15804 15805 15806 15807 15808 15809 15810 4697 15811 15812 15813 15814 15815 15816 15817
15818 15819 15820 5223 15821 15822 15823 15824 15825 15826 4841 15827 15828 15829 5711 15830 4878 15831 15832
15833 15834 15835 15836 15837 15838 15839 4875 15840 15841 15842 15843 15844 15845 15846 15847 15848 15849 15850
15851 15852 15853 15854 15855 15856 15857 15858 15859 15860 15861 15862 15863 4721 4807 15864 15865 15866 15867

15868 15869 15870 5099 5421 15871 15872 15873 15874 15875 15876 15877 15878 15879 15880 15881 15882 15883 15884
15885 4749 15886 15887 15888 15889 15890 15891 15892 15893 15894 15895 4682 15896 15897 15898 15899 15900 15901
15902 5853 15903 15904 5521 15905 15906 5770 15907 15908 15909 15910 15911 15912 15913 15914 15915 15916 15917
5565 15918 15919 15920 15921 15922 15923 15924 15925 15926 15927 15928 15929 15930 15931 5991 15932 15933 15934
15935 15936 15937 15938 15939 15940 5879 15941 15942 15943 15944 15945 15946 15947 15948 15949 15950 15951 15952
15953 15954 4621 15955 15956 15957 15958 4744 15959 15960 15961 5584 15962 15963 15964 15965 15966 15967 4788
15968 15969 5220 15970 15971 15972 15973 15974 15975 15976 15977 15978 15979 5415 5798 15980 15981 15982 15983
15984 15985 15986 15987 15988 15989 15990 15991 15992 15993 15994 15995 15996 15997 15998 5254 15999 16000 16001
16002 16003 16004 16005 16006 16007 16008 16009 16010 16011 16012 5029 16013 16014 16015 16016 16017 16018 4880
16019 5510 16020 16021 16022 16023 5914 16024 16025 5332 16026 5485 16027 16028 5456 16029 16030 5707 16031
16032 16033 16034 16035 16036 16037 16038 16039 16040 5317 16041 16042 16043 4966 16044 6068 16045 16046 16047
16048 16049 16050 16051 16052 16053 5102 16054 16055 16056 16057 16058 16059 16060 16061 16062 16063 16064 16065
16066 16067 16068 16069 16070 16071 16072 16073 16074 16075 16076 16077 5376 16078 16079 16080 16081 16082 16083
16084 16085 16086 16087 16088 5032 5305 16089 16090 16091 16092 16093 16094 16095 16096 16097 16098 16099 16100
5463 16101 16102 16103 16104 5228 4895 16105 16106 16107 16108 16109 16110 16111 16112 16113 16114 16115 16116
16117 16118 16119 16120 4997 16121 16122 4864 16123 16124 16125 16126 16127 16128 16129 5618 5467 6065 16130
16131 16132 16133 16134 16135 16136 16137 16138 16139 16140 16141 16142 16143 16144 16145 16146 16147 16148 16149
16150 16151 16152 16153 16154 16155 16156 16157 16158 16159 16160 16161 16162 16163 5480 16164 16165 16166 16167
16168 16169 16170 16171 16172 16173 16174 16175 5078 16176 16177 16178 16179 16180 16181 16182 16183 16184 16185
16186 5782 16187 6049 16188 16189 16190 16191 16192 5243 16193 16194 16195 16196 16197 4839 16198 16199 16200
16201 16202 16203 16204 16205 16206 16207 16208 16209 16210 16211 4596 16212 16213 16214 16215 16216 16217 16218
16219 16220 16221 16222 16223 5489 16224 16225 16226 5391 16227 16228 16229 16230 16231 16232 16233 16234 16235
16236 16237 16238 16239 16240 16241 16242 4574 5160 5340 16243 16244 16245 16246 16247 5458 16248 16249 5692
16250 16251 16252 16253 16254 16255 16256 16257 16258 16259 5543 16260 16261 5333 16262 16263 16264 16265 16266
16267 16268 16269 16270 5810 16271 16272 16273 16274 16275 16276 16277 16278 16279 16280 16281 16282 16283 16284
16285 16286 16287 16288 16289 16290 16291 16292 16293 16294 16295 16296 16297 16298 16299 4916 5433 5872 16300
16301 16302 16303 16304 16305 16306 16307 16308 16309 16310 16311 16312 16313 6053 16314 16315 6016 16316 16317
16318 16319 16320 16321 16322 5365 16323 16324 4978 16325 16326 16327 16328 16329 16330 16331 16332 5246 4790
16333 16334 5033 16335 16336 16337 16338 16339 16340 16341 5270 16342 16343 16344 16345 16346 16347 16348 16349
16350 16351 16352 5069 16353 5401 16354 16355 16356 5615 16357 16358 16359 16360 4977 16361 16362 16363 16364
16365 16366 16367 16368 16369 16370 16371 16372 16373 16374 16375 16376 16377 16378 16379 16380 16381 16382 16383
16384 5320 16385 16386 4733 16387 16388 5417 16389 16390 16391 16392 16393 16394 16395 16396 16397 16398 16399
16400 16401 16402 16403 16404 16405 16406 16407 4826 16408 16409 4616 16410 16411 16412 16413 16414 16415 16416
16417 16418 16419 16420 5294 16421 5977 16422 5796 16423 16424 16425 16426 16427 16428 16429 16430 16431 5469
16432 16433 16434 16435 16436 16437 16438 16439 16440 16441 16442 16443 16444 16445 16446 16447 16448 16449 16450
5832 16451 16452 16453 16454 16455 16456 16457 16458 16459 16460 16461 16462 16463 16464 16465 5669 16466 16467
16468 16469 16470 16471 16472 16473 6017 16474 16475 16476 16477 16478 5959 16479 16480 16481 16482 16483 16484
16485 16486 16487 16488 16489 5153 16490 16491 16492 16493 16494 16495 16496 4706 16497 16498 16499 16500 16501
5498 16502 16503 16504 16505 5238 16506 16507 16508 16509 16510 16511 16512 16513 16514 16515 16516 16517 16518
16519 16520 16521 16522 16523 16524 16525 16526 16527 16528 16529 16530 16531 16532 16533 6062 16534 16535 16536
16537 16538 5634 16539 16540 16541 16542 5314 16543 16544 16545 16546 16547 16548 5409 16549 16550 5944 4767
16551 16552 16553 16554 16555 5547 16556 16557 16558 16559 16560 4627 16561 16562 16563 5393 16564 16565 16566
16567 16568 16569 16570 16571 16572 16573 16574 16575 16576 16577 16578 16579 16580 16581 16582 16583 5766 5585
16584 16585 5384 5207 16586 16587 16588 16589 16590 16591 16592 16593 5090 16594 16595 16596 16597 5104 16598
16599 16600 16601 16602 16603 16604 16605 16606 16607 16608 16609 16610 16611 16612 16613 16614 16615 6018 16616
16617 16618 16619 16620 4848 16621 16622 16623 16624 16625 16626 16627 16628 16629 16630 16631 16632 16633 16634
16635 16636 16637 16638 16639 16640 16641 16642 16643 16644 16645 16646 16647 16648 5662 16649 16650 16651 16652
16653 16654 16655 16656 16657 16658 16659 16660 16661 16662 16663 16664 16665 16666 16667 16668 16669 4601 16670
16671 16672 16673 16674 16675 16676 16677 16678 16679 16680 16681 16682 16683 16684 16685 16686 16687 16688 16689
16690 16691 16692 16693 16694 5484 16695 5877 16696 16697 16698 16699 16700 16701 16702 16703 16704 16705 16706
16707 16708 16709 16710 16711 16712 16713 16714 16715 16716 16717 16718 5381 16719 16720 16721 16722 16723 16724
16725 16726 16727 16728 16729 16730 16731 16732 16733 6002 16734 16735 16736 16737 16738 16739 5568 16740 16741
16742 5221 16743 16744 16745 16746 16747 16748 16749 4660 16750 16751 16752 16753 16754 16755 16756 16757 16758
5997 16759 16760 16761 16762 16763 5633 16764 16765 6019 16766 16767 16768 5052 16769 16770 16771 16772 16773
16774 16775 16776 16777 16778 16779 16780 16781 16782 16783 16784 16785 16786 16787 16788 16789 16790 16791 16792
16793 16794 16795 16796 16797 16798 16799 16800 16801 16802 16803 16804 16805 16806 16807 16808 16809 16810 16811
16812 16813 16814 16815 16816 16817 16818 16819 16820 16821 16822 16823 5960 16824 16825 16826 16827 16828 16829
16830 6043 16831 16832 5855 16833 16834 16835 16836 16837 5134 16838 16839 16840 16841 16842 16843 16844 16845
4728 16846 16847 16848 16849 16850 16851 16852 16853 16854 16855 6103 16856 16857 16858 5138 16859 16860 5064
16861 16862 16863 16864 16865 16866 16867 16868 16869 16870 16871 4773 16872 16873 16874 16875 16876 16877 16878
6093 16879 16880 16881 16882 16883 16884 16885 16886 4811 16887 16888 16889 16890 16891 16892 16893 16894 16895
16896 16897 16898 16899 16900 16901 16902 16903 16904 16905 5321 6095 16906 16907 16908 16909 16910 16911 16912
16913 16914 16915 5366 16916 4799 16917 16918 16919 16920 16921 16922 16923 16924 16925 16926 5926 16927 16928
5986 16929 16930 16931 16932 6081 16933 16934 16935 16936 16937 16938 16939 16940 16941 16942 16943 16944 16945
16946 16947 16948 5382 16949 16950 16951 16952 16953 16954 16955 16956 16957 16958 16959 16960 16961 16962 16963
16964 16965 16966 16967 16968 16969 16970 16971 16972 16973 16974 16975 16976 16977 16978 16979 16980 16981 16982
16983 16984 16985 5930 16986 6030 16987 16988 16989 16990 16991 16992 16993 16994 5115 16995 16996 16997 16998
16999 17000 17001 17002 17003 17004 4838 17005 17006 17007 17008 17009 17010 17011 17012 17013 17014 17015 17016
17017 17018 17019 17020 17021 5614 17022 17023 17024 17025 17026 17027 17028 17029 17030 17031 17032 17033 17034
17035 17036 17037 17038 17039 17040 17041 5923 17042 17043 17044 17045 17046 17047 17048 17049 5205 17050 5418
17051 17052 5259 17053 17054 17055 17056 17057 17058 17059 17060 17061 17062 17063 17064 17065 17066 17067 17068
17069 17070 17071 17072 17073 17074 17075 17076 17077 17078 17079 17080 17081 17082 17083 17084 17085 17086 17087
17088 17089 17090 17091 17092 17093 17094 17095 17096 17097 17098 17099 4679 17100 17101 17102 17103 17104 17105
17106 17107 17108 17109 17110 17111 17112 17113 17114 17115 17116 17117 17118 17119 5354 5430 17120 17121 17122
17123 17124 17125 17126 17127 4711 17128 17129 17130 17131 17132 17133 17134 17135 17136 17137 17138 17139 17140

17141 17142 5272 17143 17144 17145 17146 17147 17148 17149 17150 17151 17152 17153 17154 17155 17156 17157 17158 17159 6013 17160 17161



(a) ***(b)***

(c)

Figure III.12. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,* ***(c)*** *Histogrammes de l'image chiffrée.*

III.4.1.2. Description de PosESecL2

Comme la clé générée par notre première alternative codant le premier niveau de sécurisation, est de taille variable d'une donnée à une autre et qui est égale à la taille de la donnée en terme d'éléments (caractères ou pixels), telle que : taille de la donnée = n éléments => taille de la clé générée = n nombres ; alors la résistibilité à l'attaque exhaustive est strictement dépendante de la taille de la donnée. C'est pourquoi, nous proposons, ici, une variante de PosESecL1 et que nous l'appelons PosESecL2 (Position based Evolutionary Secure Level 2).

L'évolution vers la solution optimale (donnée cryptée) par PosESecL2 est assurée par application des opérateurs de reproduction (croisement, mutation) et de sélection des meilleurs individus. Dans ce qui suit nous ne présentons que les phases du processus évolutionnaire faisant la différence entre les deux algorithmes PosESecL1 et PosESecL2. Il s'agit, principalement, des phases de codage et de reproduction. Toutefois, le même mécanisme est adopté pour créer la population initiale en permutant aléatoirement le chromosome codant l'image originale.

a. Codage

Dans le but d'augmenter le niveau de sécurité du premier algorithme décrit dans la section précédente, nous présentons, ici, le nouveau codage proposé donnant lieu à un nouvel algorithme de chiffrement : PosESecL2. Ce dernier opère sur le codage des différents éléments de la donnée à chiffrer : dans le cas d'une donnée texte et de même que pour PosESecL1, le codage utilisé est l'hexadécimal, toutefois, chaque quatre bits des 16 bits codant un élément sont considérés comme un gène. Pour une donnée image, chacune des

composantes R, V et B est considérée comme étant un gène. Ainsi, la taille de la nouvelle clé sera plus grande que celle générée par le premier algorithme. Elle est d'une taille quatre fois plus grande dans le cas de manipulation des données texte et de trois fois plus grande dans le cas de manipulation d'une donnée image. Le codage proposé est résumé à travers la figureIII.13.

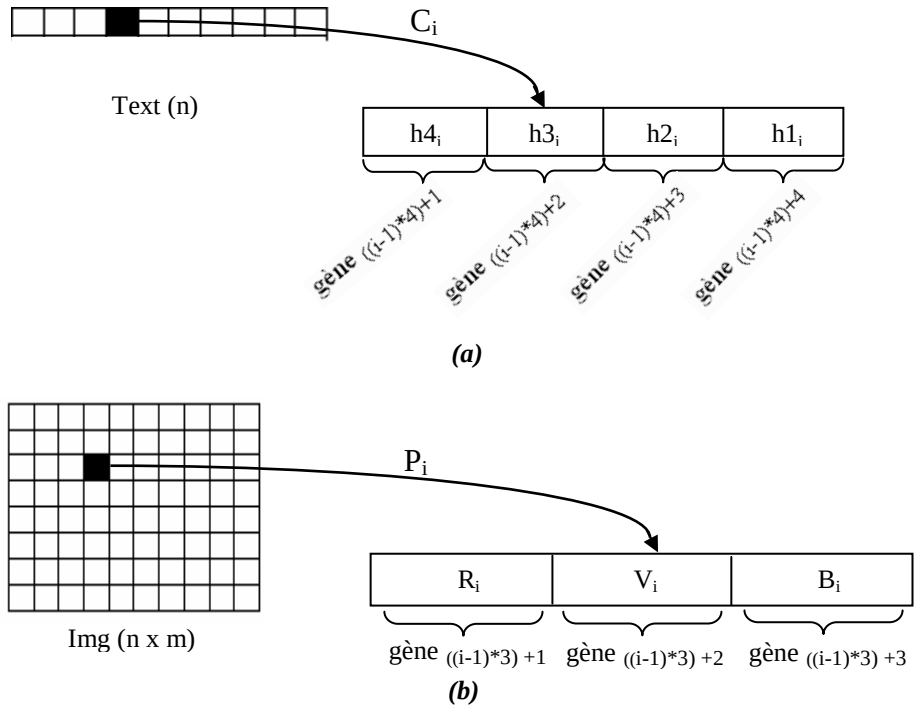


Figure III.13. **(*a*)** *Représentation chromosomale sous PosESecL2 d'un caractère C_i du texte Text(n),* **(*b*)** *Représentation chromosomale sous PosESecL2 d'un pixel P_i de l'image Img(n*m).*

Avec : ($h4_i$ $h3_i$ $h2_i$ $h1_i$) représente le codage Unicode hexadécimal du caractère C_i.

b. Reproduction

Contrairement à l'algorithme PosESecL1, l'algorithme PosESecL2 considère le codage des éléments d'une donnée indépendamment. Ainsi, l'application des opérateurs de croisement et de mutation ne nécessite pas la prise en compte des contraintes à vérifier comme dans le cas de PosESecL1, mais il enrichit d'avantage la population d'individus par de nouvelles caractéristiques.

De même que pour le PosESecL1, l'opérateur de croisement utilisé est le OX « Order Cross-over » appliqué avec un taux fixé expérimentalement et compris entre 60% et 100%. Toutefois, l'opérateur de mutation utilisé est une permutation multipoints pour éviter une convergence trop lente vu la grande différence des tailles des chromosomes dans PosESecL1 et PosESecL2 (le deuxième est d'une taille huit ou trois fois plus grande que le premier). Cet opérateur est appliqué avec un taux, aussi fixé expérimentalement et compris entre 0.1% et 5%.

c. Evaluation des individus

Le choix des individus qui survivront d'une génération à une autre s'effectue selon la fonction objective F, donnée ci-dessous, représentative de l'efficacité des solutions générées sur le problème posé.

$$F(I_i) = \sum_{j=1}^{n \times l} \left| G_{ji} - G_{j0} \right| \quad \text{(III.12)}$$

Avec : G_{ji} est la valeur du j^{ème} gène du i^{ème} individu.
I_i est le i^{ème} individu d'une certaine population.
n est la taille de la donnée à chiffrer en terme d'éléments.
l est la taille du codage d'un seul élément de la donnée à chiffrer.

Ainsi, la fonction d'évaluation prendra les deux instanciations suivantes dans le cas de chiffrement de données texte et celui de chiffrement de données images respectivement :

$$F(I_i) = \sum_{j=1}^{n \times 4} \left| G_{ji} - G_{j0} \right| \quad \text{(III.13)}$$

$$F(I_i) = \sum_{j=1}^{n \times 3} \left| G_{ji} - G_{j0} \right| \quad \text{(III.14)}$$

d. Critère d'arrêt

La condition d'arrêt assurant la convergence de l'algorithme PosESecL2 est la même que celle de PosESecL1 puisque le codage des individus dans les deux algorithmes ne diffère que sur le plan contenu de gènes mais les mêmes informations sont présentes dans les deux codages (codage Unicode des caractères du texte à chiffrer ou représentation RVB des pixels de l'image à chiffrer). Autrement dit, c'est la manipulation du codage des éléments de la donnée à chiffrer qui fait la différence entre les deux codages utilisés par les deux algorithmes proposés. Donc, la condition d'arrêt adoptée par ESecL2 dans le cas de chiffrement de données texte et celui de chiffrement de données images, respectivement, est la suivante :

$$0 \le F(I_i) \le F \times 4 \times n \quad / (F)_{16} = (15)_{10} \quad \text{(III.15)}$$

$$0 \le F(I_i) \le 255 \times 3 \times n \quad \text{(III.16)}$$

De même que pour PosESecL1, cette unique condition n'assure pas dans tous les cas la convergence de PosESecL2. C'est pourquoi un nombre maximal de générations sera fixé pour résoudre le problème.

e. Déchiffrement

De même que pour PosESecL1, ce deuxième algorithme utilise le même mécanisme de calcule de clé de chiffrement ; elle change d'un chiffrement à un autre, donc, elle dépend de la donnée à chiffrer et de sa taille. Ainsi, même la taille de la clé de session générée varie d'une donnée à une autre.

f. Réglage des paramètres et résultats

Dans cette section nous présentons les résultats de l'application du deuxième algorithme développé opérant suivant le paramétrage reporté à travers le tableau III.3. À partir des mêmes données originales précédemment utilisées pour tester PosESecL1, nous avons appliqué notre deuxième algorithme PosESecL2 afin d'obtenir les données chiffrées correspondantes. Ces dernières sont celles faisant l'objet des figures III.18.a, III.19.a, III.20.a et III.21.a.

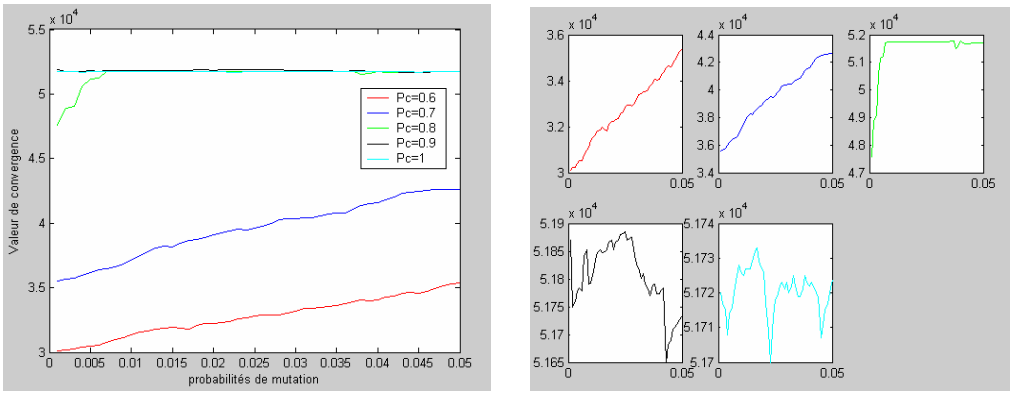


Figure III.14. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

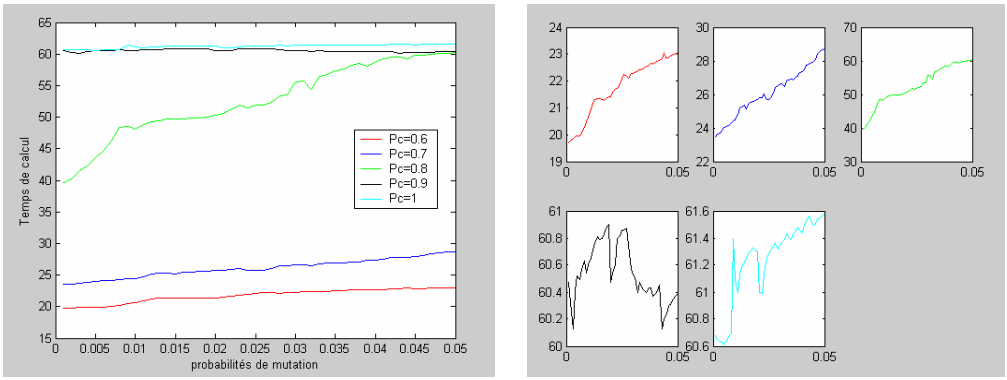


Figure III.15. *Influence des paramètres Pc et Pm sur le temps de calcul.*

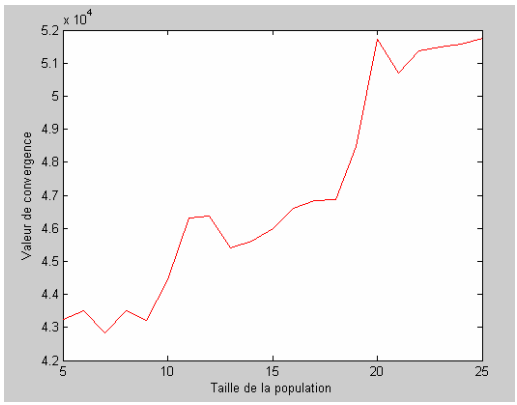


Figure III.16. *Evolution des valeurs de convergence en fonction de la taille de population.*

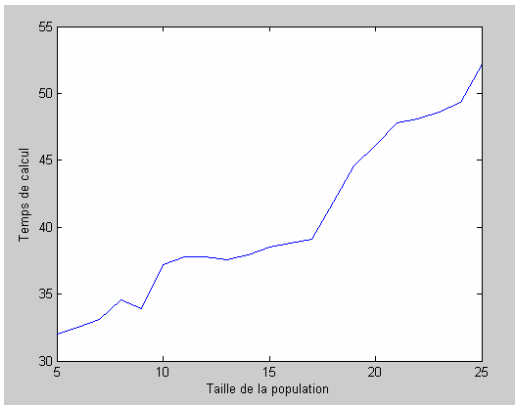


Figure III.17. *Evolution du temps d'exécution en fonction de la taille de population.*

	Valeurs des paramètres de PosESecL2
Taille de la population	20
Nombre de générations	100
P_c	0.8
P_m	0.007

Tableau III.3. *Valeurs adoptées pour les paramètres de PosESecL2.*

L'application de ce deuxième algorithme suivant le paramétrage reporté dans le tableau III.3 sur les mêmes données tests précédentes (Texte 1, Texte 2, Lena et Logo) a permis d'obtenir les résultats résumés à travers le tableau III.4.

D'après ces résultats, il est clair que la taille de la clé augmente proportionnellement avec l'augmentation de la taille de la donnée à chiffrer. Cette dernière influence la taille des chromosomes résultants du codage utilisé. Donc, c'est la taille des chromosomes augmentant en fonction de la taille des données à chiffrer qui conduit à une augmentation de la taille des clés générées et du temps de calcul correspondant.

		Taille donnée (éléments)	Taille clé (bits)	VC	Temps calcul (s)
Données texte	Texte 1	1084	56368	30729	28.74
	Texte 2	1151	59852	35100	31.5
Données images	Lena	131 X 131	823728	51720	46.15
	Logo	420 X 395	9456300	368712	73.06

Tableau III.4. *Résultats obtenus par PosESecL2.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)

´1à1њњ╬‹à+1ﻭﺍî¨à+#cЖéZ0¨£1#§£î¨î ﻍﻭàî+6+¨1ﻭcњ♀њ{à0é6¨à+ﺽ¨éﻍ+‹0++qﻭ61ë̈c¨00c¨î{ﻭﺍî8à‹9¨+њZtt01{î1£#à˚0î♀њ1à+╬0î£01↔
↔¨àâîéë¨+#↔↔¨1´éà0ààà+§8¨£§¨¨0´£è#§é1ë+♀++£11¨{éˀ↔1£ëàאַאַ¨++§+à+¨++9§t´£09£ﻭ╬+ëtˀé#éà0ﻍњ↔6àא18ﻭאַ##ë£¨+0
+¨00‹ˀ+ﻭ+9}1¨˚+8ˀc8££↔à+11++§#‹îﺧﻭZ0ˀ#ﻭﻭëé{§1‹+9é86£009ﻭZﻍ¨cˀ}18+î¨cëîé+§א0‹0+ﻍﻭ§£§îњ1˚0+1c+אﻭàﻭà+Z#q£18£אַﻍﺧ♀ﻭ06#
î1¨ﻍs‹1ﻍ´î#é0#њZ╬§£é↔è+1+ﻍﻭ8£+ëà9{њњњ¨éﻭ¨#ﻭ£àt¨é+++00é˚#î1+î#6à‹0#0##àé11+ˀ‹é#אַЖZ0#+ﻍ££+↔Ж+à1ﺧén£¨¨à#8}ﻭé↔+¨î
╬90+ﻭ+Жë¨î‹§1++↔ëéאַˀ+î£àé#ëZ11א§#אַ♀à#î£+1£é+1ët↔c¨ﺧ+‹ëאַcﻭﺍé#îéˀ#ˀî1#ˀî¨+8#+1+ﻭ£éﻭé╬¨ˀ1£ﻭZZc8њﻍ#é10ﺧà#+↔08¨╬1#+
£é0té£+88+#^ﻭאַ9‹î¨î+אַ¨{˚£§£+´6§6╬╬↔+¨њњњњà{6+##ﻭ↔↔ëˀ+îà+16îњ1¨11£1£0#9+ﻍà§+é£+ë+#11ˀ£ëî‹î+£‹1+´£00£+ﻍ¨+ﻭé#£19
♀♀c§î+c01àî¨8#8ﻭ+╬à§¨{î¨+60¨§6♀0Z8é#1#à{+1£‹à9#1ˀà6+¨1א£˚6à+1î1ë+ˀt9£+îî+80£6é#¨+£¨ZZ1c¨њњ{ëc0♀t0#ﻭtﻭ}+0è↔{ècc8î#
£¨£#ﻭàà£#+1£+њЖњ╬à˚116↔1ˀ+éîZà£c{´+#

(b)

Figure III.18. *La donnée test Texte1 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

Clé de session (Taille$_{\text{Clé}}$ = 6,8808 K octets) :

1	450	496	406	440	458	465	500	437	477	491	298	487	444	468	497	441	471	493
494	454	102	225	455	453	484	473	330	429	432	428	447	469	480	436	488	475	74
446	478	490	443	430	486	483	481	466	499	239	449	489	456	120	464	435	498	451
442	479	474	412	434	467	336	485	5	457	101	213	448	470	431	460	463	438	307
476	445	461	427	459	462	290	495	482	472	433	157	439	426	452	740	1330	545	1331
508	1332	975	1333	564	1334	1335	961	818	1336	886	812	1337	908	674	875	1054	1338	808

725	1339	1340	1341	1342	1343	1344	786	732	1226	1297	228	635	1345	1346	863	696	758	1347
689	1348	1349	1350	710	126	855	548	756	667	522	892	604	1351	1352	602	1353	146	920
770	691	678	729	901	1354	876	1025	1355	4	3	6816	609	993	1356	1357	1229	1358	309
1359	1360	502	1361	706	1362	1363	1364	1365	880	911	644	784	1366	773	359	768	1367	811
299	940	578	198	563	1368	149	957	742	668	687	833	738	1369	623	878	852	636	1370
634	253	1371	873	586	1151	836	620	1372	948	1004	928	1373	799	882	830	270	684	847
964	550	759	1374	870	937	1375	765	905	514	1376	528	739	583	959	337	1377	693	752
612	796	540	1378	552	680	1379	510	946	216	637	630	992	127	978	1309	643	1380	516
1381	559	889	1382	797	567	1383	645	1384	820	949	844	747	1385	866	1386	1387	1160	848
652	384	1388	560	976	985	603	1003	942	501	1389	633	841	974	1093	1390	657	39	912
1391	982	1392	1393	1394	699	750	894	794	1395	1396	932	864	1397	987	716	807	509	676
708	505	1398	838	1399	231	1400	315	339	523	556	326	673	1020	737	599	577	791	751
1329	613	898	1401	1402	419	1001	1403	1404	900	1405	1406	492	1407	536	766	1408	1409	702
787	1410	187	924	1411	845	793	1412	1138	804	1413	1414	1415	546	731	1416	18	1417	66
1418	650	511	662	1288	971	715	817	962	694	1419	968	631	966	537	654	364	520	1104
850	1193	150	981	782	1420	543	915	827	1421	995	1422	843	542	934	1423	1424	775	517
642	709	9	1425	607	772	529	80	994	842	819	712	639	656	2	989	1426	938	610
851	735	666	832	651	541	600	227	241	884	781	895	175	515	417	1237	525	825	734
965	278	890	138	1427	1428	1429	596	1430	553	1293	923	589	730	1431	555	917	1261	663
1432	1433	1000	58	130	1434	790	672	1435	534	1436	1437	1169	44	805	641	800	1438	958
1291	954	1439	1440	1441	1442	835	1443	640	595	1444	933	717	700	888	1445	538	601	616
1446	675	617	701	972	1447	757	909	1448	918	931	1171	704	660	573	998	1449	871	771
1450	763	906	1451	185	1116	557	1452	1453	711	403	916	983	1454	769	1455	221	979	713
1456	846	903	813	839	743	558	910	1295	585	1457	597	1458	1459	1460	967	1461	1462	310
362	313	697	1463	1464	935	719	16	105	856	135	921	952	726	896	1240	1465	705	927
930	1466	397	1467	720	647	980	1468	829	575	950	1469	580	883	664	527	746	323	780
561	614	877	611	828	776	1470	626	1471	570	681	688	569	622	618	879	1472	810	973
572	1473	314	1474	615	683	1475	1476	108	627	1264	665	343	714	382	52	801	1287	152
1477	521	579	28	1478	1479	721	532	1480	1481	549	951	874	46	91	744	592	284	1210
254	1482	206	1483	606	646	945	872	767	320	774	55	348	1484	593	698	535	1485	659
1486	1155	1487	1488	1489	670	1490	990	1491	240	518	815	1492	653	1493	754	899	857	755
798	1494	887	861	1495	143	897	544	996	503	1496	1497	1243	1498	1499	824	1500	1501	1502
533	551	840	977	1503	837	590	1504	988	1505	1506	1507	565	963	788	727	777	1508	1509
809	1510	109	1511	174	1512	869	547	1234	986	904	991	638	1513	999	598	1514	566	722
748	1515	513	802	524	96	1516	1517	1518	919	245	679	507	255	690	1519	733	506	939
922	685	860	1520	853	865	628	504	1521	1522	1523	941	1037	1524	686	1525	355	1526	677
1527	1528	295	914	1189	834	749	661	718	581	1529	764	1530	1531	785	280	823	854	1532
1533	244	859	658	519	648	649	1534	831	984	792	608	554	122	1535	1536	1537	955	723
1538	947	1539	997	1540	568	1541	582	1031	1542	1543	692	1207	588	821	956	671	418	1544
1545	760	1546	943	1547	1548	902	1549	703	1550	1289	695	574	728	745	41	594	970	587
1551	1552	233	1553	1554	655	1555	1556	421	1557	1558	893	562	803	7	814	1559	891	1560
1142	512	1561	629	1562	311	936	73	530	1563	1564	1565	783	1566	1567	1568	137	881	669
624	926	1569	741	415	736	753	953	302	1073	318	1570	779	1571	162	1572	1573	867	266
1574	605	1064	1575	334	90	12	1200	576	1576	1577	969	762	907	1038	619	862	1272	1578
1579	571	822	526	1580	45	49	913	795	1315	761	960	849	944	95	625	1581	1582	201
1583	1005	929	1584	1585	632	531	925	591	1586	789	868	300	682	1587	826	584	885	707
539	1002	806	621	778	724	858	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
1600	1159	1276	1601	1103	1602	1603	1604	1605	1606	1607	128	1608	1609	1610	1611	1612	1613	1614
34	1615	1328	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
1632	1317	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649
1062	1650	1651	1652	1653	1654	1655	1656	1657	1246	1658	1659	1660	1661	1662	1663	1664	1132	1665
1666	1667	349	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683
1684	199	1685	1083	1686	1687	1688	1689	1690	1691	1692	1211	166	1693	1694	1695	1696	1697	291
1247	1153	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1084	1709	1710	1711	140	1712
1713	1714	196	1715	1716	141	1717	1718	1719	1720	1721	1722	1723	1724	1327	287	1725	1726	1009
1727	1728	1729	1730	235	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744
1745	1746	1747	1748	1749	1750	1167	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762
1763	1764	1765	383	57	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779
1780	1144	1781	1185	1782	1783	1784	1294	1785	1786	1324	1787	1156	1788	1789	1790	1791	1792	1793
1195	1281	1794	1795	1796	1797	1798	1799	1800	1014	1801	1802	246	1803	1804	1805	1806	1145	1807
1808	1809	1810	1811	79	1812	1813	1814	1050	1274	286	1815	1816	1817	1818	1819	1820	1821	1822
1823	1824	1825	374	1186	1826	1827	395	1828	1112	1829	1830	112	1831	1832	82	1833	1834	1835
1836	1837	1838	1839	1840	1091	1841	1842	1843	1844	1845	1846	1847	1848	269	1849	1850	17	1275
1851	1177	204	1852	1853	1854	1855	1856	1857	1858	1859	1860	1216	1861	1862	1863	1864	1865	142
1866	1867	1090	1868	1869	1870	56	27	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1039
1881	1882	1883	1884	1075	1885	1886	154	1887	1888	1115	1889	1063	1890	1891	1892	208	1893	1894
1895	1896	1897	1898	1899	1900	1901	1902	1903	1107	1904	1905	1906	1907	1908	1909	1910	1911	1912
1913	1914	1915	1916	125	1917	1918	169	1919	1920	1921	1922	1923	1924	1925	1926	1927	398	1928
1929	1930	1931	1267	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	352	1943	1051	1944
1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	8	1957	1958	1959	1960	1961	1962
293	1963	1964	1965	1966	1967	1968	53	1969	1970	1296	1971	1015	1972	1973	1974	1975	1976	1977
1978	1979	1074	1980	1279	1981	312	1072	1982	1983	1984	1985	1986	264	1987	1988	1989	265	1990
1991	1992	1993	1994	1995	1996	1997	1998	1999	1017	2000	2001	2002	252	2003	2004	2005	342	2006
2007	2008	2009	2010	2011	1130	2012	2013	2014	2015	2016	2017	2018	2019	2020	1041	2021	2022	2023
1058	133	2024	2025	2026	2027	2028	2029	2030	2031	2032	1184	2033	2034	2035	2036	2037	1187	2038
2039	1136	2040	381	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055

1304	305	410	2056	2057	2058	2059	2060	10	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070
2071	2072	2073	2074	35	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088
2089	2090	2091	283	2092	2093	2094	2095	2096	2097	2098	1069	2099	2100	2101	170	2102	2103	2104
2105	2106	2107	2108	2109	2110	1111	2111	297	226	2112	2113	64	2114	2115	267	2116	33	2117
2118	2119	2120	2121	2122	2123	2124	288	2125	2126	2127	2128	2129	2130	2131	176	2132	2133	2134
2135	2136	2137	214	2138	2139	2140	2141	2142	2143	2144	1179	356	2145	2146	2147	2148	2149	2150
2151	2152	2153	2154	2155	2156	2157	2158	2159	99	2160	2161	2162	2163	2164	2165	139	2166	2167
2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	89	2182	2183	2184	153
2185	2186	2187	54	2188	2189	2190	2191	2192	2193	2194	186	2195	1125	2196	2197	2198	2199	2200
1057	2201	2202	2203	2204	2205	261	2206	1158	2207	2208	2209	2210	2211	1087	2212	2213	2214	2215
1273	2216	2217	2218	2219	2220	2221	2222	2223	1230	2224	2225	2226	2227	2228	2229	229	2230	2231
2232	2233	2234	2235	2236	2237	2238	2239	2240	86	2241	2242	1266	2243	2244	103	2245	2246	2247
2248	2249	2250	2251	1235	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	1231	1137	2262	2263
2264	2265	2266	2267	2268	219	2269	2270	376	2271	2272	2273	2274	2275	2276	2277	1113	2278	2279
2280	2281	304	2282	2283	2284	2285	2286	72	2287	2288	2289	2290	2291	394	2292	2293	2294	24
2295	2296	2297	2298	1258	165	2299	1259	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310
107	2311	2312	2313	2314	2315	1110	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327
2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	136
2346	1008	1024	2347	2348	2349	2350	2351	2352	2353	404	2354	1232	2355	2356	2357	2358	2359	2360
2361	2362	1102	2363	2364	2365	2366	371	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377
2378	2379	2380	2381	2382	2383	1217	1046	2384	2385	2386	2387	2388	2389	2390	2391	2392	123	2393
2394	2395	379	2396	2397	2398	2399	2400	1203	2401	274	1223	1071	2402	388	2403	2404	2405	1081
2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	1019	2416	2417	2418	2419	2420	2421	2422	2423
2424	2425	2426	2427	2428	2429	2430	2431	2432	40	2433	2434	2435	2436	2437	2438	2439	2440	1162
2441	2442	2443	1109	2444	306	2445	1325	2446	2447	2448	2449	2450	2451	2452	2453	2454	134	2455
2456	2457	76	2458	2459	2460	43	2461	2462	2463	2464	2465	129	2466	2467	2468	2469	2470	2471
399	191	2472	2473	2474	2475	2476	2477	405	2478	2479	2480	2481	2482	2483	2484	2485	2486	2487
2488	294	2489	2490	2491	2492	2493	2494	2495	2496	177	2497	2498	276	247	2499	2500	256	2501
2502	2503	2504	2505	2506	2507	389	2508	2509	2510	2511	2512	2513	2514	2515	2516	1012	346	2517
2518	350	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528	2529	2530	2531	2532	2533	2534	2535
2536	2537	2538	2539	2540	2541	2542	2543	1222	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553
1206	2554	2555	2556	369	2557	2558	2559	292	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569
2570	2571	2572	2573	2574	1028	2575	1205	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586
104	2587	2588	2589	2590	2591	2592	2593	2594	328	2595	2596	1120	2597	2598	2599	2600	2601	2602
2603	2604	2605	2606	2607	316	236	2608	2609	1286	2610	2611	2612	2613	2614	2615	2616	2617	2618
411	2619	2620	2621	2622	2623	2624	2625	2626	1016	2627	365	2628	1214	2629	2630	2631	2632	2633
2634	2635	2636	2637	2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652
2653	2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	19	2664	2665	2666	2667	2668	2669	2670
2671	1176	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687	2688
2689	366	2690	2691	2692	2693	2694	2695	2696	2697	2698	271	2699	172	2700	1114	2701	2702	2703
2704	2705	193	2706	2707	2708	2709	205	2710	2711	1122	2712	2713	2714	2715	2716	1248	2717	2718
1150	2719	2720	2721	2722	2723	2724	189	373	2725	2726	2727	2728	2729	1183	2730	2731	2732	2733
2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	1182	2748	2749	2750	1199
2751	2752	1036	2753	420	2754	2755	145	1148	2756	2757	2758	2759	2760	1194	2761	2762	2763	1106
2764	2765	2766	115	2767	2768	368	2769	2770	163	2771	2772	2773	2774	2775	390	2776	2777	2778
2779	111	113	2780	248	2781	2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794
2795	2796	2797	2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	173
2813	2814	2815	168	2816	2817	2818	2819	1061	2820	2821	1249	2822	1094	2823	159	2824	2825	2826
2827	2828	2829	2830	2831	2832	2833	2834	2835	1146	2836	1022	2837	2838	2839	1027	2840	2841	2842
48	2843	2844	2845	2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	360	2857	2858	2859
2860	2861	2862	2863	2864	1209	22	178	2865	2866	1181	2867	2868	2869	2870	1085	32	14	2871
1174	2872	2873	2874	2875	2876	1307	1134	1096	2877	1218	2878	2879	2880	2881	2882	2883	2884	2885
2886	329	2887	2888	144	2889	2890	2891	2892	2893	2894	2895	2896	1066	2897	2898	2899	2900	2901
2902	2903	2904	1244	409	2905	400	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915	2916	2917
2918	2919	1170	2920	2921	2922	132	42	2923	2924	2925	2926	2927	2928	2929	2930	77	2931	2932
211	1143	1121	242	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943	1192	2944	2945	422
2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	87	2961	2962	2963
2964	2965	2966	2967	1040	2968	2969	2970	2971	2972	2973	2974	2975	2976	2977	322	2978	2979	2980
2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999
3000	3001	1013	3002	3003	37	3004	3005	3006	3007	3008	1300	3009	3010	1250	3011	3012	3013	257
3014	3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029	1166	3030	3031
3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	1262	3043	3044	1313	1149	372	1078	3045
3046	3047	3048	3049	3050	3051	3052	1215	3053	3054	3055	3056	3057	3058	3059	3060	3061	1265	3062
3063	3064	181	3065	3066	3067	268	3068	3069	3070	3071	182	3072	3073	3074	3075	3076	3077	3078
401	3079	3080	3081	3082	161	3083	3084	3085	3086	3087	3088	3089	3090	3091	3092	3093	3094	1178
3095	1268	3096	3097	3098	3099	3100	3101	3102	3103	3104	3105	3106	3107	3108	3109	1055	3110	3111
3112	3113	3114	1256	3115	3116	3117	3118	3119	3120	3121	3122	3123	3124	1204	3125	3126	3127	3128
3129	3130	3131	3132	3133	3134	3135	3136	3137	3138	50	3139	3140	1131	3141	3142	3143	3144	3145
3146	3147	3148	3149	151	3150	3151	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	1320
3163	317	1070	327	3164	3165	1242	3166	3167	1299	402	3168	3169	367	3170	3171	3172	3173	3174
3175	3176	3177	3178	131	3179	94	3180	3181	3182	1154	3183	1047	3184	3185	3186	1168	3187	3188
363	3189	3190	3191	3192	3193	3194	258	3195	3196	121	3197	3198	93	3199	319	3200	1099	3201
3202	3203	202	3204	3205	354	416	3206	3207	3208	3209	3210	3211	1316	3212	3213	3214	3215	3216
3217	3218	3219	3220	3221	303	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231	3232	243	3233
1269	3234	1233	3235	3236	3237	85	3238	3239	3240	3241	3242	3243	1319	1213	3244	3245	3246	3247
3248	218	1225	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	1048	3259	3260	119	3261	3262

3263 3264 3265 3266 3267 3268 3269 238 3270 3271 3272 3273 3274 3275 3276 223 3277 1032 3278
3279 1101 3280 3281 3282 3283 1076 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 1260 3294
3295 3296 3297 3298 3299 3300 279 3301 3302 3303 3304 3305 118 3306 3307 3308 1129 3309 3310
3311 97 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328
3329 1135 3330 3331 3332 3333 184 3334 370 3335 3336 224 3337 3338 3339 1191 3340 3341 3342
3343 110 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 1117
1254 3360 3361 3362 3363 3364 3365 188 3366 3367 3368 1157 1147 3369 155 3370 3371 3372 1282
21 3373 3374 3375 3376 1241 3377 3378 425 3379 3380 3381 3382 3383 1190 3384 3385 1033 3386
3387 3388 3389 3390 3391 413 3392 3393 1065 3394 3395 285 1228 3396 3397 3398 3399 3400 3401
3402 3403 3404 194 3405 3406 3407 321 230 1098 3408 3409 3410 3411 3412 3413 197 3414 3415
3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 1251 3428 3429 3430 3431 3432 3433
1152 3434 3435 353 1030 3436 1292 3437 3438 215 1180 3439 3440 3441 3442 3443 3444 3445 3446
3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 164 3457 3458 3459 361 3460 3461 3462 3463
3464 3465 3466 3467 25 3468 3469 331 3470 3471 3472 3473 3474 3475 1270 3476 195 3477 3478
3479 3480 3481 3482 3483 3484 3485 3486 20 340 3487 3488 3489 3490 84 3491 3492 3493 3494
3495 3496 1290 209 1108 1006 3497 3498 3499 3500 3501 190 3502 3503 3504 3505 3506 1018 3507
3508 3509 3510 1224 1053 1124 207 3511 212 3512 3513 3514 3515 1285 114 3516 3517 3518 3519
324 3520 3521 3522 3523 3524 3525 3526 1221 3527 3528 158 3529 3530 3531 3532 3533 1236 3534
335 3535 3536 1172 3537 3538 3539 3540 3541 3542 1035 3543 1318 1119 1220 1011 3544 3545 3546
3547 3548 3549 3550 3551 3552 3553 3554 424 3555 3556 387 3557 3558 3559 3560 3561 3562 3563
3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 156 3574 3575 3576 3577 3578 3579 3580 3581
3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 1118 3592 1303 3593 3594 3595 3596 31 263
3597 3598 3599 3600 3601 3602 3603 69 3604 3605 3606 3607 3608 3609 1133 3610 3611 3612 3613
1196 3614 3615 3616 1219 1173 3617 396 1188 3618 3619 3620 3621 282 3622 3623 3624 3625 3626
3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 1239 1198 3637 3638 3639 408 3640 3641 3642
1049 1089 3643 3644 1306 3645 3646 3647 3648 3649 296 3650 3651 3652 1271 3653 3654 3655 160
3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674
3675 3676 3677 3678 3679 3680 1305 3681 3682 3683 1284 3684 3685 3686 3687 3688 3689 3690 3691
3692 3693 3694 3695 3696 3697 3698 15 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 217
3709 3710 1127 3711 1298 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725
1140 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 1067 3738 3739 3740 3741 180
3742 3743 3744 1023 3745 3746 3747 3748 3749 3750 3751 3752 1161 3753 332 3754 3755 3756 3757
1252 3758 3759 3760 3761 3762 171 3763 3764 3765 3766 3767 3768 100 3769 3770 3771 3772 3773
3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 192 3787 3788 3789 3790 3791
3792 3793 3794 3795 3796 3797 262 1082 333 3798 3799 124 3800 3801 3802 1312 1052 3803 3804
3805 3806 391 3807 3808 1278 3809 1042 83 3810 347 3811 3812 3813 70 3814 3815 3816 3817
3818 3819 3820 3821 3822 3823 3824 3825 1126 3826 3827 3828 3829 3830 3831 3832 3833 3834 98
3835 51 81 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 1255 3846 3847 67 3848 3849
3850 3851 3852 3853 3854 3855 1208 3856 3857 30 3858 3859 3860 3861 65 301 47 1021 3862
3863 3864 3865 3866 3867 3868 29 3869 3870 3871 3872 3873 3874 3875 1253 3876 3877 3878 3879
1321 3880 3881 3882 3883 63 3884 71 3885 3886 3887 3888 275 1175 3889 3890 3891 3892 3893
3894 3895 1045 3896 3897 3898 3899 423 260 3900 3901 3902 308 3903 3904 3905 3906 26 59
3907 3908 3909 1322 3910 3911 3912 3913 3914 3915 1197 1308 3916 3917 3918 3919 3920 3921 3922
3923 3924 1086 3925 3926 3927 3928 1263 68 1128 272 3929 3930 3931 3932 3933 167 3934 3935
3936 3937 3938 3939 3940 3941 3942 147 3943 1034 3944 3945 3946 3947 3948 3949 3950 3951 3952
3953 3954 3955 3956 3957 3958 3959 3960 357 3961 3962 3963 3964 3965 3966 3967 1165 1277 3968
38 3969 3970 232 3971 3972 3973 3974 3975 277 3976 3977 3978 3979 3980 3981 3982 3983 3984
3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003
13 4004 4005 4006 1202 4007 4008 4009 4010 4011 4012 4013 4014 4015 1314 4016 4017 4018 1326
4019 4020 4021 1141 4022 250 4023 4024 4025 4026 4027 4028 4029 4030 1201 4031 1227 4032 386
4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 1095 179 4043 4044 4045 1080 4046 4047 4048
4049 75 1029 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 237 4060 4061 4062 4063 4064
4065 4066 203 4067 4068 1301 414 249 4069 4070 380 4071 4072 1283 4073 4074 4075 78 4076
378 4077 61 4078 4079 4080 4081 4082 4083 4084 88 4085 4086 325 4087 1044 4088 4089 4090
200 4091 377 4092 4093 1164 4094 4095 4096 4097 222 4098 4099 4100 4101 4102 1059 4103 4104
4105 4106 4107 1026 4108 4109 1100 4110 358 4111 4112 1238 4113 1139 4114 4115 4116 4117 4118
4119 4120 1060 1302 4121 4122 4123 4124 4125 4126 4127 344 148 4128 4129 4130 4131 1092 4132
4133 4134 4135 4136 4137 4138 4139 4140 4141 1310 273 4142 4143 4144 4145 4146 4147 4148 4149
4150 4151 1163 116 4152 4153 4154 4155 259 251 4156 4157 4158 4159 4160 4161 4162 407 4163
4164 375 60 4165 4166 4167 1043 4168 4169 4170 4171 36 4172 4173 4174 1105 385 4175 4176
4177 4178 4179 4180 4181 4182 4183 220 4184 1077 4185 4186 4187 1007 4188 4189 4190 106 4191
11 4192 4193 1280 4194 4195 4196 4197 4198 4199 4200 289 92 4201 4202 4203 4204 4205 4206
1311 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223 4224
4225 4226 1097 1123 4227 4228 4229 4230 62 4231 351 4232 4233 4234 4235 4236 4237 4238 4239
4240 1257 4241 4242 4243 4244 4245 4246 4247 4248 1056 4249 183 4250 1245 4251 4252 4253 4254
4255 4256 4257 338 4258 4259 4260 4261 117 341 4262 4263 4264 1079 4265 4266 4267 4268 4269
4270 4271 4272 4273 4274 4275 4276 4277 4278 4279 4280 4281 4282 4283 4284 4285 4286 4287 4288
4289 4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303 4304 4305 4306 4307
4308 345 4309 4310 234 4311 4312 281 4313 4314 1212 1088 4315 4316 1323 4317 4318 4319 392
210 4320 4321 4322 4323 4324 393 4325 4326 4327 4328 4329 4330 1010 23 4331 4332 4333 4334
1068 4335 4336

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملا أساسيا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردى بعدما كان الأمر مقصوراً فى الماضى على المراسلات الرسمية أو السرية بطبيعة الحال.

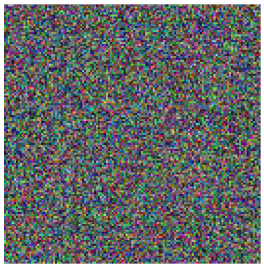
(a)

8ʊ8▒8}¢j^∂ز╔╔ש╔╔j8♂♂®±ĀʊĀ¦88¦(▒}÷╔♂}j¢∂àא`j0ıoز®שʊʊâ♠â&∑▒÷(8♠┤÷┤j∂
(&g8}¦`¦8j(ë÷Ж÷Жjv`♂8±┤¢Æ8^){&ıoıo88j&שÓÖ¦ʊ¢א3ز÷`!ë}ıo¦(ز∑┤¦÷÷3j8®Ā&#j8∂ë8â}╔÷8#8ʊʊʊıoâ:}jj8(&שıoıo8┤j╔8®ĀĀ3¢±
88ë(¦àjא∂}:8¢¢jj¦ç83÷8`ز&∂0&ز®&j÷8▒â┤}Ö&}¢}!j┬jâ3¦¦#`8┤jÖ&Ā8¢â¦±¢j3♠8}ʊ♠Ā3¢`¦ë3±▒▒▒ت&÷j¢ے 8!8┤
∂Æ3÷}jj}¦¢¦&¦8♠♠♠j3ت¦ز¦ıo¦¢8∂8jت÷ʊâ÷^â¢{ز:8÷¢ز∂8¢±¢ıoıoıo╔╔8!¢¦8¢â♂∑∂∑+_¦&زÆ ے {÷`&
8jj(&▒Æ╔╔╔â8┤א8Ā&א÷^±3∑8}{¢ت Ж®¢8¦&♠&¦ت▒0א8±ש∑∑Ā&(&}3¦â3¦®8&â&(÷àj¦(¦ے
ëâ_8ززjj&88&88ت&àj(¦±╔+!}÷ززت÷} Ж&&}ʊجë:8÷╔Ö8!¢ ש}8±8Öے ت╔8{8&8*!ت: { ʊز±&
ے®3ëâ3زشززjʊא0}∂¢j÷&gıoıo¢3┤♠}3♠j8÷âj88زg0¢¢∂ıo∂¦â?ت÷┤▒8÷&j♠+Æ8¢^±ت(`jıo^▒א ▒ز (àت Ā3: ıo&8±ש¢`3ژ88┤ ے ıo3Ж8ے
8:8}±}8زj`88#┤¦&j¦88: Ö3ز8¢¦}∑8&▒زj&8ЖК (¢&ز®
(j88ژ¦ت÷ف}ë÷&8÷&♂¢88ززjh&Æ&÷8¦3&¦88¢8!{¢)¢&8÷ز±±}^!âj«!â¦à8&&!8ز
♠┤&▒8{j8j±¦ש¦j8à÷:8®8ز±∑3j&3ששש8±┤}j}j÷÷+i#∂j¦(3¦ز3┤àĀЖ}3! ש±÷â83&â0¦â3`0&ت▒ے ز ë83شàא∂ıoא∂∂3&8┤Æے
Æ8^â♠^3}&â¦┤┤ز^Öj¦â888ے
â}÷&}&88÷ز¢ز8ز*:zj3¢}j¢زא8&jزıoj}8¢¢ز♠♠8â&┤+3¢8¢ت:âjژ8{+┤`âà÷#8¦`}ز÷âЖ&j:`jà÷83∑Æj÷÷j`â÷}÷*Öj30₪8÷8زà*8¢┤÷ز±÷
3÷¢8«8±┤&j8¦8â÷8╔8¢j88±8ز♠8∂ز3j&♠¦¢ë╔┤÷زâ388àzj¦¦0ت®®÷jâ┤}÷«ɜ3â8}₪8▒&8â3*¦₪₪¢â¦¦â÷¦88¦*8┤ʊ±¦à÷^¦שЖ88∑8ë¦jâĀĀĀ&8
*08#«8ЖÆ8j#¢ʊʊâ3÷!زÖ&┤{{*ÖÖÖשששש∂88ژتj┤÷¦àà: ╔▒▒ש8ز÷¦0Æ♠}j&}∂}jزʊא₪ЖК╔Жâ¦÷ت₪ı┤¦¢ששز}3¢8ت₪±jâ∑!÷38j}¦÷♠

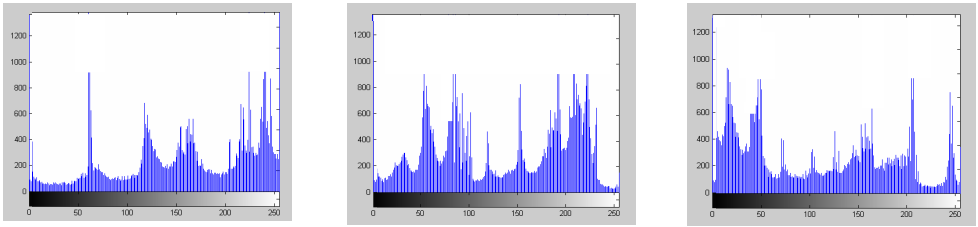
(b)

Figure III.19. *La donnée test Texte2 :* ***(a)*** *Donnée originale,* ***(b)*** *Donnée chiffrée.*

(a)

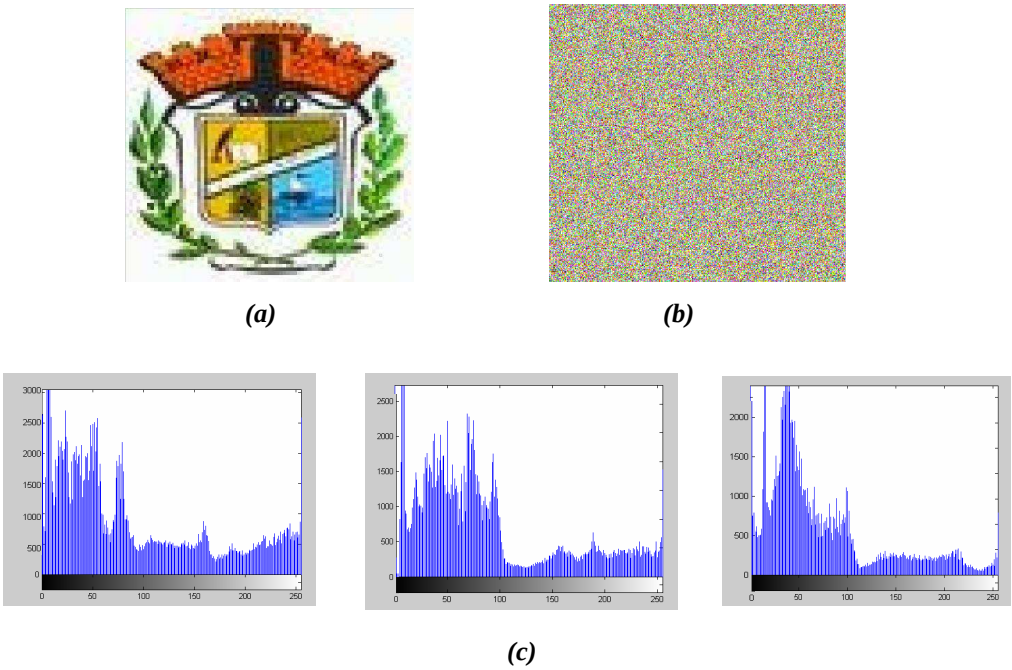


(b)



(c)

Figure III.20. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*



(a) ***(b)***

(c)

Figure III.21. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

III.4.2. Chiffrement à base d'occurrences

Nous avons vu dans la section précédente que le codage défini des individus influence grandement l'efficacité de l'algorithme. Bien qu'il soit étroitement dépendant du problème à résoudre, sa définition permet de cerner l'espace des solutions possibles. Ce codage doit, de plus, être aussi compact que possible pour permettre une évolution rapide. Ainsi, dans la présente section nous allons présenter un nouveau codage totalement différent de ceux vu dans les sections précédentes qui ont été bâti autour de la notion de positions de la représentation des éléments constituant la donnée à chiffrer.

Pour appliquer l'approche évolutionnaire à notre problème, nous présenterons les choix utilisés en terme de : représentation chromosomale des solutions du problème, méthode de création de la population initiale des solutions, fonction d'évaluation, opérateurs génétiques et procédure de sélection. Pour mieux approcher le problème, nous présentons une formalisation de ce dernier.

III.4.2.1. Formalisation du problème

Soit D une donnée à chiffrer (texte ou image) qui est une suite de n éléments e_i, et que l'on peut formaliser comme suit :

$$D = \{e_i\}, i \in [1,n] \qquad \text{(III.17)}$$

Dans le cas où D soit une donnée texte, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{C(e_i)\}, i \in [1,n] \qquad \text{(III.18)}$$

où : $C(e_i)$ représente le codage Unicode du caractère e_i.

Dans le cas où *D* soit une donnée image, la formalisation (III.17) peut être instanciée comme suit :

$$D = \{p_i\}, i \in [1,n] \quad \text{(III.19)}$$

Cette représentation (structure) distinguant les différents pixels formant cette image, facilite la conversion en une autre structure en choisissant le RGB (Red Green Blue) comme espace de représentation d'images. Il suffit de remplacer les n pixels par leurs représentations correspondantes en RGB. Nous obtenons, ainsi, la formalisation suivante :

$$D = \{R_i G_i B_i\}, i \in [1,n] \quad \text{(III.20)}$$

Ce mode de représentation (structure) permet de distinguer les différents codages d'éléments de la donnée à chiffrer (les codages des caractères d'une donnée texte, ou les différentes composantes codant les pixels d'une donnée image), et par conséquent de déterminer le nombre d'occurrences de chacun d'eux. C'est ce mécanisme qui sera d'ailleurs exploité pour bâtir notre suivant algorithme de chiffrement proposé.

III.4.2.2. Description d'OEEA (*Occurrences based Evolutionary Encryption Algorithm*)

Comme l'opération de chiffrement consiste à perturber la donnée originale de telle manière à avoir le maximum de désordre dans la donnée chiffrée, nous avons choisi, cette fois-ci, de jouer sur les nombres d'occurrences des éléments (caractères d'un texte ou valeurs des composantes R, G et B codants les pixels d'image), pour avoir la plus grande différence entre les nombres d'occurrences des valeurs similaires d'éléments (nombres d'occurrences des caractères d'un texte original et leurs nombres d'occurrences dans le texte chiffré correspondant ; ou nombres d'occurrences des valeurs des composantes R, G et B codant les pixels d'une image originale et leur nombre d'occurrence dans l'image chiffrée). Ce choix est dû au fait que cette unique donnée (nombres d'occurrences) ne représente pas une information pertinente pour les cryptanalystes.

Dans ce qui suit, nous décrierons les différentes étapes de notre nouvel algorithme de chiffrement proposé.

a. Codage

Les structures (III.18) et (III.20) vues précédemment représentent les données de base à partir desquelles nous sommes arrivés à définir notre codage d'individus en calculant le nombre d'occurrences des valeurs possibles d'éléments dans une certaine donnée. Ainsi, le codage proposé sera le suivant :

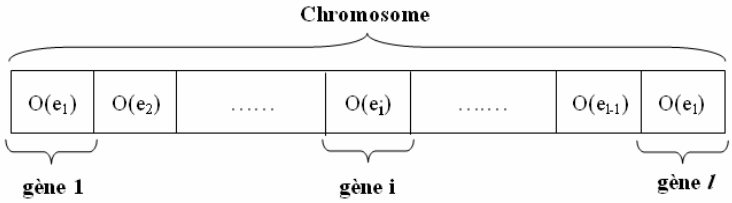


Figure III.22. *Codage des individus sous OEEA.*

Où : O (e_i) est le nombre d'occurrences de l'élément e_i dans la donnée,
l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Remarque :

1) $\sum_{i=1}^{l} O_i = n$ Où n ∈ N représente la taille de la donnée en termes d'éléments (caractères ou pixels).

2) Les éléments qui ne figurent pas dans la donnée auront une occurrence nulle.

Dans le cas de manipulation d'une donnée texte, l'opération de codage consiste à transformer le message en code particulier en calculant le nombre d'occurrence dans le message des 1393 caractères admissibles et l'attribuer à la case qui correspond à son rang dans une table, désignée par TCAR, regroupant les 1393 affichables en Unicode.

Le codage adopté dans ce cas, sera une instanciation du codage général présenté par la figure III.22. C'est celui illustré par le synoptique de la figure III.23.

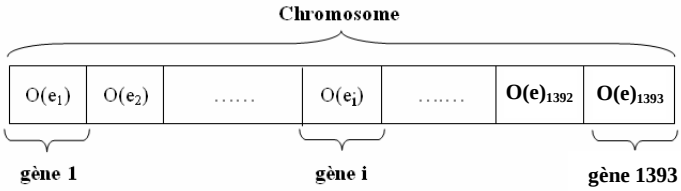


Figure III.23. *Codage des données texte sous OEEA.*

Où $i = \overline{1-1393}$ est le rang des caractères dans la table TCAR et $O(e_i)$ le nombre d'occurrences dans le massage du caractère ayant comme rang i.

Dans le cas de manipulation d'une donnée image et vu que les composantes R_i, G_i et B_i / $i = \overline{1,n}$ prennent leurs valeurs dans l'intervalle [0, 255], notre codage consiste à compter parmi les valeurs de chacune des composantes R, G et B, les nombres d'occurrences relatifs aux valeurs de l'intervalle [0, 255].

Il s'agit de compter à partir des valeurs des n éléments de R, les nombres d'occurrences des valeurs de l'intervalle [0, 255], et de même pour les valeurs des éléments de G et de B. La structure résultante dans ce cas, représente le codage adopté par notre algorithme développé. C'est celle résumée par la représentation chromosomale présentée à travers le synoptique de la figure III.24.

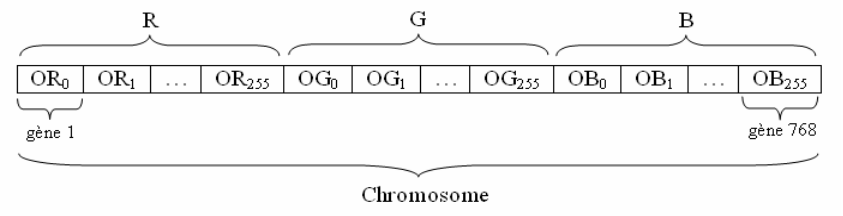


Figure III.24. *Codage des données images sous OEEA.*

Où : OR_i est le nombre d'occurrence des valeurs de la matrice composante R qui égale à *i*,
OG_i est le nombre d'occurrence des valeurs de la matrice composante G qui égale à *i*,
OB_i est le nombre d'occurrence des valeurs de la matrice composante B qui égale à *i*.

Si le codage binaire semble tout à fait adapté pour certains problèmes d'optimisation, cette représentation semble peu appropriée dans notre cas car elle est non intuitive. La représentation qui nous semble la plus naturelle est le codage entier, puisque toutes les opérations de l'algorithme vont manipuler des nombres d'occurrences qui sont des nombres entiers.
Dans ce qui suit, on désigne par individu la donnée à chiffrer ou toute autre donnée, et par chromosome tout individu ayant subi une telle opération de codage.

b. Création de la population initiale

Les m individus (I_1, I_2,…, I_m) formant notre population initiale sont obtenus par application de m perturbations aléatoires sur le chromosome initial I_0 représentant le codage de la donnée originale.

c. Reproduction

Les opérateurs génétiques servent à diversifier la population gérée par l'AE afin d'explorer le plus efficacement possible l'espace de recherche. Nous avons vu dans le chapitre 2 que les AEs disposent généralement de deux opérateurs : l'opérateur de croisement et l'opérateur de mutation.

- **Le croisement :**

Le croisement de solutions peut être toute opération permettant d'obtenir une nouvelle solution à partir de plusieurs individus existants.

Dans le but de compliquer la tâche des cryptanalystes tout en maintenant raisonnable le temps de calcul global de l'algorithme et en évitant toute convergence prématurée, nous avons utilisé un croisement à deux points. C'est le OX « Order Cross-over » proposé par Davis [Davi, 1985]. Les deux points de croisement sont choisis aléatoirement et l'opérateur est appliqué avec un taux qui varie entre 60% et 100% [Gren, 1986].

- **La mutation :**

Pour notre problème, nous avons opté pour une simple mutation, celle qui consiste à permuter aléatoirement deux gènes d'un chromosome. Le meilleur taux varie entre 0.1% et 5% [Gren, 1986].

Les individus résultants de ces deux opérations (croisement et mutation) seront rajoutés à la population des parents pour les acheminer vers l'étape suivante qui est celle d'évaluation.

d. Evaluation des individus

La fonction d'évaluation que nous avons définie pour associer des valeurs d'adaptation à chaque individu I_i d'une population Pop, est la suivante :

$$F(I_i) = \sum_{j=1}^{l} \left| O(e_j)_i - O(e_j)_0 \right| \qquad \text{(III.21)}$$

Avec : $I_i = [O(e_1), O(e_2),\ldots, O(e_l)]$, $Pop = \{I_1, I_2,\ldots, I_m\}$ où m est un paramètre de réglage et $O(e_j)_i$ représente le nombre d'occurrences du $j^{ème}$ élément dans le $i^{ème}$ individu.

Dans le cas de manipulation d'une donnée texte, F peut être instancié comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O(e_j)_i - O(e_j)_0 \right| \quad \text{(III.22)}$$

Ou tout simplement :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.23)}$$

Où O_{j_i} est le j^{ème} gène du i^{ème} individu.

Dans le cas de manipulation d'une donnée image, F sera instancié comme suit :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{j_i} - R_{j_0} \right| + \sum_{j=1}^{256} \left| G_{j_i} - G_{j_0} \right| + \sum_{j=1}^{256} \left| B_{j_i} - B_{j_0} \right| \quad \text{(III.24)}$$

Cette fonction est équivalente à celle donnée ci-dessous.

$$F(I_i) = \sum_{j=1}^{768} \left| O_{j_i} - O_{j_0} \right| \quad \text{(III.25)}$$

e. Sélection des individus les plus adaptés

La stratégie de sélection de l'algorithme développé va décider de la survie d'une donnée dans la population. Nous avons utilisé la méthode de la sélection par roulette.

f. Critère d'arrêt

Les étapes de reproduction, d'évaluation et de sélection sont répétées jusqu'à ce que l'optimum recherché soit trouvé. Ce dernier représente l'individu ayant la plus grande valeur de convergence durant tout le processus évolutionnaire. Donc, c'est la donnée la plus différente de la donnée originale.

Dans le cas d'une donnée texte, la condition d'arrêt assurant la convergence de notre algorithme évolutionnaire est exprimée à l'aide de la fonction F comme suit :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.26)}$$

Preuve :

$$F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} (O_{j_i} + O_{j_0}) \quad \text{(III.27)}$$

$$\leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.28)}$$

Et comme :

$$\forall I_i, I_k \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_k} \quad \text{(III.29)}$$

Et :

$$\forall I_i \in Pop : \sum_{j=1}^{1393} O_{j_i} = \sum_{j=1}^{1393} O_{j_0} \quad \text{(III.30)}$$

Donc :

$$(III.27) \Rightarrow \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq \sum_{j=1}^{1393} O_{j_i} + \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.31)}$$

$$\leq 2 * \sum_{j=1}^{1393} O_{j_i} \quad \text{(III.32)}$$

Dans le cas d'une donnée image, nous avons :

$$F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right|$$

Et comme : $((0 \leq R_{ji} \leq n) \,\&\, (0 \leq R_{j0} \leq n)) \Rightarrow (0 \leq \left| R_{ji} - R_{j0} \right| \leq n)$ (III.33)

$((0 \leq V_{ji} \leq n) \,\&\, (0 \leq V_{j0} \leq n)) \Rightarrow (0 \leq \left| V_{ji} - V_{j0} \right| \leq n)$ (III.34)

$((0 \leq B_{ji} \leq n) \,\&\, (0 \leq B_{j0} \leq n)) \Rightarrow (0 \leq \left| B_{ji} - B_{j0} \right| \leq n)$ (III.35)

Alors, d'après (III.33), (III.34) et (III.35) nous aurons :

$$0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq (256 \times n) + (256 \times n) + (256 \times n) \quad \text{(III.36)}$$

$$\text{(III.36)} \Rightarrow 0 \leq \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n \quad \text{(III.37)}$$

$$\Leftrightarrow 0 \leq F(I_i) \leq 768 \times n \quad \text{(III.38)}$$

Donc, les inéquations (III.26) et (III.38) représentent les conditions d'arrêt mettant fin à notre processus crypto-évolutionnaire dans le cas de manipulation de données texte ou images, respectivement. Nous constatons que $F(I_i)$ est une fonction bornée.

Après quelques tests d'exécution, nous avons constaté que le fait de prendre cette condition uniquement comme condition d'arrêt peut poser le problème de boucle infinie du moment où la valeur de convergence maximale devient inchangée d'une itération à une autre tout en vérifiant la condition d'arrêt ((III.26) ou (III.38)). Pour remédier à ce problème, nous avons pensé à fixer expérimentalement un nombre maximum d'itérations (de générations) à ne pas dépasser. Ainsi et suite à plusieurs exécutions, nous avons arrivé à déterminer ce nombre :

MaxGen = 50

La condition d'arrêt finalement adoptée englobe les deux conditions suivantes dans le cas de chiffrement de texte :

1) $F(I_i) = \sum_{j=1}^{1393} \left| O_{j_i} - O_{j_0} \right| \leq 2 * \sum_{j=1}^{1393} O_{j_i}$

2) MaxGen = 50

Toutefois, elle englobe les deux conditions suivantes dans le cas de chiffrement d'images :

1) $F(I_i) = \sum_{j=1}^{256} \left| R_{ji} - R_{j0} \right| + \sum_{j=1}^{256} \left| V_{ji} - V_{j0} \right| + \sum_{j=1}^{256} \left| B_{ji} - B_{j0} \right| \leq 768 \times n$

2) MaxGen = 50

g. Déchiffrement

Dans cette phase représentant l'opération inverse du chiffrement, une clé de session est générée suivant le même mécanisme utilisé pour nos deux précédents algorithmes. Toutefois, elle est de taille fixe vu que toutes les données texte ont une représentation chromosomale fixe englobant 1393 gènes et toutes les données images ont une représentation chromosomale fixe regroupant 768 gènes.

h. Réglage des paramètres et résultats

Afin de construire un jugement objectif à propos de l'algorithme proposé et de pouvoir noter le maximum de remarques et de comprendre les détails, une large panoplie d'images tests de différentes dimensions a été utilisée. Nous présentons ci-dessous et à titre d'exemple les résultats obtenus pour les mêmes données test précédemment utilisées dans la présentation de nos deux algorithmes développés et précédemment présentés PosESecL1 et PosESecL2 : "Texte 1" de taille 1084 caractères, "Texte 2" de taille 1151 caractères, "Lena" de taille 131X131 pixels et "Logo" de taille 420X395 pixels.

- **Réglage de paramètres**

L'algorithme décrit ci-dessus possède différents paramètres, dont les valeurs influent fortement sur sa qualité. Cette section présente une étude empirique du réglage de ces paramètres afin d'obtenir de bonnes performances, ainsi que le comportement général de l'algorithme qui en résulte.

Les figures III.25 et III.26 récapitulent les résultats moyens de chiffrement en termes de valeurs de convergence et de temps d'exécution suivant les différentes valeurs de probabilités de croisement et de mutation.

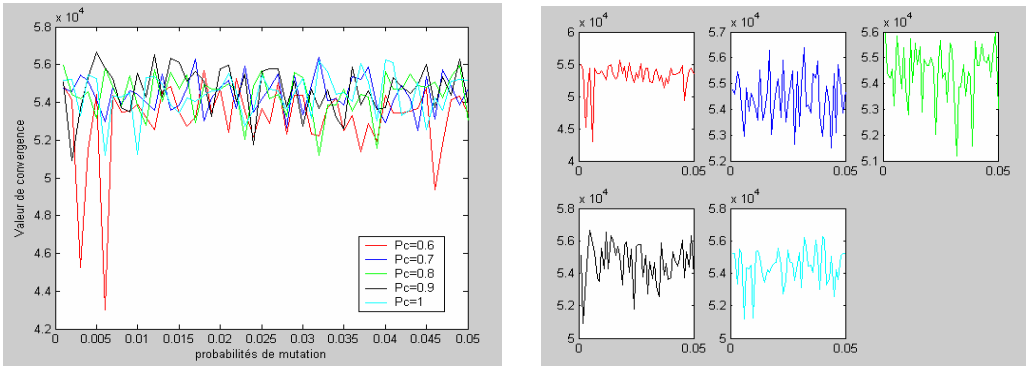


Figure III.25. *Influence des paramètres Pc et Pm sur la valeur de convergence.*

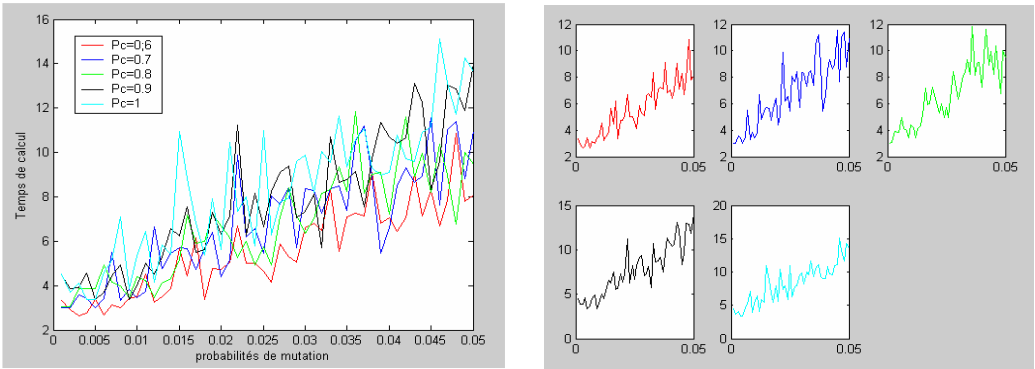


Figure III.26. *Influence des paramètres Pc et Pm sur le temps de calcul.*

Pour choisir les valeurs des paramètres de la probabilité de croisement et celui de la probabilité de mutation, le niveau de confusion n'est pas la seule exigence à satisfaire ; le temps de calcul est aussi important. Ainsi, les valeurs qui nous semblent les plus adaptées sont : *Pc=0.9* et *Pm=0.005* assurant un haut niveau de confusion (VC=56674) en un temps de calcul très raisonnable (T=3.37s).

Après avoir fixé les paramètres de la probabilité de croisement et de mutation, la taille de la population est un autre paramètre à régler expérimentalement aussi. Les valeurs testées sont résumées à travers les figures III.27 et III.28 en indiquant pour chacune d'elles les résultats de chiffrement obtenus en terme de degré de confusion (représenté par la valeur de convergence) et de temps de réponse.

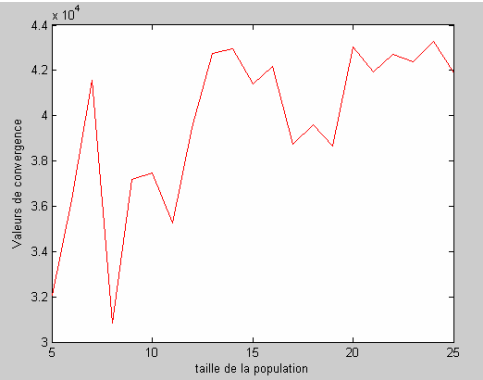


Figure III.27. *Evolution des valeurs de convergence en fonction de la taille de population.*

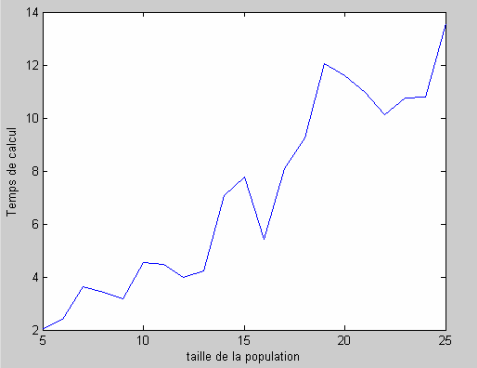


Figure III.28. *Evolution du temps de réponse en fonction de la taille de population.*

D'après la variation de l'influence de la taille de population sur la valeur de convergence et le temps de calcul résumé à travers les deux figures III.27 et III.28, nous optons pour régler le paramètre relatif à la taille de population en lui attribuant la valeur 13 (VC = 42739, T = 4.24s).

Le même principe a été utilisé pour fixer le paramètre relatif au nombre de générations. Donc, et suite à une série de tests sur un ensemble de données soumises au chiffrement et en utilisant la fonction fitness proposée, nous avons retenu certaines valeurs de paramètres qui permettent d'avoir un résultat exploitable en un temps de calcul très raisonnable (de l'ordre de 5 secondes). Ces valeurs sont données par le tableau III.5.

	Valeurs des paramètres de OEEA
Taille de la population	13
Nombre de générations	50
P_c	0.9
P_m	0.005

Tableau III.5. *Valeurs adoptées pour les paramètres d'OEEA.*

- **Résultats**

Après avoir décrit en détail notre méthode nous rapportons, dans cette section, les résultats de l'application de notre troisième algorithme proposé, OEEA, sur les données test citées précédemment (Texte 1, Texte 2, Lena et Logo) en adoptant le paramétrage fixé ci-dessus.

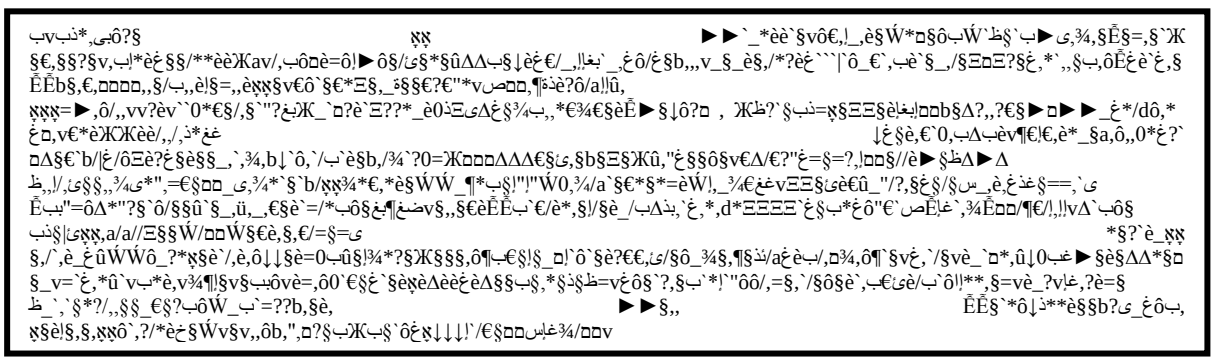
Nous constatons que la taille des clés est indépendante de la taille des données à chiffrer et elle est fixe pour des données de même nature (15323 bits pour les données texte et 7680 pour les données images).

		Taille donnée (éléments)	**Taille clé (bits)**	**VC**	**Temps calcul (s)**
Données texte	Texte 1	1084	15323	2173	6.54
	Texte 2	1151	15323	2284	6.73
Données images	Lena	131 X 131	7680	46034	3.37
	Logo	420 X 395	7680	431340	3.85

Tableau III.6. *Résultats obtenus par OEEA.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)

Figure III.29. ***(a)*** *Donnée originale Texte1,* ***(b)*** *Donnée chiffrée correspondante.*

Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

78 72 75 95 104 96 110 91 108 163 135 87 160 144 129 116 126 83 142
123 102 157 114 82 156 101 164 36 141 103 81 137 151 107 7 111 9 86
100 140 145 153 84 128 133 162 1053 1054 1055 1056 1057 136 120 1089 1090 1091 1092
1093 124 105 80 12 94 134 159 147 127 139 150 155 98 161 90 119 106 154
125 85 152 92 143 148 130 99 132 1281 1282 1283 1284 97 149 131 109 113 93
354 570 347 922 437 923 243 662 465 3 849 213 206 910 924 832 823 456 583
480 925 758 340 568 926 551 369 927 498 791 777 594 638 704 8 394 505 5
789 432 840 690 739 350 813 449 928 796 911 357 525 282 782 604 541 929 810
487 59 698 366 812 462 453 731 403 596 799 930 931 932 225 281 569 933 648
934 820 1386 1387 935 500 476 682 683 921 497 936 261 768 649 904 937 639 447
622 177 938 769 727 496 900 412 200 264 410 914 563 746 901 783 318 642 625
436 778 284 451 459 328 833 43 185 827 186 319 939 592 441 847 502 220 517
455 218 839 603 556 609 286 772 51 850 940 906 664 316 368 209 786 629 589
352 941 878 23 657 273 942 515 869 183 331 790 346 279 861 247 593 943 445
400 558 30 418 344 856 256 595 795 676 624 187 471 944 807 170 757 892 566
64 945 265 779 879 27 269 637 315 168 946 655 607 644 853 742 857 666 564
417 477 947 272 781 305 872 495 806 522 19 580 587 514 640 948 949 876 735
228 920 528 588 950 240 211 1231 1232 1233 915 458 33 287 312 951 545 863 860
571 667 490 952 953 954 888 916 913 299 391 905 671 378 442 669 575 591 726
811 865 202 398 358 399 955 956 314 454 172 341 234 621 546 214 643 737 230
957 223 788 891 259 577 736 958 416 181 263 755 486 50 289 618 959 292 846
864 174 960 519 433 395 434 20 902 961 907 962 1389 1390 1391 1392 1393 275 39
359 964 965 173 966 687 653 337 473 756 700 842 463 329 895 345 818 539 236
787 601 221 793 599 333 919 773 899 504 69 300 701 802 581 58 483 297 967
464 201 721 635 309 255 4 1 800 808 590 605 334 968 548 815 324 969 970
37 203 508 195 835 443 402 870 831 882 65 190 971 250 392 972 307 406 22
401 973 801 387 974 501 237 540 610 222 822 450 798 843 470 765 561 301 439
975 608 482 976 260 351 467 660 885 555 977 552 320 409 267 348 868 373 189
251 205 734 559 14 978 229 431 24 375 659 628 837 612 466 825 979 227 452
619 841 257 178 509 805 980 311 656 871 204 981 623 407 193 1060 1061 1062 1063
754 364 985 385 986 761 520 530 198 288 29 41 485 280 987 1230 1231 1232 1233
1234 290 764 988 989 469 1337 1338 1339 1340 1341 524 507 646 239 627 991 884 526
858 852 681 661 317 1197 1198 1199 1200 1201 826 992 322 699 353 1127 1128 1129 1130
475 993 47 994 887 1140 1141 1142 1143 877 672 70 430 384 684 192 995 527 996
303 997 674 383 828 1049 1050 1051 1052 342 2 326 60 678 491 866 283 26 404
792 1293 1294 1295 650 422 31 886 338 549 68 48 277 484 531 207 585 536 49
1000 705 258 600 651 521 1001 1002 523 438 363 460 15 293 513 381 747 444 550
753 310 1213 1214 1215 1216 1217 1003 898 335 248 245 1004 912 1005 386 274 529 634
630 1006 216 785 503 1257 1258 1259 584 356 620 1007 547 745 271 743 304 617 729
217 285 855 397 330 1008 1009 1010 557 516 875 750 771 355 512 728 296 472 492
1011 361 1012 880 582 809 166 427 446 803 420 494 1013 1371 1372 1373 1378 1379 1380
1381 167 576 784 1014 474 688 733 759 567 468 829 343 1271 1272 670 636 543 626
867 370 598 327 308 606 380 28 658 1015 774 295 844 665 61 560 188 268 732
821 1035 1036 1037 1038 1039 1040 874 371 360 673 663 196 1016 692 184 597 298 1017
614 415 16 1374 1375 723 1376 1377 55 752 715 244 685 215 1018 199 253 848 362
419 212 52 171 616 613 1019 917 425 390 534 1020 918 889 738 897 565 232 270
481 1021 252 488 535 641 396 838 893 797 578 896 766 1022 249 424 1023 1376 1377
1378 1379 388 834 1024 372 224 633 242 794 276 1025 854 631 1026 336 393 775 1356
1357 1358 1359 1360 691 532 675 197 235 499 194 572 730 1027 776 54 414 654 883
862 573 1028 506 349 909 679 586 56 780 873 493 332 428 231 602 377 632 542
824 423 816 53 749 1029 562 554 717 763 11 696 408 894 210 1030 179 702 645
751 804 680 1031 323 718 448 819 457 175 489 325 851 238 1032 306 814 741 374
233 611 461 1033 313 510 668 376 1034 57 748 724 44 1041 1042 1043 1044 1045 695
1046 1277 1278 1279 421 246 294 740 1058 1059 710 1064 1065 1066 1067 1068 719 1069 1070
1071 1072 1073 1074 1075 1076 1077 1078 1079 706 1080 77 73 76 74 79 117 146 1081
1082 1083 1084 63 1085 1086 1087 1088 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104
1105 1106 1107 1108 712 1109 1110 1111 1112 1113 25 169 881 413 382 1114 1115 708 1116
1117 1118 1119 1120 1121 1122 1123 714 1124 1125 1126 615 770 1131 720 1132 1133 1134 1135
1136 1137 1138 1139 18 1144 1145 1146 365 208 440 429 1147 1148 1149 1150 1151 1152 1153
1154 1155 35 1156 1157 1158 1159 1160 1161 903 537 1162 1163 1164 1165 1166 1167 1168 1169
1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 709 1180 1181 982 983 405 984 845 226
1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 66 1202 1203 1204
1205 713 1206 1207 1208 1209 1210 1211 1212 115 138 118 112 122 1218 1219 1220 1221 1222
1223 17 1224 21 1225 1226 1227 1228 1229 998 999 6 836 817 180 1235 1236 1237 1238
1239 1240 1241 1242 711 703 1243 1244 707 722 1245 1246 1247 1248 1047 1048 367 1249 1250
10 62 697 1251 1252 1253 1254 1255 1256 254 389 176 339 191 1260 716 1261 1262 1263
1264 1265 13 1266 1267 1268 1269 34 182 574 1270 1273 1274 693 1275 1276 1280 1285 1286
1287 725 1288 1289 1290 1291 1292 1296 1297 744 278 219 1298 1299 1300 1301 1302 1303 1304
1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 694 1320 32 1321
1322 158 88 89 121 40 1323 1324 1325 1326 1327 1328 1329 45 1330 1331 1332 1333 1334
1335 1336 1342 1343 67 1344 1345 1346 1347 1348 1349 1350 1351 46 1352 1353 42 1354 1355
544 426 291 647 762 262 963 411 1361 1362 1363 1364 1365 990 302 38 579 479 435

1366 1367 1368 1369 1370 538 859 686 241 908 760 533 165 830 511 266 553 1382 1383
1384 1385 518 677 890 1388 689 652 478

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسيًا في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز، لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال

(a)

gé¥¥§&◄¥c:¥~x┼Ч‌x;>ش╪♫♫⇑¦;>}&◄◄¦¥"<;g§§q5⇑ ¥¥;}¬;§¦<<<î{xcx;<¥¥§xקקt;:g¥¥;ى";<xx¥:§¥§}§;¦;¥¥x;¥<
³x۞۞⇑۞rgt¦¥xrxg◄t¬g¦x4và¦x¥x>§_⇑¦ƒîx¦;<ƒv*§r¥ىx§{¥x¥xqx;⇑ ↔↔îsxî§⇑;¥§<{Ч<j³:&r'קק¥╪xr:
ى۞q¬xƒ>♫¦¥<:q¸↔~®¥xé§g§۞۞;ش;g&x؟♫§xq;
۞¬{î<¥l§;c§¥xî¥§§x{xî۞0>¦◄хњхњхîàњ¦{¥x_:q§Ч§&r¥x¥étsЧЧ{&>¥x;;¦x╪ى;¥x&4§¥:&³³<"îx§ЧЧЧé§xq{¬<ى&{□¥à&x¦¥⇑⇑&>;<
¥î§§¥§ît¥¥rr§;î&;;>g{x¥x¥¥q;<x§;+ ┼┼┼>g§§{¥x;qs;îrxgxƒ¥⇑&x§¥{_î<³¦cx¥ƒ{x>;x۞x5¥⇑xrg;§x&¥¬xx: ◄x§ƒ³{xj¥¥q{î¥;¥j◄5&{g;;
њњq¥¥x&é;g;/>:;l¶‌x³xg¥î;¦;îxx۞۞q<¥:&&♫;¦p¦:/v¥xxxקp¥&§jx;x¥q◄◄¥>;{↔۞î~_::{;x_§xњ&{ﺔ¬¥gƒ¥éî>¥¥§§tl۞;¦x;x,x¶‌¥¦l:g{
qsq¥¥x<¦¶‌Ч;ºx¦>¦x;&q¥:§; ◄xàtgƒ{¦§¥⇑cq¥ggî³¦xî♫x<t{§>;{¥t¬q>¬╪g<§q:>®&_§;<ىr↔xx>¥¦¥Ч‌g<¥àtt~¥éxà5¥ƒ;_<<¥
♫¬xrîx¥§¥¥g~¥¦j¥{x¥↔↔¥¥g{¥§~§¥§¥¥⇑¥;;¥gƒ!lx;:¬;{gњњњx§╪╪xx&¦g↔↔{¥q۞x¥¥xx¥r{t{<®ggî;□۞!gx_ﺔ{_:x{t¦¶‌Чt>◄◄◄
4g¥x<<¥¥q;à¥⇑qg³x¥;¥ƒs<_và¥r§♫xàx;³ §>xƒƒ◄◄x۞tq¥{>îxt⇑x¥¥éxxs;_xىs;{>
קà'xq{ƒ<<קקx¶‌xrgsrv<xњњ4¥;x<x³sg;§ج¦x>¦t♫§§&g{<{;;¥qקx³gњњxƒ¥;ىxéj¥s۞њsss¥¥¥¥קxq_x;¥>¦<§î¥qstc5;sj³³¥s§>;:_ﺔé~g¥¥<x:4<
۞۞{¥ts§<{s§xxrq¥<¥{;:¥¥>&¥;x¦g♫:s;:®¥gsx;¥<v¦s/¬}>ى¥¸x⇑⇑s~sx;w{◄x:ىº¦ig§jx_¶‌Ч&;g_؟>s{&◄۞¥t¦5;s®{{♫⇑¥ش§¥¥>!x
<۞¥x<x;0>¦<>xt;¥î♫{x¥⇑îjs<¦;↔x¦¥><;&xrקקקs¥¬g&<x⇑:x{î³³x:îЧЧ_'{g§◄r§¥;<x/¥<⇑⇑{;¥{{{:□~¥x۞

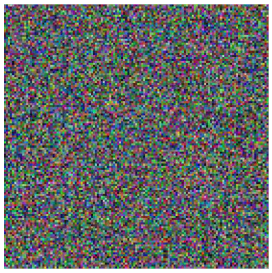
(b)

Figure III.30. **(*a*)** *Donnée originale Texte2,* **(*b*)** *Donnée chiffrée correspondante.*

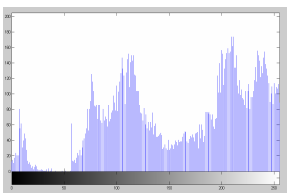
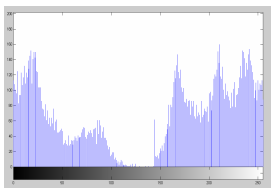
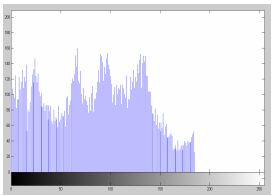
Clé de session (Taille$_{Clé}$ = 1,8704 K octets) :

59 1091 1 96 52 8 21 1200 47 54 71 16 43 14 1366 1369 25 64 26
1271 30 32 89 15 65 68 53 28 1303 1305 42 84 81 78 1090 34 24 62
20 58 74 56 39 1117 4 85 99 1324 18 98 1306 29 72 55 66 5 45
49 17 40 23 73 80 97 50 608 102 848 575 286 1220 1223 167 980 869 803
925 1201 134 310 1278 622 866 190 472 727 1118 535 818 1367 114 377 326 835 553
1368 166 488 100 1270 874 206 981 307 232 873 880 676 538 712 578 982 305 860
1325 899 933 842 332 865 315 713 239 445 619 203 939 983 785 192 1302 163 749
872 1327 297 605 513 722 270 940 269 814 198 943 201 570 195 422 746 897 802
402 904 945 250 794 560 691 1276 217 693 514 1304 852 935 804 666 518 1222 133
467 915 156 910 407 1221 170 257 738 542 692 912 984 508 443 245 815 671 372
63 61 67 985 1277 174 128 477 930 639 986 800 987 667 258 988 456 309 989
817 856 688 607 559 520 459 577 188 439 369 325 643 314 172 363 796 592 312
105 193 124 157 148 582 669 822 566 390 779 806 397 1187 1188 573 849 901 917
813 149 317 285 696 180 337 725 824 634 1057 223 990 829 554 953 466 732 242
733 905 647 308 168 101 487 958 1326 793 572 812 454 517 449 991 831 366 540
704 868 356 212 913 847 321 408 1059 280 623 909 484 271 137 662 220 433 324
881 1375 841 1333 896 837 624 992 891 1334 649 724 581 161 502 1275 545 685 924
843 1058 473 585 1374 828 833 275 993 382 450 176 107 1285 375 427 291 929 795
299 362 599 1283 863 178 736 515 531 937 331 292 994 742 888 1332 313 596 126
381 648 359 411 489 431 376 234 120 934 1284 175 714 522 955 534 247 614 598
652 1378 995 143 410 601 996 997 1273 453 883 158 208 719 298 430 689 446 462
357 104 840 928 184 413 303 657 364 399 870 256 396 636 424 159 391 1384 1385
115 1319 702 277 279 419 241 329 1000 734 698 627 469 871 521 690 602 262 140
342 687 1376 1001 464 887 350 116 268 1272 1274 895 720 465 267 197 1002 588 541
468 361 808 132 204 141 1377 287 444 547 438 150 761 300 221 425 921 1320 348
403 737 611 496 304 862 171 586 709 792 820 834 127 406 274 229 145 194 1003
886 546 392 922 288 673 731 448 205 276 561 263 1064 956 625 1004 186 227 492
1005 442 254 593 1006 1007 395 584 260 902 1065 530 968 590 334 389 1008 558 708
877 393 412 475 678 646 526 1203 1204 273 533 832 470 243 1009 579 884 474 451
786 850 908 723 1066 405 501 1010 1011 117 677 1012 498 658 1382 699 765 819 1013
214 1014 1015 233 423 135 1016 650 164 504 615 721 222 457 900 892 911 455 954
1017 642 1018 351 151 838 138 500 1381 505 1019 437 458 226 436 385 580 946 1020
179 131 400 653 173 1101 306 481 213 218 240 686 707 106 626 380 1021 511 358
333 146 1063 620 674 610 111 811 567 1022 185 507 603 878 338 706 1023 283 482
1100 266 379 323 799 926 432 1098 906 893 951 845 1024 388 668 568 718 855 764
110 701 949 224 112 864 756 551 125 920 680 1206 209 142 409 1025 207 612 844
670 633 587 367 645 302 130 591 1312 805 383 621 1026 169 679 103 349 1027 660
655 781 571 544 861 641 730 480 1028 322 1029 628 705 255 682 493 1030 1031 162

890 211 919 1099 478 927 1032 661 355 1205 768 932 1033 853 1067 236 1034 340 681
556 527 801 537 486 182 327 613 189 606 604 703 265 384 1035 617 231 1193 246
165 942 1036 728 640 821 875 200 597 659 261 717 441 415 854 1037 108 1192 569
823 374 281 894 539 923 807 452 202 154 1359 483 440 177 882 485 248 562 319
495 160 387 695 259 697 122 876 401 1190 651 576 414 879 1038 230 966 635 426
519 916 600 950 210 352 118 751 237 918 563 386 963 282 630 656 754 365 797
1182 1183 609 716 295 740 339 476 370 1039 330 378 675 1055 228 885 557 851 113
715 272 594 741 747 826 238 858 672 574 109 497 1191 447 152 353 121 663 494
760 532 499 129 898 404 1040 583 1184 524 565 638 344 294 416 253 644 629 119
938 284 429 1056 320 360 479 555 798 346 825 183 710 947 1041 936 1080 318 595
249 632 827 199 743 1180 336 637 683 435 421 191 729 684 944 810 460 512 509
503 857 1042 418 1094 739 654 1043 1044 525 529 839 1093 809 510 1181 664 771 244
394 311 434 859 1095 931 368 589 316 123 941 957 948 181 139 354 1045 1046 776
784 1047 1048 1049 1050 830 398 264 335 1051 1052 775 1053 1054 420 543 75 76 1060
979 1061 1062 1068 1069 1070 1071 1072 762 1073 1074 196 506 1075 1076 1077 969 1078 1079
11 9 1123 1124 1081 1082 293 373 1083 1084 973 965 1085 1086 296 564 1087 1088 1089
83 36 90 77 6 92 3 91 10 95 33 1092 371 219 1096 1097 726 251 1102
252 903 1103 1104 1105 1106 1107 147 889 552 1108 1109 774 1110 1111 1112 790 1113 773
1114 735 289 1115 1116 37 69 1119 1120 1121 1122 1125 788 975 1126 1127 461 780 1128
960 1129 1130 341 215 1131 1132 1133 971 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143
1144 1145 1146 976 1147 1148 1149 1150 769 1392 1393 1151 1152 1153 962 1154 1155 1156 225
694 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 216 914 153 1347
301 1171 1172 1173 1174 1175 752 1176 1177 1311 290 836 1178 1179 1185 1186 490 471 550
1189 1353 1194 766 1195 782 1196 1313 1197 1198 1199 665 428 1202 907 616 763 1207 977
1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 767 379 321 1224 758 1225 1226
1227 1228 1229 523 278 1230 1234 1235 770 1236 1237 1238 1239 748 1240 1241 1242 22 31
44 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1310 1254 1255 967 1256 1257 1258
1259 1260 753 964 1261 1262 816 1265 1266 618 516 1267 1268 1269 789 1279 1280 13 12
48 41 1281 1282 783 463 1286 1287 1288 1289 1290 959 1291 1292 1293 1294 70 7 1295
1296 1297 548 187 1298 1299 1300 1301 136 536 1307 1308 1309 970 744 1314 1315 757 777
1316 1317 1318 791 1321 1322 1323 27 35 1328 1329 144 345 952 1330 1331 961 1335 1336
1337 51 79 1338 1339 1340 711 235 1341 1342 1343 974 1344 745 1345 1263 1346 94 86
1348 1349 1350 867 155 1351 1352 528 750 1354 998 328 999 1355 1356 60 88 1357 82
46 38 87 57 1358 772 1360 1361 1362 343 549 491 417 1363 1364 1365 1370 1371 767
1372 1373 347 700 1379 1380 778 631 1383 759 846 1264 1386 1387 93 2 1388 1389 755
1390 978 787 972 1391

(a) ***(b)***

(c)

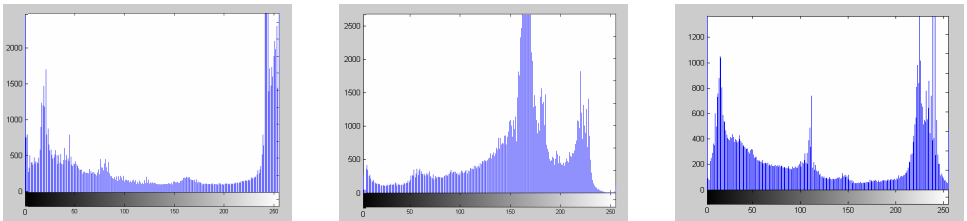
Figure III.31. *L'image test Lena :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session ($Taille_{Clé}$ = 0,9375 K octets) :

369 423 529 171 397 412 474 15 689 64 112 390 267 21 118 304 6 28 574
465 356 338 717 638 339 110 243 458 508 450 361 608 635 115 593 749 151 742
575 716 428 109 486 584 294 722 521 141 87 695 373 225 452 602 679 13 208
85 468 634 709 298 70 297 514 601 439 359 320 251 222 292 319 765 456 614
142 645 625 288 383 217 125 476 718 274 236 729 483 567 114 487 492 303 542

426	628	675	713	296	505	370	730	409	395	764	641	396	669	460	337	264	230	204
623	358	293	418	150	520	134	328	469	762	440	533	651	551	535	286	438	462	103
355	347	63	158	16	561	205	410	643	703	432	519	693	247	620	346	40	736	366
129	652	164	327	371	506	363	154	241	447	434	624	585	163	748	376	380	144	407
59	143	667	252	10	536	548	766	360	335	455	311	497	200	466	249	751	35	119
165	341	36	741	398	448	656	18	65	233	734	17	531	512	595	660	104	586	81
435	299	101	92	309	454	436	710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579	352	739	698	306	253	577	248
735	515	342	285	61	219	138	568	705	658	427	511	532	210	357	307	153	633	711
98	433	619	691	60	20	137	733	485	324	659	419	719	684	269	392	754	68	259
284	467	295	48	300	224	541	600	715	630	613	172	66	425	558	604	38	692	34
557	263	258	384	51	317	375	111	325	146	701	472	388	607	289	94	644	377	745
580	500	598	496	240	443	594	364	554	74	637	187	712	655	747	576	145	47	501
417	490	622	162	491	626	578	699	750	14	128	493	282	226	459	351	589	209	582
545	740	596	147	275	121	362	470	53	193	291	333	290	179	523	527	42	30	402
738	525	113	374	664	646	507	609	685	756	189	700	148	260	654	464	191	350	378
176	257	538	599	603	11	702	372	571	237	12	96	170	192	653	88	416	540	25
180	522	642	166	72	246	194	707	680	33	639	159	215	75	728	227	516	681	344
90	99	559	127	513	265	7	23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555	69	708	345	494	271	238	726
393	534	132	385	587	686	647	368	203	3	481	482	332	199	167	687	560	539	495
743	44	583	91	610	690	336	31	261	403	379	255	725	498	323	386	198	572	220
484	73	322	677	606	340	272	343	58	503	381	26	421	517	761	79	676	334	348
760	49	663	235	597	566	313	627	39	414	618	518	479	55	752	186	694	563	649
175	223	133	549	420	277	510	727	305	391	228	632	404	196	206	84	52	184	755
212	581	763	429	107	697	724	590	77	737	445	71	682	502	509	120	254	413	758
640	149	616	43	195	242	239	211	611	453	757	218	86	19	310	387	2	105	117
424	244	504	116	449	591	8	229	605	399	629	182	57	461	314	588	177	234	190
102	106	714	122	617	612	216	673	312	431	326	621	80	382	721	93	45	156	665
553	767	475	174	329	394	415	547	188	331	543	365	308	46	67	670	330	157	354
302	389	480	367	232	768	27	207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631	41	471	131	201	245	54	202
444	556	544	720	270	32	688	759	442	279	478	136	197	123	273	671	437	262	457
672	268	349	95	50	488	706	546	183	221	178	160	528	696	62	9	214	615	573
489	430	97	744	22	753	524	446	213	746	731	181	231	283	661	570	401	276	78
400	4	562	526	683	301	89	477											

(a) ***(b)***



(c)

Figure III.32. *L'image test Logo :* ***(a)*** *Image originale,* ***(b)*** *Image chiffrée,*
(c) *Histogrammes de l'image chiffrée.*

Clé de session (Taille$_{Clé}$ = 0,9375 K octets) :

503	463	515	233	412	71	181	502	153	365	302	449	225	362	500	33	523	198	519
471	404	521	497	145	496	504	494	205	138	505	6	522	214	149	451	422	498	146
520	63	453	320	516	510	350	287	495	39	387	385	20	115	80	361	472	69	513
213	204	105	425	68	311	507	506	208	508	24	41	499	511	237	54	415	509	37
518	517	52	458	384	493	390	8	191	78	501	40	163	514	229	512	99	574	372
579	351	396	36	389	17	370	312	626	436	5	1	284	118	629	334	594	625	174
704	648	28	534	469	657	342	411	479	401	45	533	234	675	379	23	235	231	223
77	324	729	634	548	217	555	142	673	159	346	568	551	85	195	343	333	408	158
10	563	532	373	264	73	242	359	179	280	473	709	605	128	327	82	756	308	638
588	152	31	766	272	285	166	667	714	51	613	157	439	265	561	271	314	277	491
286	32	125	2	332	194	547	552	241	156	443	661	143	455	431	120	215	484	578
434	526	647	344	257	371	744	168	199	584	260	707	486	763	369	244	524	298	540
689	405	764	196	595	538	345	478	636	464	95	216	29	567	46	735	57	331	397
752	426	377	288	133	114	134	62	424	668	720	399	200	336	696	459	141	553	173
702	169	148	171	460	622	210	35	409	201	570	209	444	14	624	558	299	483	66
366	184	224	600	100	671	738	293	677	549	192	193	291	554	230	76	481	639	67
124	545	620	226	93	529	303	136	164	470	603	150	687	419	304	683	414	581	74
565	127	245	144	577	337	760	475	557	705	301	49	418	137	44	110	7	79	306
207	13	635	380	276	758	420	278	248	238	292	630	457	47	437	583	445	375	723
354	492	684	255	679	81	263	119	221	61	682	751	669	652	104	681	659	290	92
121	454	188	172	564	355	754	708	489	251	394	220	12	60	97	126	566	604	403
601	15	395	429	330	715	645	546	490	759	108	154	423	34	441	87	741	575	643
666	101	19	693	686	718	111	631	326	542	617	674	706	84	178	543	381	50	167
733	402	716	654	462	614	239	300	254	725	262	619	232	428	305	593	335	161	427
83	250	341	180	374	632	465	406	719	183	378	623	382	243	147	740	765	296	649
658	736	582	252	165	712	745	89	413	569	160	106	633	329	177	608	340	461	589
537	261	456	38	103	607	640	627	98	4	30	597	416	281	596	599	536	253	734
26	742	477	642	266	438	135	386	573	295	339	721	711	190	487	753	591	539	162
737	610	726	313	140	197	70	16	525	621	637	86	749	182	572	11	474	139	612
730	347	598	43	556	541	319	48	761	187	664	748	275	646	672	258	151	219	710
259	256	236	348	450	609	109	325	186	571	580	678	703	435	430	694	606	376	274
240	328	206	393	175	466	615	315	587	75	383	602	616	527	25	273	762	448	697
113	452	132	155	724	364	739	294	485	728	202	750	476	650	544	116	480	688	42
338	530	352	123	743	247	641	676	368	211	176	107	467	698	322	55	560	56	731
685	270	170	767	227	655	690	279	310	746	644	203	732	680	663	662	417	64	528
651	433	691	96	59	246	585	757	112	318	283	618	628	22	297	665	592	535	94
65	58	222	102	586	699	3	695	358	228	717	212	392	576	531	289	130	88	356
388	316	122	768	660	421	446	185	611	268	670	410	398	468	249	321	755	72	9
18	692	700	117	323	722	447	307	353	360	131	90	407	129	363	562	432	488	590
27	367	400	559	349	317	391	440	656	653	727	218	442	482	282	713	53	747	91
269	309	189	267	21	701	550	357											

III.5. Discussion et évaluation des résultats

Un bon crypto-système doit satisfaire les six points clés de Kerckhoffs [Kerc, 1883] et ceux de Shannon [Shan, 1949]. Plus explicitement, la qualité d'un crypto-système se mesure principalement par sa résistibilité face aux différents types d'attaques. La technique de cryptage proposée à travers nos trois algorithmes proposés PosESecL1, PosESecL2 et OEEA et qui consiste en une perturbation évolutionnaire de la donnée originale, assure une bonne sécurisation contre les attaques les plus destructives. Les critères utilisés pour évaluer leurs sécurités sont les suivants : la vitesse de l'algorithme, l'attaque statistique, l'attaque différentielle, l'attaque exhaustive et l'analyse de l'espace des clefs.

III.5.1 Vitesse des algorithmes

En plus du paramétrage, la vitesse d'un algorithme évolutionnaire dépend étroitement du codage utilisé. Un mauvais codage affecte non seulement la qualité de l'algorithme en termes d'optimalité de solutions mais aussi en termes de temps de convergence. Dans le cas de nos deux premiers algorithmes proposés, PosESecL1 et PosESecL2, le codage dépend de la taille de la donnée à chiffrer ce qui affectera la vitesse des algorithmes puisque le temps de traitement des données de grandes tailles sera beaucoup plus grand que celui de traitement des données de petites tailles, chose due à la taille des chromosomes manipulés durant les générations de l'évolution. Ceci se voit clairement à travers les résultats obtenus par ces deux algorithmes où le temps de convergence de PosESecL1 est meilleurs que celui de PosESecL2

du fait que la taille des chromosomes codant une donnée pour le PosESecL2 est trois fois plus grande que celle des chromosomes codant la même donnée pour le PosESecL1. Toutefois, le codage proposé à travers OEEA unifie la taille des chromosomes codant des données de même nature (textes ou images) et de différentes tailles. Ainsi, le temps de traitement est indépendant de la taille des données traitées. En plus et d'après les résultats expérimentaux présentés précédemment, nous constatons que le temps de convergence de ce dernier algorithme est très raisonnable (de l'ordre de 4 secondes).

Comparons, maintenant, la vitesse de nos trois algorithmes développés avec celle des principaux crypto-systèmes (AES du coté des crypto-systèmes symétrique, RSA du coté des crypto-systèmes asymétrique et SEC qui est un algorithme de chiffrement exploitant les algorithmes génétiques). Le tableau 7 et la figure 33 présentent un résumé des temps de chiffrement et de déchiffrement de chacun des algorithmes cités.

	PosESecL1	**PosESecL2**	**OEEA**	**AES**	**RSA**	**SEC**
Temps de chiffrement (s)	24.5	46.15	3.37	0.5	42.8	1.8
Temps de déchiffrement (s)	0.26	0.75	0.12	0.08	42.5	0.06

Tableau III.7. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

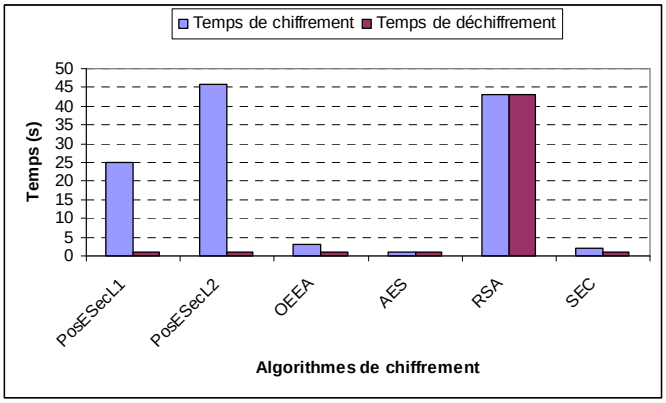


Figure III.33. *Temps de chiffrement et de déchiffrement de PosESecL1, de PosESecL2 et de OEEA en comparaison des principaux standards de chiffrement.*

D'après le tableau et la figure, ci-dessus, présentant le temps de chiffrement et celui de déchiffrement de nos algorithmes en comparaison des algorithmes SEC [Omar, 2007], RSA [Mene, 1996] et AES [www4], nous remarquons que :

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement beaucoup plus petit que celui de RSA. De son coté PosESecL1 possède aussi un temps de chiffrement/déchiffrement meilleur que celui de RSA. Toutefois, PosESecL2 a un temps de chiffrement de l'ordre de celui de RSA et un temps de déchiffrement meilleur que celui de RSA.
- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin de celui d'AES. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes PosESecL1 et PosESecL2 sont plus grands que celui d'AES.

- ✓ Notre algorithme OEEA possède un temps de chiffrement/déchiffrement pas trop loin que celui de SEC. Toutefois, les temps de chiffrement/déchiffrement de nos deux autres algorithmes sont plus grands que celui de SEC.

De même et d'après les résultats expérimentaux obtenus et présentés précédemment, nous constatons que le temps de calcul de OEEA est indépendant de la taille de la donnée à chiffrer du fait de l'unicité de taille de codage de toutes les données de même nature soumises au chiffrement. Toutefois, ce n'est pas le cas pour PosESecL1 et PosESecL2 où le temps de calcul des deux algorithmes est strictement dépendant de la taille de la donnée à chiffrer. Ainsi, OEEA est l'algorithme ayant un temps moyen de calcul le plus adapté.

III.5.2 Niveau de confusion

La confusion est l'une des notions essentielles énoncées par *Shannon* sur lesquelles se base un algorithme de cryptage. Elle sert à cacher la relation entre la donnée en clair (originale) et la donnée chiffrée correspondante. Donc, elle vise à rendre la donnée aussi peu lisible que possible.

Dans notre cas, les deux modes de chiffrement développés, à savoir le chiffrement à base de positions et le chiffrement à base d'occurrences, sont de niveaux de confusion différents.

Pour le premier, les deux algorithmes proposés dans ce mode sont aussi de niveaux de confusion différents. En effet, le niveau de confusion de PosESecL2 est meilleur que celui de PosESecL1 du fait que PosESecL2 exploite les positions des éléments codant les différents composants de la donnée, considéré chacun comme unité séparée (gène) du codage final de la donnée (chromosome). Ainsi, la donnée chiffrée aura de grande chance de contenir de nouveaux éléments ou composants ne figurant pas dans la donnée originale. Cette chose complique considérablement, voire même, pénalise toute cryptanalyse possible comme nous allons le démontrer dans les sections qui suivent. Toutefois, PosESecL1 est de niveau de confusion strictement dépendant de la taille de la donnée à chiffrer puisqu'il exploite les positions des éléments de la donnée. Donc dans le cas de données de petites ou de moyennes tailles, une attaque exhaustive finira de trouver la bonne combinaison initiale de positions et, ainsi de casser l'algorithme.

Pour le deuxième mode de chiffrement développé, il offre un très bon niveau de confusion grâce à l'utilisation d'un codage bâti autour de la notion de nombres d'occurrences. Ce mode ne présente ni information utile pouvant être exploitée par un cryptanalyse ni possibilité de reproduction de la donnée en se basant uniquement sur cette donnée surtout que le codage adopté assure que les composants de la donnée (caractères ou pixels) changent dans la donnée chiffrée par rapport à la donnée originale. En plus, le niveau de confusion de OEEA, l'algorithme développé dans ce mode, est meilleur que celui de PosESecL2 et ainsi de celui de PosESecL1 du fait que l'espace des nouveaux caractères qui peuvent apparaître dans une donnée chiffrée par rapport à la donnée originale correspondante est beaucoup plus grand dans OEEA que dans PosESecL2.

III.5.3 Attaque statistique

Ce type d'attaque considère le crypto-système comme une boite noire, il analyse statistiquement les entrées et les sorties de ce système. Pour ce faire, nous avons utilisé les mesures suivantes dans le but d'évaluer ou de quantifier la différence entre l'image originale et l'image cryptée correspondante :

Le facteur *NPCR* (*Number of Pixels Change Rate*) donné par l'expression (III.39), l'erreur absolue moyenne (*MAE : Mean Absolute Error*) donnée par l'expression (III.41) et l'erreur

quadratique moyenne (*MSE :Mean Square Error*) donnée par l'expression (III.42). Le tableau 8 résume les valeurs des différentes mesures obtenues après les tests qui ont été effectués sur l'image Lena et l'image chiffrée correspondante obtenue dans le cas de nos différents algorithmes proposés.

	NPCR	**MAE**	**MSE**
PosESecL1	0.6314	0.91	0.58
PosESecL2	0.9073	0.97	0.61
OEEA	0.9955	1.03	0.75

Tableau III.8. *Niveaux de confusion.*

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \tag{III.39}$$

$$D(i,j) = \begin{cases} 0 Si\ \mathrm{Im}_O(i,j) = \mathrm{Im}_C(i,j) \\ 1 Si\ \mathrm{Im}_O(i,j) \neq \mathrm{Im}_C(i,j) \end{cases} \tag{III.40}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \tag{III.41}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \tag{III.42}$$

D'après les résultats d'application des mesures quantifiant la différence entre l'image originale et ses versions chiffrées calculées par nos algorithmes, il est clair que la majorité des pixels formant l'image originale ont changé soit de positions (dans le cas d'application de PosESecL1) ou de contenu (dans le cas d'application de PosESecL2 ou de OEEA) pour, ainsi, calculer l'image chiffrée. En effet cela se voit surtout à travers les valeurs de NPCR qui calcul le niveau de différence entre les deux images où les résultats obtenus sont proches de la valeur optimale qui vaut un (01). Toutefois, ces valeurs ne sont pas les mêmes pour les trois algorithmes, chose qui est tout à fait normale vu que les algorithmes sont de niveau de confusion différent (voir section précédente). Et comme nos algorithmes présentent l'avantage d'être non déterministes, donc chaque nouvelle application d'un même algorithme donnera lieu à des résultats différents ce qui entraîne que l'application des mesures NPCR, MAE et MSE donnera lieu, de son tour, à de nouvelles valeurs. Ainsi, aucune propriété statistique ne peut être déduite d'une donnée chiffrée et l'attaque statistique ne possédera aucune chance de casser nos algorithmes.

III.5.4 Attaque différentielle

Cette attaque essaye de tirer des conclusions sur le fonctionnement d'un algorithme de chiffrement en comparant des versions chiffrées de plusieurs données originales et dans un meilleur cas des versions chiffrées de plusieurs blocs formant une même donnée. Ainsi, dans le cas des algorithmes de chiffrement par blocs (DES [Stin, 1996], [Biha, 1991], [Mats, 1994], 3DES [Stin, 1996], [Gant, 2001], AES [Lepr, 2000], etc), quand la donnée (surtout pour le cas d'images) contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement, ce qui fait que la donnée cryptée contiendra des zones texturées ; mais vu que nos algorithmes traitent la donnée en une seule passe, c'est-à-dire ils opèrent sur la donnée totale et non pas des blocs de la donnée ; alors la cryptanalyse

différentielle sera mise à l'écart. De ce fait aussi, nos algorithmes présenteront une certaine robustesse au bruit par opposition aux algorithmes de chiffrement par blocs, et qui se traduit par le fait qu'une erreur sur un bit chiffré ne va pas propager des erreurs importantes dans tout le bloc courant et par la suite dans toute la donnée lors de la recombinaison des blocs.

Même si le cryptanalyste pense à essayer de tirer des observations à partir des résultats de cryptage de plusieurs données pour essayer d'imiter le fonctionnement des algorithmes, l'approche évolutionnaire pseudo-aléatoire complique considérablement la tâche de ce type d'attaque. Une sensibilité à la donnée originale est aussi un point avantageux des algorithmes proposés vu que la donnée cryptée correspondante à une certaine donnée originale change d'une instanciation du problème à une autre. Autrement dit, nos algorithmes développés sont des algorithmes non déterministes à l'inverse du RSA, par exemple, qui a nécessité au préalable de randomiser les données soumises au chiffrement pour éviter les attaques exploitant les modèles connus des données en clairs [Ster, 2004].

III.5.5 Attaque exhaustive

L'attaque exhaustive ou encore dite attaque à force brute est une attaque qui demeure fatale pour les crypto-systèmes utilisant des clés de petites tailles. À ce stade, nos trois algorithmes proposés présentent des niveaux de sécurité différents. La robustesse de PosESecL1 est strictement dépendante de la taille de la donnée originale. PosESecL2 présente un niveau de sécurité plus élevé que celui de PosESecL1 du fait que sa clé soit de taille quatre fois plus élevée que celle de PosESecL1 dans le cas de manipulation de données textes et de trois fois plus grande dans le cas de manipulation des données images. Ainsi, et vu qu'à l'heure actuelle la taille de la clé assurant une résistibilité face à une attaque exhaustive est de 512 bits, alors le cryptage par PosESecL1 de toute donnée possédant une taille inférieure à 512/le nombre de bits nécessaires pour coder la taille de la donnée (nombre de caractères ou de pixels), sera exposée au risque d'une attaque réussite par force brute. C'est le cas aussi pour les données cryptées par PosESecL2 et qui sont de taille inférieure à : 4*(512/le nombre de bits nécessaires pour coder la taille de la donnée texte) ou de 3*(512/le nombre de bits nécessaires pour coder la taille de la donnée image). Toutefois, OEEA ne peut en aucun cas être cassé par une telle attaque vue que la taille de la clé utilisée est largement sécuritaire. Ainsi, OEEA est l'algorithme le plus sûr parmi nos trois algorithmes développés.

Considérons le texte à chiffrer « **there is only one God and Mohamed his prophet** », nous reportons dans le tableau suivant (tableau 9) les tailles de clés de chiffrement de nos trois algorithmes développés en comparaison de DES, 3DES, AES et SEC, ainsi que le nombre d'opérations nécessaires pour générer toutes les combinaisons de bits formant les clés pour chaque algorithme. Ceci donnera une idée sur la complexité de l'attaque exhaustive appliquée contre chacun des algorithmes où, il se voit clairement que l'attaque exhaustive menée contre nos algorithmes est la plus complexe.

	Taille de clé (bits)	**Nombre d'opérations**
DES	56	2^{56}
3DES	128	2^{128}
AES	256	2^{256}
SEC	240	2^{240}
PosESecL1	352	2^{352}
PosESecL2	1408	2^{1408}
OEEA	1393	2^{1393}

Tableau III.9. *Complexité de l'attaque exhaustive.*

III.5.6 Analyse de l'espace des clés

Ici, nous testons la sensibilité du processus cryptographique proposé aux clés utilisées. Dans notre cas, la clé générée est une clé calculée à partir de la donnée originale et de la donnée chiffrée correspondante, donc elle change d'une instanciation du problème à une autre. Ainsi, le cryptage successif d'une même donnée donne lieu à un ensemble de données chiffrées différentes ce qui entraînera, à chaque fois, la génération d'une clé de session différente pour le chiffrement d'une même donnée originale. Cela dotera notre méthode de cryptage d'une très grande sensibilité aux clés puisque toute clé interceptée d'une manière illégale ne servira que pour le déchiffrement d'une seule version chiffrée d'une même donnée donc elle ne sera plus utile par la suite.

Le problème rencontré dans le cas des crypto-systèmes symétriques, et par conséquent dans le cas de nos algorithmes développés qui sont des algorithmes symétriques, est la communication de la clé secrète au destinataire en vue de l'utiliser dans l'extraction de l'information originale (le déchiffrement). Pour ce faire, certains concepteurs ont proposé de combiner les fonctionnalités de la cryptographie symétrique et asymétrique. C'est le cas du PGP par exemple. Il crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ; puis, il crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire. Ainsi, le même principe peut être adopté pour sécuriser le transfert de nos clés en leurs chiffrant par un crypto-système asymétrique. Dans ce cas un schéma hybride de transmission sécurisée peut être envisagé comme le montre la figure suivante :

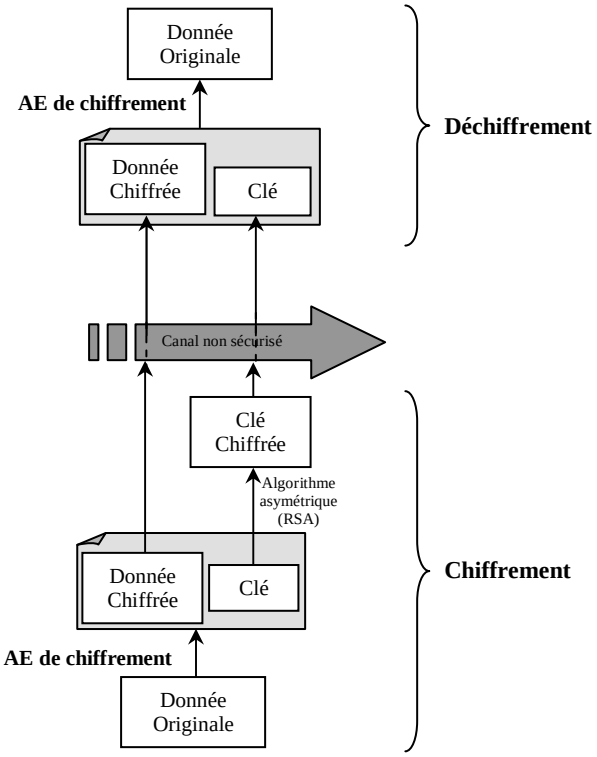


Figure III.34. *Schéma hybride de transmission sécurisée.*

Or, pour nos algorithmes PosESecL1 et PosESecL2, il est clair que la taille de la clé dépend strictement de la taille de la donnée. Le problème qui se pose lors du chiffrement d'une donnée de grande taille, c'est que la taille des clés générées elle aussi devient grande ce qui consommera un grand temps de chiffrement de clés du fait que les crypto-systèmes publiques sont déjà lents. Le problème s'aggrave de plus en plus lorsque la taille de la clé

devient égale ou dépasse la taille du codage de la donnée en termes de bits. Dans ce cas, l'utilisation de ces deux algorithmes devient inutile et il sera plus intéressant d'utiliser un crypto-système publique pour chiffrer la donnée au lieu de la clé générée malgré que la sécurité de tels crypto-systèmes soit grandement affectée par les choix paramétriques (p et q dans le cas de RSA [Zimm, 2005]). Pour remédier à cette anomalie, nous proposons de coder (ou bien comprimer) la clé générée par un système de codage sans perte, tel que le codage de Huffman par exemple [Huff, 1952], en vue de réduire sa taille avant de la chiffrée par un crypto-système asymétrique, puis d'envoyer au destinataire le package englobant la donnée chiffrée et la clé codée chiffrée. Lors de la réception, on procède inversement. On déchiffre la clé codée en utilisant la clé publique du destinataire, puis on la décode pour obtenir la clé proprement générée par PosESecL1 ou PosESecL2 qui va permettre de reproduire la donnée originale. Ainsi, le schéma du processus de sécurisation englobant cette dernière fonctionnalité sera le suivant :

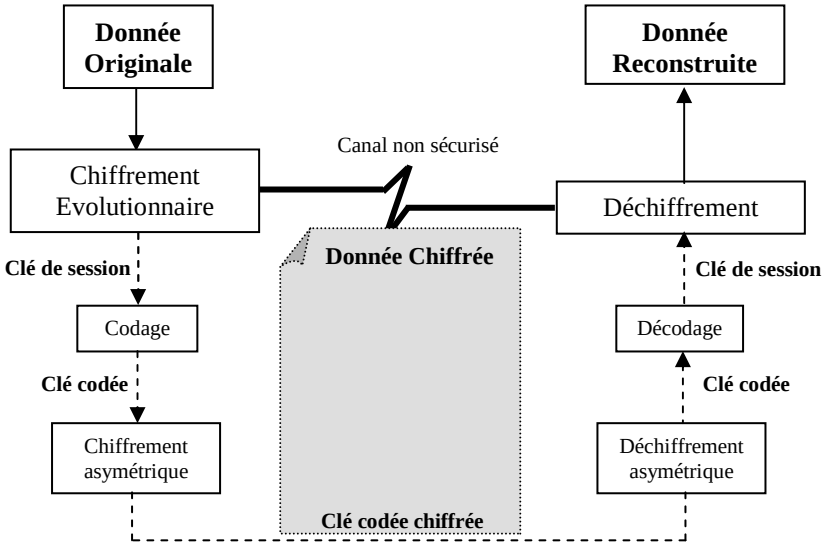


Figure III.35. *Schéma général du processus de chiffrement et de transmission sécurisée de la clé générée par chiffrement publique.*

Une deuxième solution de transmission sécurisée de la clé de session générée peut être envisagée dans le cas de manipulation de données images. Il s'agit de la technique de tatouage. Après avoir codé la clé générée par notre algorithme de la même manière proposée dans le cas de la première solution de transmission sécurisée de la clé (en utilisant un codage de Huffman par exemple), nous proposons, ici, de marquer l'image chiffrée par la clé de session codée. Ainsi, le schéma du processus de sécurisation englobant cette deuxième solution de transmission de la clé peut être résumée par le schéma de la figure III.36.

Cette deuxième solution de transmission est conditionnée par la taille de la clé codée pour satisfaire le compromis Robustesse-Capacité-Invisibilité lors du tatouage [Katz, 2000], [Barn, 2004], [Koba, 1990].

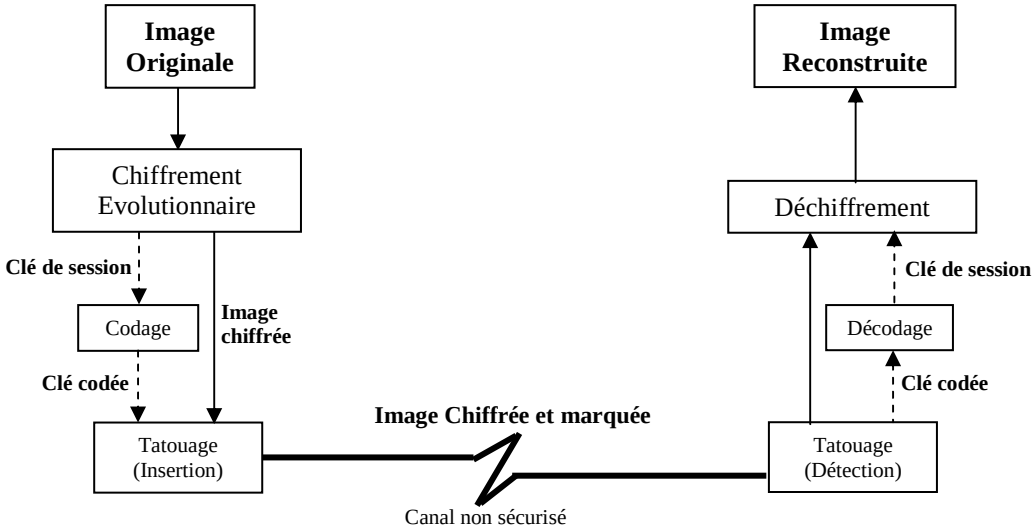


Figure III.36. *Schéma général du processus de chiffrement d'images et de transmission sécurisée de la clé générée par tatouage.*

III.6. Conclusion

À travers le présent travail, nous avons démontré que les algorithmes évolutionnaires, inspirés fondamentalement de la sélection naturelle des espèces, peuvent être également appliqués au cryptage de données. En effet, avec ces algorithmes, nous avons ramené le problème de chiffrement en un problème d'optimisation.

L'application de ces algorithmes requiert le réglage d'un certain nombre de paramètres pour la phase de chiffrement. La taille de la population initiale, le taux d'exploration et le nombre de générations affectent le temps de convergence des algorithmes, quant aux probabilités *Pc* , *Pm*, elles possèdent, théoriquement, une influence directe sur la qualité du résultat. Or, dans notre problème, les opérateurs génétiques (croisement et mutation) n'agissent pas sur l'information traitée (caractères du texte ou pixels de l'image) mais sur leurs coordonnées ; ils sont utilisés comme un outil pour explorer le reste de la population. De ce fait, les paramètres P_c , P_m affectent également la convergence des algorithmes.

Dans la pratique, les paramètres des algorithmes évolutionnaires sont réglés approximativement par tâtonnement [Gold, 1989]. Suite à une série de tests sur diverses données et en utilisant les fonctions fitness proposées, nous avons retenu certaines valeurs qui permettent d'avoir un résultat exploitable (donnée chiffrée) en un temps de calcul variant d'un algorithme à l'autre.

Les résultats obtenus montrent que les schémas proposés présentent des aptitudes dans la confusion et dans la sensibilité à la donnée originale qui les rend loin des attaques différentielles. De même, et pour pénaliser ou plus ou moins compliquer la tâche de l'attaque exhaustive, une deuxième alternative (PosESecL2) a été proposée en augmentant de trois ou de quatre fois la taille de la clé générée par la première alternative (PosESecL1). Une troisième alternative de cryptage robuste et sûre a été également définie. Il s'agit d'OEEA opérant suivant un mode de chiffrement (chiffrement à base d'occurrences) inexploitable par aucune attaque avancée.

Et pour résumer l'efficacité des deux modes de cryptage exploités, le cryptage à base de positions et le cryptage à base d'occurrences, le plus correcte est de dire que le deuxième

mode est plus important que le premier. En effet, ce mode est bâti autour de la notion d'occurrences, chose qui ne peut en aucun cas guider ou être exploitée par un cryptanalyste puisqu'elle ne présente aucune information utile pour aucune attaque possible. Ce qui n'est pas le cas pour le premier mode de cryptage à base de positions où une recherche exhaustive finira, un jour, par retrouver la bonne combinaison de positions d'éléments d'une donnée et ainsi de remettre en cause l'algorithme de cryptage utilisé.

Ainsi c'est les qualités du mode de chiffrement à base d'occurrences qui nous a motivées à le tester à travers d'autres métaheuristiques qu'elles soient à population ou à une seule solution. D'ailleurs le prochain chapitre présente une métaheuristique de colonie de fourmis traitant le problème de cryptage en opérant suivant ce mode de cryptage proposé.

Finalement, il sera utile de rappeler que l'objectif en cryptage est d'assurer la sécurité des données pendant leur durée de validité, donc, toute attaque tardive n'aura aucune importance.

CHAPITRE IV

CRYPTAGE PAR COLONIES DE FOURMIS DES DONNEES TEXTES ET IMAGES

IV.1. Introduction

L'auto-organisation est un phénomène décrit dans plusieurs disciplines, notamment en biologie dans la branche qu'est l'éthologie (étude des comportements des espèces animales dans leurs environnements). Une définition a été proposée par Deneubourg [Dene, 1977] **:** *«l'auto-organisation est un processus dans lequel, un modèle de niveau global émerge uniquement, d'un grand nombre d'interactions entre les composants de bas niveau du système. De plus, les règles spécifiant les interactions entre ces composants sont suivies en utilisant uniquement des informations locales, sans références au modèle global ».*

Autrement dit, l'auto-organisation explique l'émergence d'un comportement collectif macroscopique par des interactions simples au niveau microscopique. L'auto-organisation n'exclut pas la complexité au niveau individuel. Elle suppose simplement qu'à un certain niveau, les individus se comportent comme des entités simples [Monm, 2000].

Depuis longtemps déjà, des comportements auto-organisés ont été découverts dans la nature. Par exemple, dans une colonie de fourmis, on peut observer différentes castes spécialisées dans un certain nombre de tâches : élevage de couvain, recherche de nourriture, construction de nid, etc. Aussi, le nid est construit sans que les insectes soient dirigés, ils répondent à un certain nombre de stimuli provenant de leur environnement.

D'importantes recherches ont eu pour intérêt principal l'étude de ces comportements intelligents afin de savoir comment ces populations interagissaient, accomplissaient des tâches et évoluaient. Ces recherches ont abouti à des modélisations inspirées du principe qu'un groupe d'entités plutôt simples et obéissant à des règles locales de coordination, sont capables d'engendrer des comportements globaux beaucoup plus complexes. Ces modélisations caractérisent ce que l'on dénomme « l'intelligence collective » [Bona, 1994], [Monm, 2000].

Dans ce chapitre, nous nous intéressons à l'algorithme de colonies de fourmis. Nous l'avons étudié, adapté et appliqué au problème de cryptage de données texte ou images. Le vocabulaire employé par cet algorithme est directement calqué sur celui de l'éthologie, nous parlerons donc, de fourmis, de connaissance collective, de phéromone, d'évaporation, etc.

Dans la première section de ce chapitre, nous présentons notre approche utilisée pour la conception du système de chiffrement dont l'objectif principal est l'obtention d'un système de

chiffrement symétrique non déterministe dont la clé secrète est générée pendant l'application de l'algorithme ; donc elle n'est pas connue à l'avance ce qui rend difficile le travail du cryptanalyste. Pour ce faire et étant donné que dans le précédent chapitre le problème de cryptage a été ramené en un problème d'optimisation, il a fallu, tout d'abord, coder le problème d'une manière adéquate permettant, ainsi, d'effectuer les opérations propres aux fourmis tel que : le dépôt, l'incrémentation et l'évaporation de phéromone, etc. Ainsi, le codage adopté est celui validé par la première métaheuristique d'AEs testée qui est le codage à base d'occurrences. Ce dernier permet la concrétisation des déplacements de fourmis par exploitation de la notion de permutation.

Ensuite, nous avons bâti notre algorithme en définissant soigneusement : la fonction d'évaluation cherchant à maximiser la différence entre l'information initiale et celle codée, le mécanisme de sélection adopté pour le déplacement des fourmis, le dépôt et l'incrémentation de phéromone, l'ensemble de la connaissance collective, etc.

IV.2. Motivation

Les algorithmes de colonies de fourmis forment une classe des métaheuristiques récemment proposée pour les problèmes d'optimisation difficile. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés dans les colonies de fourmis. Une colonie de fourmis communiquent indirectement via des modifications dynamiques de leur environnement (les pistes de phéromone) et construisent ainsi une solution à un problème, en s'appuyant sur leur expérience collective. C'est d'ailleurs cette dernière notion que nous cherchons à exploiter à travers l'utilisation de la métaheuristique de colonies de fourmis pour la résolution du problème de cryptage.

Ainsi, les objectifs que nous nous sommes fixés consistent à comprendre et isoler les mécanismes intéressants de cette métaheuristique, utiliser l'approche de chiffrement à base d'occurrences validée dans le précédent chapitre, suivant cette optique et appliquer l'algorithme ainsi conçu au problème de chiffrement de données texte ou images.

D'une manière générale, notre objectif est de développer un algorithme de chiffrement à base de colonies de fourmis par du constat simple que les colonies de fourmis résolvent des problèmes complexes, bien que l'intelligence d'une fourmi soit limitée ce qui est équivalent au fait de dire que, l'intelligence du système entier est plus grande que celle de la simple somme de ses parties.

IV.3. Algorithme proposé

Afin de bien comprendre notre algorithme de chiffrement proposé, AntCrypt, une description complète des différentes étapes de l'algorithme profondément inspirées de la métaheuristique *ACO (Ant Colony Optimisation)*, sera présentée. Des retouches ont été apportées sur l'algorithme de base pour l'adapter au problème étudié qui est celui de cryptage. Ainsi, le schéma de l'algorithme adopté est le suivant :

Algorithme IV.1 *AntCrypt*

Entrées
Codage de la donnée originale
Ensemble de paramètres (*m* : taille de la population, ρ taux d'évaporation)
DEBUT
1) Initialiser la connaissance collective // connaissance collective $\leftarrow \Phi$;
2) Créer la population initiale comportant *m* solutions initiales (*m* : nombre de fourmis) ;
Répéter
3) Construire les solutions;
4) Enrichir la connaissance collective ;
5) Evaluation des solutions de la connaissance collective ;
6) Sélection ;
7) Mise à jour de phéromone ;
Jusqu'à Satisfaction du critère d'arrêt
Retourner la solution trouvée ;
FIN

IV.3.1. Codage adopté

Pour prouver encore une fois l'efficacité de l'approche proposée et précédemment présentée exploitant un chiffrement bâti autour de la notion de nombres d'occurrences d'éléments de la donnée à chiffrer, le codage adopté par AntCrypt sera le même que celui d'OEEA.

La figure IV.1 rappelle le codage de données à utiliser.

$O(e_1)$	$O(e_2)$	.....	$O(e_i)$	.....	$O(e_{l-1})$	$O(e_l)$

Figure IV.1. *Codage d'une solution sous AntCrypt.*

Où : $O(e_i)$ est le nombre d'occurrences de l'élément e_i dans la donnée, l représente le nombre de valeurs possibles d'éléments (1393 valeurs possibles Unicode des caractères affichables pour les données texte et 256 valeurs possibles pour chacune des composantes R, G et B pour les données images).

Ainsi, notre algorithme AntCrypt cherche à changer itérativement la répartition des nombres d'occurrences $O(e_i)$ sur les différents éléments d'une certaine donnée avec génération aléatoire de positions dans la solution afin de créer le maximum de désordre dans leurs positions. Cela donne, aussi, l'avantage de produire des éléments (caractères/pixels) inexistants dans la donnée initiale.

IV.3.2. Création de la solution initiale

Initialement la connaissance collective (le savoir partagé) est vide. Dans cette étape on crée la population initiale comportant *m* solutions (*m* : le nombre de fourmis).

Nous désignons par *DonneeInitiale* le codage de la donnée originale. Donc, il est représenté par un vecteur dont les éléments contiennent les nombres d'occurrences des 1393 caractères (respectivement des 256 valeurs possibles de chacune des composantes R, V et B codant les pixels) dans le texte (respectivement dans l'image). Ces éléments sont notés :

$$O(e_1), O(e_2), ..., O(e_n) / n = \begin{cases} 1393 : Texte \\ 768 : \text{Im}age \end{cases}$$

Nous générons ensuite *m* fourmis susceptibles de créer, chacune, une nouvelle solution. Pour ce faire, chacune des fourmis est positionnée aléatoirement sur un élément de *DonneeInitiale*. Nous avons choisi de noter les différentes positions courantes par *PosCourante*. Ensuite, chaque fourmi va se déplacer un certain nombre de fois vers d'autres éléments (d'autres nœuds) notés chacun *PosSuivante* par application de la roulette aléatoire, et elle permute lors de chaque déplacement le nombre d'occurrence de l'élément « *DonneeInitiale[PosCourante]* » avec celui de l'élément « *DonneeInitiale[PosSuivante]* ». Une fois une fourmi s'arrête de se déplacer, nous obtenons une solution qui sera ajoutée à l'ensemble de connaissance collective. Alors à la fin de cette étape, la connaissance collective sera augmentée de *m* nouvelles solutions (chaque solution est obtenue par une fourmi).

IV.3.3. Construction de solutions

Après l'étape d'initialisation et dans le but de construire de nouvelles solutions sur une itération donnée de l'algorithme, les fourmis vont choisir à partir de la connaissance collective cette fois-ci leurs points de départ pour construire de nouvelles solutions. Ces points représentent les solutions ayant les plus grandes quantités de phéromone, c.-à-d. celles qui perturbent le plus la donnée initiale par rapport à une version cryptée représentée par la solution choisie. Donc le mécanisme permettant la construction d'une solution est le suivant :

1) Choisir les *m* meilleures solutions de la connaissance collective ;
Pour i = 1 : *m* ***faire***
    2) Choisir une solution *Sol* parmi les *m* solutions choisies en (1) ;
    Pour j = 1 : NbrMaxDép ***faire***
        3) Positionner aléatoirement la fourmi sur un élément *Sol[PosCourante]*;
        4) Déplacer la fourmi suivant la méthode de roulette aléatoire vers une autre position *PosSuivante* ;
        5) Permuter les deux éléments *Sol[PosCourante]* et *Sol[PosSuivante]* ;
    Fin pour
    6) Ajouter *Sol*, la solution qui vient d'être construite, à la connaissance collective.
Fin pour

IV.3.4. Evaluation

La fonction d'évaluation (ou d'adaptation) quantifie la qualité des solutions calculées. Celle que nous avons définie pour associer des valeurs d'adaptation à nos solutions calculées est la suivante :

$$F(Sol_j) = \sum_{i=1}^{n} \left| O_j(e_i) - O_0(e_i) \right| \quad \text{(IV.1)}$$

Tels que :

Sol_j : Solution j,

n : Nombre des éléments du vecteur *DonneeInitiale*,

$O_j(e_i)$: Nombre d'occurrences de l'élément (caractère / pixel) i dans la solution j,

$O_0(e_i)$: Nombre de répétitions de l'élément (caractère / pixel) i dans la donnée initiale.

IV.3.5. Sélection

Le rôle de la sélection est de distinguer les solutions sur la base de leur qualité, en particulier, pour permettre aux meilleures solutions d'une génération donnée d'être copiées dans la génération suivante.

Dans notre cas, lors du passage d'une génération g à la génération qui suit (g+1), nous favorisons les m meilleures solutions parmi celles de la connaissance collective globale y compris celles construites durant la génération g.

IV.3.6. Manipulation de phéromone

IV.3.6.1. Incrémentation de phéromone

Suivant la fonction F donnée en (IV.1), nous devons choisir la quantité de phéromone déposée pour chaque solution. Le mode de dépôt sélectionné peut fortement modifier le mode de convergence de l'algorithme. On pourrait choisir de déposer la même quantité à chaque solution. Cependant, on peut aussi décider de donner une puissance plus forte aux phéromones déposées sur les solutions améliorant l'altitude. C'est cette seconde solution qui a été appliquée dans notre algorithme.

Pour chaque solution *Sol*, nous calculons leur efficacité selon la formule (IV.1). Si cette solution existe déjà dans la connaissance collective on incrémente leur quantité de phéromone comme suit sans la réinsérer :

$$Phéromone(Sol_i) = Phéromone(Sol_i) + \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.2)}$$

Tel que EffMax est l'efficacité de l'une des meilleures solutions calculées de manière déterministe en permutant dans un ordre choisi le $i^{ème}$ élément avec le meilleur des l-i éléments restants (en maximisant la différence entre les deux éléments en question).

Sinon, si la solution n'existe pas, elle sera ajoutée à la connaissance collective en lui attribuant une quantité de phéromone selon la formule suivante :

$$Phéromone(Sol_i) = \left(\frac{F(Sol_i) \times 100}{EffMax} \right) \quad \text{(IV.3)}$$

IV.3.6.2. Évaporation de phéromone

Comme dans la nature, les phéromones sont des substances chimiques qui s'évaporent au fil du temps. Donc, il sera nécessaire que l'algorithme imitant ce coté de la nature représente ce phénomène afin d'éviter aux fourmis de converger vers un maximum local.

$$Phéromone(Sol_i) = Phéromone(Sol_i) \times (1 - \rho) \qquad \text{(IV.4)}$$

Où ρ est le taux d'évaporation.

L'évaporation de phéromone de toutes les solutions après une période de temps permettra de redonner de l'intérêt à d'autres solutions qui peuvent nous amener à la meilleure solution. Le taux d'évaporation ρ doit être bien choisi, puisque s'il est très faible, les phéromones s'accumulent et atteignent la borne max et nous risquons alors de s'enfermer dans un maximum local, et s'il est trop grand, les phéromones atteignent rapidement la borne min et beaucoup de solutions n'auront pas la chance d'être choisies par les fourmis.

Dans notre application, ce taux doit aussi dépendre des valeurs des éléments du codage. Si la différence entre deux éléments est grande, c.à.d. la fonction F va prendre de grandes valeurs correspondantes à de grandes quantités de phéromones, donc le taux d'évaporation peut prendre une valeur importante. De même, il dépend du nombre de fourmis travaillant sur une même génération : beaucoup de fourmis dans une génération augmentent la possibilité d'avoir deux solutions identiques, c.-à-d. la phéromone sera incrémentée plusieurs fois de suite, alors le taux d'évaporation sera important.

Ainsi et après un certain nombre de générations fixé expérimentalement, nous avons varié un pourcentage d'évaporation de 0% jusqu'à 0.5% en s'inspirant, au début, de l'application de l'algorithme ACO pour la résolution du problème de TSP [Barg, 2005]. Ce choix a été, ensuite, validé par expérimentation.

IV.3.7. Critère d'arrêt

Après un certain nombre de générations fixé expérimentalement, l'algorithme converge vers la meilleure solution trouvée. Reste de générer ou de reconstituer la donnée cryptée à partir du codage de la solution proposée. Ceci correspond à l'opération inverse de l'opération de codage. Il s'agit d'une opération de décodage qui procède comme suit : à partir du codage de la solution proposée qui est un vecteur de nombres d'occurrences des éléments de la donnée chiffrée, pour chaque élément *i* du vecteur *Sol*, on génère *Sol[i]* positions aléatoires du caractère/pixel correspondant à l'élément *i* dans la donnée cryptée.

IV.3.8. Déchiffrement

Nous arrivons maintenant au processus inverse qui permet de rendre la donnée à nouveau intelligible sans aucune perte d'information ; il s'agit du processus de déchiffrement.

Notre algorithme proposé est un algorithme symétrique, donc la clé générée doit être maintenue secrète. Cette clé s'obtient au fur et à mesure du calcul de l'information chiffrée au fil des générations. Sa valeur finale correspond aux permutations des positions des nombres d'occurrences des éléments de la donnée chiffrée pour obtenir celles des nombres d'occurrences des éléments de la donnée en clair. Le processus de déchiffrement consiste donc à replacer les éléments dans leurs positions initiales suite à l'introduction de la bonne clé.

IV.3.9. Réglage des paramètres et résultats

Cette section est consacrée d'une part, à l'optimisation ou la fixation du jeu paramétrique et d'autre part, à l'application de l'algorithme proposé *AntCrypt* sur les mêmes données de test utilisées pour l'évaluation des AEs proposés et présentés dans le précédent chapitre. Ils s'agissent des données faisant l'objet de la figure III.1. Dans un premier temps, nous allons exposer notre stratégie pour établir et fixer le choix paramétrique de notre algorithme. Puis nous allons présenter les résultats d'application d'*AntCrypt* sur les données modèles (textes et images), à savoir la donnée chiffrée, la clé générée, la quantité de phéromone, l'efficacité, le temps de chiffrement et de déchiffrement.

IV.3.9.1. Réglage des paramètres

Les figures IV.2 et IV.3 présentent les résultats des différents tests effectués en termes d'efficacité et de temps de chiffrement. Ces valeurs représentent la moyenne de dix (10) exécutions successives appliquées sur l'image de test Lena pour différentes valeurs de nombre de générations et de fourmis.

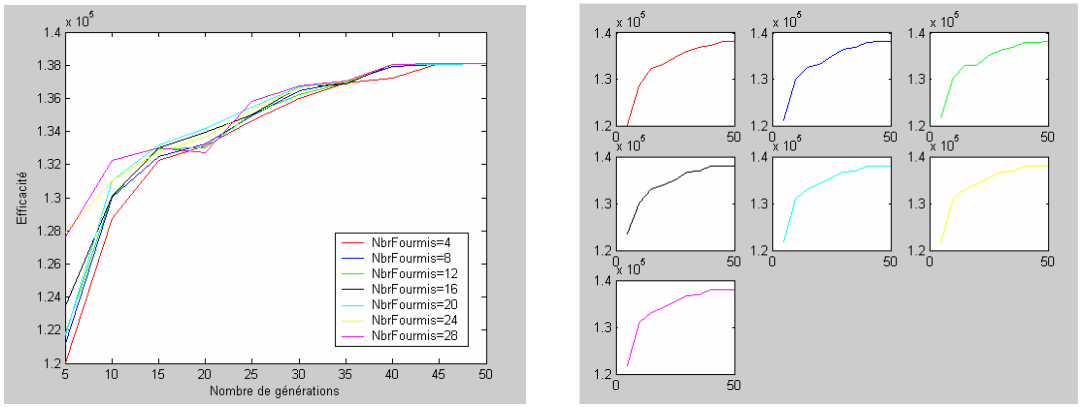


Figure IV.2. *Influence du nombre de générations et du nombre de fourmis sur l'efficacité.*

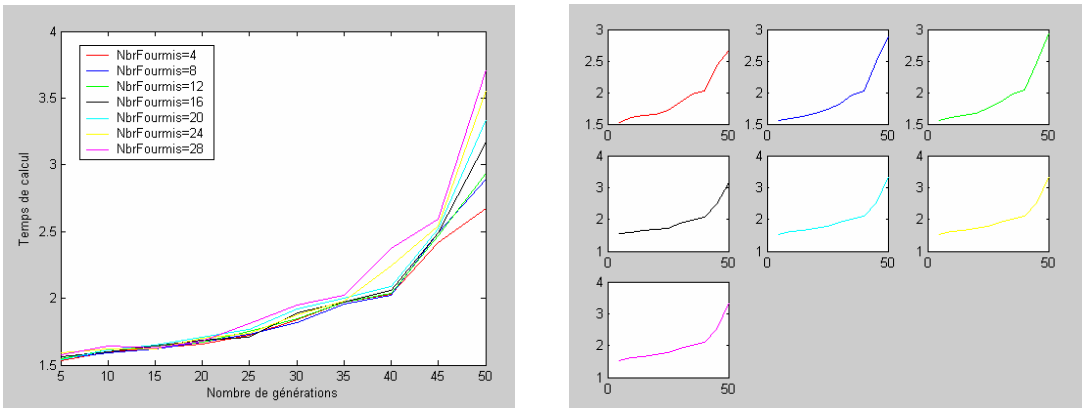


Figure IV.3. *Influence du nombre de générations et du nombre de fourmis sur le temps de calcul.*

D'après les résultats résumés à travers les figures IV.2 et IV.3, nous constatons que la majorité des résultats des tests ont des valeurs d'efficacité assez proches. La meilleure valeur d'efficacité est obtenue au bout de 50 générations à 20 fourmis (Efficacité = 138077) mais avec un temps de chiffrement assez important par rapport à l'ensemble de test (3,17 s). Cependant, le meilleur temps de chiffrement est obtenu pour le test qui a été déroulé sur 5 générations et 4 fourmis (1,53 s) mais malheureusement avec la plus faible valeur d'efficacité (Efficacité = 120009).

Les tests déroulés sur 45 générations avec 4, 20, 24 ou 28 fourmis, et 50 générations avec 4, 8, 12, 20, 24 ou 28 fourmis ont donné de très bonnes valeurs d'efficacité (138073, 138070, 138073 et 138074, 138075, 138074, 138076, 138073, 138072, 138075 respectivement) mais après de longs temps de calcul (2.42 s, 2.52 s, 2.54 s, 2.59 s et 2.67 s, 2.94 s, 3.34 s, 3.56 s et 3.71 s).

D'autres tests déroulés sur 5 et 10 générations ont montré, cette fois-ci, de bons temps de calcul (compris entre 1.53 s et 1.64 s) mais de mauvaises valeurs d'efficacité (comprises entre 120009 et 132263).

Donc, notre objectif sera de réaliser un compromis entre la valeur d'efficacité et le temps de calcul. En effet, le test déroulé sur 40 générations avec 12 fourmis semble le plus satisfaisant. Il permet d'atteindre une bonne valeur d'efficacité (Efficacité = 138063) en un temps de calcul très raisonnable (2.04 s). Ca sera le paramétrage à adopter pour *AntCrypt*.

IV.3.9.2. Résultats

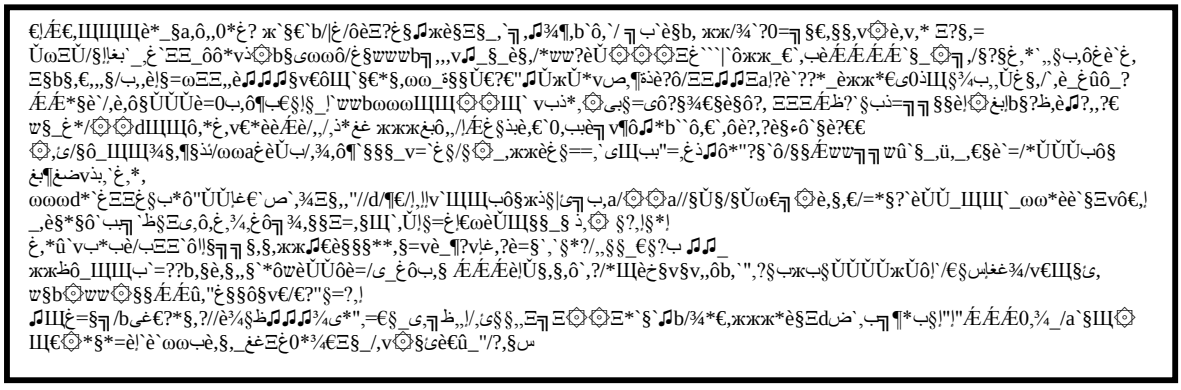
Ci-dessous, nous présentons les résultats d'application d'*AntCrypt* sur les mêmes données modèles utilisées pour tester nos trois algorithmes présentés dans le précédent chapitre : Texte1, Texte2, image Lena et image Logo suivant le paramétrage fixé dans la section précédente (résumé à travers le tableau IV.1). Toutefois, et pour démontrer l'aspect non déterministe de notre approche de chiffrement par exploitation de métaheuristiques, dont l'ACO fera l'objet ici, nous présentons, cette fois-ci, pour chaque donnée originale deux versions chiffrées en deux instances différentes accompagnées des clés de chiffrement générées dans chaque cas.

	Valeurs des paramètres d'*AntCrypt*
Nombre de fourmis	12
Nombre de générations	40

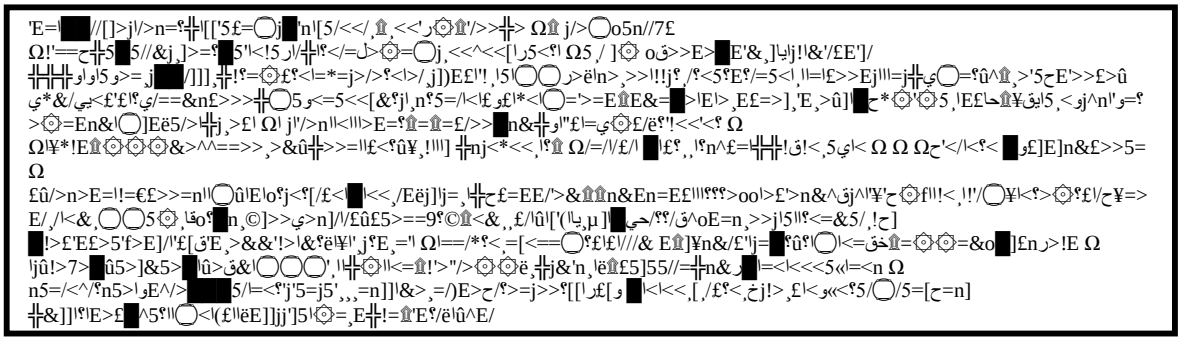
Tableau IV.1. *Valeurs adoptées pour les paramètres de AntCrypt.*

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure IV.4. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets) :

124	398	407	770	921	266	896	19	794	90	918	245
843	914	614	547	809	904	560	80	375	908	879	886
901	816	830	543	573	900	156	196	98	254	906	314
230	443	917	497	903	418	288	907	289	680	147	861
820	642	902	259	916	840	30	898	905	102	822	635
317	161	920	465	292	458	546	91	911	246	910	833
111	912	155	193	640	919	923	643	357	681	121	725
915	892	581	899	922	216	909	913	789	549	179	578
28	552	3	650	1070	379	850	996	934	933	1104	285
459	997	1124	1242	1026	897	1171	1075	8	521	64	281
1017	955	985	872	262	485	774	16	85	709	279	97
628	120	374	87	286	1112	277	862	518	47	255	1097
1203	45	163	60	599	12	461	5	848	1240	92	1032

331	174	888	83	1073	20	676	342	494	280	422	487
607	1020	958	61	1154	601	95	1100	112	386	1238	1196
1185	1119	393	669	1	756	1172	1012	1102	274	394	363
237	755	267	6	219	389	893	187	1225	437	175	683
834	659	1215	1085	1138	819	447	688	436	764	1158	117
206	1024	675	695	264	319	345	76	127	1183	484	1006
1191	540	1205	210	384	689	655	991	71	928	1130	416
1129	324	761	1134	1089	1029	449	1200	704	231	1013	466
474	1166	813	302	506	1050	490	1204	1092	382	194	744
1066	682	164	72	630	972	15	1213	738	309	1199	531
1149	1084	1202	971	535	395	1031	731	988	737	41	1188
1142	584	336	839	113	818	1245	306	723	657	620	403
252	858	994	423	1036	563	1217	1137	135	108	1133	1088
1160	810	829	946	1058	1068	952	944	751	673	1110	79
1232	835	733	557	1193	982	1179	298	1052	1101	198	1144
225	232	949	400	229	981	545	846	954	432	778	411
634	1161	1042	457	18	250	68	1123	866	462	1243	1051
1120	768	960	990	1189	1182	491	827	185	1145	1148	714
1115	428	574	995	1067	653	1091	978	27	415	132	940
145	409	665	1234	956	622	727	1227	786	402	1009	859
1236	1224	870	1039	387	239	1028	565	1043	594	176	856
787	424	323	588	1033	715	945	1195	359	1162	1165	793
791	930	684	299	716	1076	605	101	1241	149	826	412
1181	480	364	1187	1004	14	1226	579	438	1222	542	613
125	664	1176	197	202	1208	593	520	183	1207	248	604
434	1139	1127	1014	211	891	169	747	968	55	1190	339
1178	625	758	698	330	1038	446	425	611	668	646	1170
1174	144	166	283	1244	857	1231	1131	366	48	31	34
1107	52	814	315	24	414	983	1135	860	11	452	937
539	337	812	935	639	507	863	378	637	301	1055	1219
1077	1198	440	943	451	658	478	361	392	82	103	586
693	881	1210	652	1235	37	1001	970	1156	1175	808	184
278	662	799	341	1041	244	736	1030	109	730	984	195
1019	678	728	420	294	1212	796	1218	303	1192	627	645
1079	1109	1046	481	272	488	1116	1096	222	238	21	349
1034	2	1128	1011	130	157	445	167	705	1087	1201	33
470	489	10	598	453	128	39	1061	817	1209	1216	741
926	1184	1008	942	524	969	372	308	118	228	473	815
312	623	1122	516	1078	962	1143	929	1233	989	316	602
941	1125	1223	648	1072	1136	58	1062	224	931	571	1173
987	649	158	1040	612	998	233	864	241	140	9	621
177	88	1047	321	57	1000	1037	1117	340	1151	961	1221
595	973	138	152	959	483	213	479	207	477	191	287
740	522	977	654	284	227	332	141	234	105	842	305
1080	957	572	355	310	1140	660	258	78	322	831	1054
201	1229	467	1045	273	353	629	441	618	63	1074	849
527	924	43	173	710	295	948	708	1057	876	606	36
967	1025	1081	1147	1132	868	979	1082	966	844	1237	1010
1239	845	772	51	391	1168	855	865	823	65	62	874
275	276	159	885	182	1105	1118	1002	603	592	999	759
1035	762	1069	1083	1007	1005	932	1094	824	745	110	828
154	677	638	703	851	204	1180	806	208	760	884	771
610	1230	25	1211	439	925	1022	765	1155	670	672	519
469	356	548	679	1056	1015	236	1167	964	803	1106	218
1194	591	663	587	260	50	69	1141	953	1071	975	701
212	396	550	307	1063	203	811	743	510	84	333	1153
739	566	807	1159	1018	162	1157	385	1023	35	1114	22
644	1108	261	1093	711	963	296	328	706	122	358	597
938	965	504	1197	249	788	1098	790	325	1214	1228	401
1027	525	1086	951	624	529	1099	992	1150	136	667	380
1126	1111	199	1048	1206	580	404	697	251	719	475	40
585	1044	1095	1053	686	350	575	1163	1003	1060	1064	223
950	146	986	1177	752	304	726	123	853	742	168	492
454	766	804	1021	533	247	46	939	42	1121	221	555
226	980	327	17	180	974	798	927	209	1152	692	554
370	126	151	129	596	377	800	1169	1146	408	463	417
569	1103	615	498	326	59	456	1186	1049	137	139	291
427	632	685	1113	433	1220	153	1164	722	354	189	976
523	100	763	509	1016	444	399	369	1065	936	825	220
360	265	947	1059	749	568	1090	838	852	631	365	993

329	268	651	1330	390	500	1331	1332	460	320	468	729
1333	1274	583	617	753	564	56	782	1306	661	1334	263
165	1335	1318	795	713	371	367	1336	134	1305	271	748
1337	81	532	1299	343	486	1338	1339	472	1264	1302	413
1340	362	769	801	633	1273	517	534	867	94	335	732
1247	1290	1316	1260	89	781	718	1341	450	1323	551	1342
1343	1344	894	1292	687	515	1345	608	1346	1347	1246	351
368	767	1248	570	430	1322	735	235	1348	1349	1284	1350
1288	1278	802	847	257	499	746	805	1267	114	1295	514
1351	188	785	558	1317	1352	784	70	537	832	1353	448
181	1354	297	99	44	442	217	609	501	410	1355	837
1356	66	1314	313	38	1357	777	541	559	1358	1320	426
170	1312	1319	1279	186	32	882	171	429	776	1359	1360
880	720	671	1275	406	215	431	1253	347	792	717	1361
200	1362	1313	1363	54	754	1251	476	511	1364	435	1325
1365	1291	338	1257	1315	699	1282	641	712	381	96	1327
482	883	1280	421	1366	513	1367	26	1368	1250	779	1276
1326	242	143	1307	300	1249	1369	508	23	721	690	1298
773	895	887	1270	875	53	576	67	589	455	890	636
626	1328	405	536	647	582	119	656	696	1370	282	290
334	619	1371	1372	512	503	1373	1374	854	544	877	1375
205	493	106	1308	783	77	577	293	1376	1254	344	1252
148	526	256	1377	464	616	214	878	1294	1303	1321	1378
1297	1379	1272	1311	1300	1266	240	590	1380	530	243	1287
567	1258	1293	160	495	707	150	1381	666	702	873	505
528	674	1382	562	694	86	889	397	1277	93	1296	1285
1256	104	131	270	1383	311	49	190	1281	841	388	734
1283	1265	346	775	115	600	1263	1255	1384	352	1310	172
1259	1309	561	869	13	691	142	133	871	373	821	836
269	502	75	1385	253	1386	116	318	1301	376	1387	556
348	74	178	1268	724	107	1324	553	7	1388	1286	192
1389	29	73	780	1329	750	538	797	471	1262	1390	419
757	1391	1261	1392	1304	1289	496	4	700	1269	383	1271
1393											

Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets) :

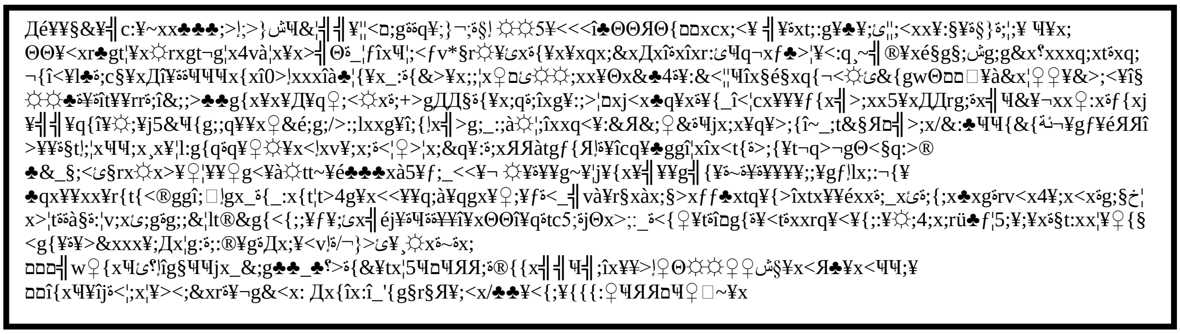
80	34	87	94	51	64	91	46	18	32	55	81
44	2	63	86	54	72	45	16	88	84	20	75
82	97	39	13	6	36	67	95	12	53	17	92
24	98	89	14	74	60	25	41	40	76	10	57
66	93	5	11	9	70	3	31	73	61	42	43
23	27	59	28	78	37	52	69	7	71	77	8
99	50	85	96	30	21	29	22	58	19	90	68
26	47	79	15	35	48	38	878	603	723	164	621
401	997	320	274	4	126	364	628	908	150	286	287
600	815	170	482	424	159	307	220	139	213	596	967
217	133	405	644	511	861	909	334	802	964	597	691
449	942	485	721	538	945	862	870	495	513	505	957
623	883	229	506	955	419	767	751	917	56	143	237
831	972	799	62	324	224	479	750	864	255	771	330
378	316	474	564	161	403	172	748	461	223	978	778
166	318	904	540	985	167	741	202	467	569	49	434
793	283	765	568	248	102	250	653	816	359	297	602
232	249	207	196	365	190	754	947	455	490	338	931
788	454	698	649	774	308	588	702	708	631	33	444
557	463	503	913	886	1	715	116	550	516	189	849
244	685	458	584	285	764	975	418	589	877	921	487
309	632	144	701	850	317	932	563	239	643	610	192
348	574	533	732	993	212	809	299	417	790	501	464
922	567	867	331	650	231	466	722	895	666	168	891
529	541	558	762	279	730	635	824	486	177	313	520
581	560	607	140	977	719	859	339	638	599	380	491
662	777	532	890	758	169	745	382	83	580	900	965
534	987	968	395	104	892	107	826	225	670	222	200
893	387	992	292	545	499	961	353	590	919	869	291
772	575	191	160	605	305	180	811	142	410	832	236
450	389	868	314	659	699	163	247	507	335	481	293
110	982	673	214	634	363	675	842	65	995	343	576
814	546	175	488	425	280	910	677	680	265	718	438
497	716	875	414	366	692	710	976	344	548	970	376

851	421	781	837	927	792	388	874	860	836	798	570
627	743	912	937	204	310	803	203	509	686	306	665
131	776	728	808	682	938	657	197	843	735	779	549
739	593	179	640	846	185	717	370	753	379	178	360
857	465	579	565	899	756	158	452	606	737	132	326
629	729	498	784	295	429	221	648	925	375	500	385
408	182	235	616	431	654	768	171	478	346	694	300
609	994	797	125	357	926	766	863	103	257	145	789
924	420	696	321	528	329	646	476	981	148	402	123
227	818	374	396	954	833	749	121	233	198	935	795
462	720	105	153	897	724	705	416	604	230	536	655
539	866	228	697	246	773	998	526	951	426	352	974
591	988	852	647	573	806	642	127	427	100	619	404
713	459	782	328	783	827	433	496	551	208	349	880
807	369	146	371	118	523	181	521	928	613	277	439
939	661	645	946	510	393	484	633	578	601	106	906
787	887	819	278	907	902	553	276	151	770	272	134
394	681	165	358	432	668	810	556	399	129	820	350
136	117	205	812	622	262	817	652	984	109	260	397
916	448	618	559	898	530	742	780	973	689	885	264
847	245	514	687	630	747	524	986	493	303	259	477
270	267	195	442	958	991	215	547	347	269	914	162
667	114	571	537	733	663	124	865	377	586	583	587
761	184	936	135	744	940	368	296	684	941	712	592
137	113	206	706	412	933	241	594	271	688	822	543
561	980	216	332	844	660	535	304	996	949	894	731
423	341	755	734	284	356	918	183	201	504	830	392
390	585	956	457	446	440	492	775	174	651	582	841
456	615	199	149	554	522	238	362	266	813	796	111
959	989	614	876	953	608	119	409	282	544	428	834
639	351	354	664	637	855	301	873	517	519	625	896
963	725	400	736	355	209	595	290	671	508	835	658
381	256	430	979	345	871	786	901	176	714	839	288
437	120	319	243	840	598	983	251	157	226	752	342
700	669	821	435	856	470	273	407	315	726	234	888
406	727	252	704	340	473	443	929	542	823	740	337
785	641	384	693	460	872	218	445	469	155	193	853
386	828	138	298	999	854	881	943	483	101	759	769
147	930	611	678	112	672	624	709	311	141	577	518
791	422	626	480	336	612	950	905	948	617	327	920
471	128	829	562	695	884	240	858	281	122	254	323
367	707	683	453	794	512	711	889	944	475	263	302
372	186	656	294	911	531	845	969	253	738	801	805
966	489	472	383	289	990	210	258	441	879	391	451
525	322	188	763	173	312	413	261	154	962	515	156
800	971	411	882	219	436	398	527	415	555	333	760
674	757	447	620	572	268	952	242	115	502	494	187
194	152	552	804	703	468	108	903	838	636	130	275
960	325	848	676	923	825	679	690	361	211	746	566
934	915	373	1097	1243	1324	1351	1271	1352	1040	1080	1241
1277	1136	1353	1089	1051	1354	1355	1039	1077	1342	1258	1320
1037	1064	1202	1242	1312	1281	1123	1054	1170	1267	1285	1222
1340	1349	1129	1111	1296	1174	1252	1345	1356	1166	1357	1115
1275	1328	1250	1128	1263	1171	1098	1284	1033	1231	1276	1005
1058	1313	1311	1059	1124	1232	1157	1013	1100	1132	1019	1099
1103	1023	1112	1225	1240	1304	1194	1215	1358	1247	1198	1334
1335	1122	1018	1210	1164	1341	1156	1127	1106	1006	1300	1323
1114	1359	1245	1113	1066	1154	1150	1036	1344	1289	1360	1238
1298	1034	1035	1093	1048	1008	1318	1253	1309	1028	1331	1361
1016	1195	1286	1362	1363	1050	1090	1038	1137	1119	1287	1079
1163	1182	1151	1125	1043	1303	1025	1325	1133	1042	1364	1002
1234	1075	1347	1088	1365	1212	1117	1230	1301	1294	1071	1214
1200	1307	1049	1264	1190	1060	1147	1299	1332	1160	1165	1366
1109	1180	1343	1305	1315	1065	1187	1055	1052	1191	1367	1110
1196	1368	1074	1205	1257	1262	1168	1336	1108	1308	1045	1369
1330	1278	1239	1134	1221	1149	1235	1104	1209	1370	1053	1265
1282	1107	1094	1233	1069	1153	1213	1244	1327	1266	1146	1229
1091	1141	1105	1346	1226	1041	1070	1186	1142	1056	1302	1063
1199	1321	1162	1193	1086	1101	1270	1083	1310	1017	1192	1177
1179	1095	1248	1184	1219	1121	1022	1273	1172	1371	1046	1372

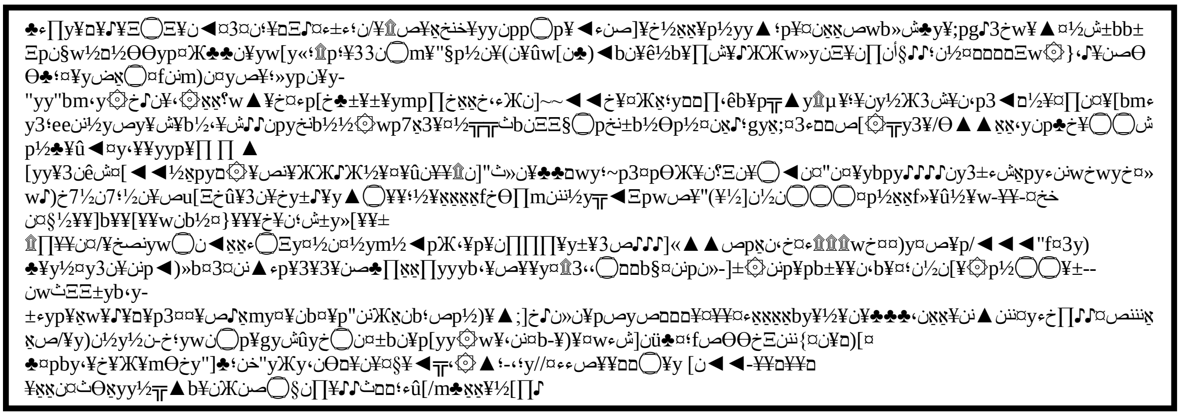
1044	1178	1096	1208	1224	1183	1003	1084	1012	1131	1130	1348
1062	1316	1159	1188	1021	1373	1009	1135	1374	1375	1203	1211
1228	1010	1087	1268	1376	1333	1081	1377	1322	1220	1145	1072
1057	1288	1061	1047	1249	1255	1120	1030	1306	1261	1189	1246
1314	1144	1378	1254	1379	1126	1295	1138	1176	1319	1152	1337
1216	1201	1236	1380	1293	1076	1185	1291	1158	1143	1381	1029
1297	1067	1259	1382	1140	1155	1068	1269	1031	1383	1384	1118
1339	1260	1290	1102	1385	1386	1082	1280	1167	1000	1078	1085
1073	1387	1197	1272	1007	1181	1227	1011	1001	1161	1206	1251
1116	1204	1388	1173	1032	1350	1024	1389	1207	1292	1217	1326
1237	1390	1223	1218	1148	1391	1392	1279	1274	1026	1027	1256
1092	1014	1015	1393	1317	1139	1338	1329	1169	1004	1175	1283
1020											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس ومنظم يدين لعالم التشفيرالذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز؛ لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)



(b)



(c)

Figure IV.5. (a) *Donnée originale Texte2,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

700	340	911	855	661	6	451	910	17	3	151	267
937	143	936	914	475	880	745	874	939	897	155	935
281	730	209	938	173	638	777	191	329	930	384	421
673	131	485	462	512	259	736	644	457	728	932	593
933	65	920	924	926	918	349	196	232	309	553	782
917	369	919	927	455	878	255	921	913	929	156	931
912	759	400	922	616	915	594	691	243	934	592	916
819	361	128	923	486	24	925	159	748	301	928	140
411	379	1093	16	599	1169	1108	1079	1040	985	1021	141
799	1006	1067	1107	82	572	1014	325	977	498	9	943
974	670	394	805	270	49	178	658	808	604	392	1229
642	456	1058	80	1184	78	669	472	1045	1214	445	1011
195	492	242	952	389	1069	814	839	1060	591	983	895
285	248	67	947	1194	460	380	38	46	584	482	430
631	1236	478	1012	1004	297	570	1240	228	332	390	1127
1153	40	992	991	1151	625	44	1106	47	1189	99	823
986	706	395	85	559	205	28	641	887	197	319	979
1094	554	882	1066	822	1166	531	564	949	305	398	654
694	544	200	160	801	1017	189	1031	1083	1143	476	1196
1002	198	894	348	1052	619	69	214	1192	1099	120	1134
318	300	534	73	862	104	425	328	1092	1141	677	296
692	1212	466	612	1190	144	960	686	1188	53	404	668
683	381	106	672	523	1145	688	42	1053	342	102	779
292	757	1055	1183	852	514	34	890	702	525	1233	863
870	79	444	858	1241	851	904	1250	948	346	224	31
807	754	528	942	794	1140	511	1101	347	295	1198	289
436	408	587	308	269	403	247	1074	25	1148	27	264
194	489	1160	163	1056	1225	298	322	965	1202	563	237
1035	1179	941	56	1051	1057	1064	434	978	1015	532	413
602	1049	71	772	162	1123	483	1142	231	1152	844	1078
1235	1159	356	418	1047	1082	737	273	1137	676	1144	263
372	1039	530	1098	440	284	1191	1081	94	1205	29	783
1216	603	630	859	877	598	473	4	274	944	674	681
709	1154	565	1008	837	662	549	671	1003	215	710	1228
1243	758	1016	962	1238	175	1217	1158	518	57	1172	628
764	876	1223	606	7	253	657	968	1245	766	999	542
324	775	399	738	989	1097	1009	515	238	406	770	245
993	499	1167	30	217	230	1077	299	787	74	537	1029
784	802	645	76	1147	1197	415	452	1126	45	517	629
1227	1000	940	569	632	618	164	172	101	18	88	216
1135	950	397	377	879	809	1226	1176	527	227	1203	431
833	276	970	87	1027	256	480	843	409	637	286	1013
1248	142	725	605	743	1114	567	643	791	640	576	1090
1244	150	689	1018	650	589	311	516	995	1178	519	1175
611	828	621	857	1122	1020	1065	294	474	1230	959	376
869	501	898	370	545	1117	816	1071	1110	43	526	623
115	836	655	680	26	494	1195	447	1023	385	513	548
969	161	250	1185	1034	110	111	860	15	561	793	1170
961	820	133	432	1116	97	114	279	93	193	495	422
1246	1059	997	112	212	1210	1130	33	219	60	469	698
1231	174	812	479	763	1207	1218	277	50	450	416	41
954	582	116	488	127	1128	552	697	1209	1080	306	490
1085	1129	539	981	953	337	449	211	726	1061	1124	510
715	1091	861	83	1007	827	477	330	800	849	546	8
747	419	10	186	454	500	66	583	899	357	401	1001
971	659	856	226	1234	240	233	91	761	343	1046	1146
595	107	1186	1237	1103	1139	1076	1048	647	579	722	353
608	386	660	1115	596	1042	610	636	405	718	290	529
1138	234	433	1112	1164	964	945	742	967	773	20	77
1070	1221	1072	804	1224	663	1161	721	1120	1118	458	797
1193	707	1119	326	453	35	505	1030	533	126	1041	64
769	753	14	63	1024	1100	387	149	744	239	1200	703
1037	504	522	32	956	875	536	723	998	345	1171	435
1068	1155	366	832	1213	491	1028	1173	829	429	23	374
1181	976	11	973	393	1187	988	1232	135	1102	169	493
278	664	134	90	821	885	1163	762	741	627	108	892
1199	996	367	958	982	1165	417	1174	95	1150	557	1033
740	333	891	382	129	972	1247	1109	1121	266	204	1206

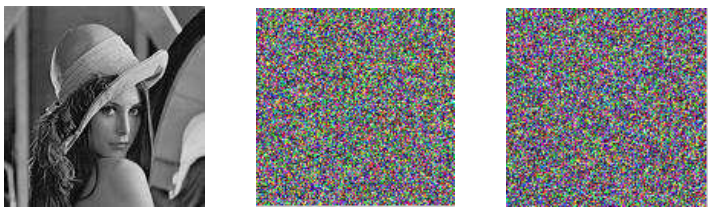
1010	388	187	966	1204	955	987	70	790	1239	503	352
957	262	1084	272	59	866	414	1168	1219	1201	1222	907
1111	317	963	1113	1026	497	487	818	810	109	442	439
190	780	423	896	130	363	1086	798	617	1050	1087	994
1105	208	1211	580	796	89	291	68	1025	251	1182	1005
682	1104	378	980	538	646	118	323	699	1157	359	1136
236	1156	1131	1075	1043	1149	853	96	229	351	179	1220
180	138	145	424	177	1019	1032	560	975	136	1132	1125
261	774	685	568	886	675	315	202	158	1036	168	1133
713	601	1215	600	176	1054	1208	817	865	678	739	864
789	1095	1162	509	1022	614	327	1249	946	607	806	760
185	1062	221	667	902	566	316	635	665	39	813	717
786	951	834	624	438	1088	58	1177	984	1044	62	719
1251	900	1096	100	1073	166	990	695	1180	463	2	334
124	1242	1038	123	148	889	615	121	1089	651	407	481
749	1063	1271	470	86	597	465	1332	1307	260	1342	1275
881	368	371	48	1343	1301	1270	1344	838	1313	1345	666
1346	132	265	871	302	840	84	502	842	1347	222	765
573	252	1348	1322	732	246	1349	835	1334	13	1319	1339
464	1350	778	771	1351	355	268	590	845	1311	210	223
1289	577	184	1282	1352	735	1269	1293	1353	1278	459	1263
241	207	1274	1354	652	1325	1298	446	12	588	1268	873
884	1355	622	1252	125	344	551	213	304	1356	1265	1357
1296	609	1330	908	693	98	5	868	555	1320	225	428
687	1331	383	541	1308	81	188	756	1267	558	154	1272
1358	1359	341	1258	52	847	467	1360	776	724	1361	22
1279	103	280	36	220	1362	795	634	578	461	571	484
181	303	321	690	1363	734	905	1285	574	19	733	412
396	313	402	1364	1365	331	139	550	51	1366	1281	365
335	746	1367	586	257	336	613	287	1253	803	883	792
653	850	72	585	310	831	1329	360	729	192	1368	218
426	21	496	727	1273	815	750	320	1326	1318	1369	1327
1370	1371	639	684	1262	358	1372	903	1373	1257	581	825
312	906	117	1374	1341	649	1375	767	1376	1264	556	1377
373	235	1378	826	1379	1336	731	872	1316	203	468	1304
1380	848	708	167	543	1291	1261	626	1256	888	712	206
1340	307	1294	854	535	338	1337	1315	1259	909	410	37
244	620	1277	443	1338	362	75	1287	471	867	153	768
575	1381	811	824	1314	171	1323	105	1254	1290	1328	271
1303	1266	846	183	441	1300	648	521	1324	1255	146	893
375	1382	1383	508	354	1283	696	54	254	282	283	656
785	788	448	841	1310	506	137	1317	1295	165	830	1333
339	1286	701	714	679	751	1299	540	704	249	147	182
1260	1384	364	1302	157	293	1306	1385	633	92	420	1309
427	288	716	1335	391	1321	55	258	1280	1288	201	61
1386	152	1387	1388	901	705	350	1389	752	1390	1	1391
720	711	755	113	562	1284	1292	1392	1297	314	437	547
1393	507	1305	520	1312	199	119	524	781	170	122	275
1276											

Clé de chiffrement de la deuxième version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets) :

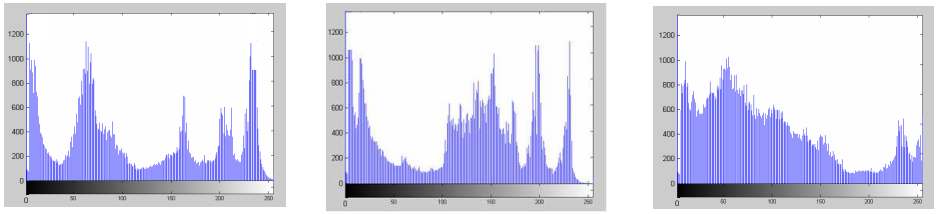
6	162	2	1	279	439	507	352	360	804	758	567
472	465	78	992	500	327	830	580	948	101	535	173
756	991	805	522	928	10	147	416	461	606	769	803
827	692	341	982	129	12	244	944	197	477	831	139
684	658	255	593	356	49	36	172	504	108	272	375
248	575	302	299	81	358	496	773	607	476	852	562
382	726	134	846	667	641	98	73	735	230	548	540
902	484	64	727	643	542	640	380	556	632	723	
835	436	748	109	721	654	619	212	277	894	211	475
705	335	257	844	856	781	938	462	961	11	623	62
651	254	74	4	746	265	525	130	943	185	52	473
457	160	251	860	59	939	447	121	946	511	757	934
258	136	974	350	35	50	483	491	965	32	941	453
339	344	514	929	137	470	27	502	724	131	184	512
362	677	213	871	802	981	39	5	570	345	848	76
328	884	412	238	68	391	376	488	325	292	882	71
142	970	810	196	220	796	730	132	296	622	550	490
590	574	3	560	87	826	158	8	518	817	973	635

819	409	689	498	206	353	526	890	41	617	440	647
396	915	935	797	96	242	975	297	285	608	177	264
782	563	99	203	659	7	544	104	761	629	719	271
333	329	877	581	332	838	778	611	704	762	674	156
972	899	610	930	366	785	927	976	956	31	263	117
312	315	664	775	752	337	413	996	878	853	963	46
637	866	111	538	310	105	30	418	445	179	814	933
151	867	597	274	86	467	609	984	273	152	834	43
408	303	594	44	505	696	419	516	631	732	794	712
284	588	682	435	874	521	858	589	969	906	33	138
398	722	698	529	372	524	93	378	122	486	554	494
448	655	103	833	547	808	9	148	924	855	728	713
386	691	751	226	468	239	311	688	482	628	28	737
753	566	373	602	543	168	118	318	326	813	14	307
309	275	485	818	942	870	653	481	868	950	993	861
690	499	766	648	443	349	679	280	493	673	824	56
424	199	321	889	621	962	166	218	977	714	487	444
336	582	281	402	624	553	387	747	441	390	546	261
743	306	492	603	58	616	267	235	89	528	703	630
811	364	497	381	715	800	250	665	893	94	379	361
913	627	532	262	530	585	980	295	388	21	697	241
903	828	971	851	995	686	672	469	888	615	519	905
595	908	501	832	169	433	897	718	63	508	228	80
816	385	922	115	862	91	423	428	95	559	427	873
786	779	676	909	15	604	979	710	354	680	370	578
601	739	150	245	157	907	342	128	953	417	454	404
881	223	240	82	17	270	646	792	125	153	940	896
266	694	571	552	135	127	898	986	901	669	772	229
842	837	916	346	371	523	47	661	464	771	783	839
790	18	308	460	577	702	154	289	770	990	57	463
114	69	24	642	400	120	671	232	650	97	48	249
774	895	605	178	919	314	863	442	926	53	558	634
921	989	458	29	587	568	741	406	434	145	657	561
791	920	369	340	246	645	821	749	429	614	403	539
330	509	144	945	225	918	541	394	112	537	706	850
405	60	879	900	845	222	374	789	474	625	549	425
209	584	276	317	421	652	426	67	843	599	198	355
947	338	557	305	37	431	319	678	290	876	155	638
596	806	451	331	124	744	809	54	411	51	182	883
912	825	170	807	16	983	857	904	666	836	955	660
988	61	175	420	159	395	742	951	515	133	26	869
815	268	410	205	219	176	191	446	729	347	252	864
534	958	849	207	466	234	65	531	90	269	75	478
384	221	181	795	183	551	22	968	459	911	479	399
414	286	734	670	192	685	925	959	545	777	300	287
174	415	357	106	88	840	960	987	42	859	636	780
471	745	294	701	592	164	600	750	733	377	764	320
949	208	422	716	681	291	140	301	449	711	83	393
165	668	343	952	84	917	700	892	126	113	639	195
954	759	966	573	85	644	102	231	224	92	885	914
180	143	613	40	141	994	887	348	875	200	20	367
188	316	456	110	720	489	503	569	450	260	626	822
119	38	383	793	767	708	186	171	167	579	736	812
572	72	784	163	77	365	555	392	243	765	452	146
283	187	322	282	107	760	683	886	407	693	662	216
841	23	707	363	564	210	359	55	999	964	401	738
217	324	847	506	598	116	201	13	586	253	576	695
823	798	865	591	699	323	256	190	389	872	788	620
288	829	533	763	247	717	313	298	19	985	854	149
278	520	931	998	455	100	687	820	34	351	768	193
923	967	740	755	480	432	25	227	725	161	799	932
189	675	937	663	204	66	957	237	565	334	891	583
618	787	233	304	259	397	997	536	656	612	437	527
495	202	368	936	510	438	633	649	801	754	910	776
880	123	194	214	236	513	978	293	215	430	517	709
731	1165	1126	1054	1145	1227	1122	1168	1335	1190	1338	1157
1224	1003	1310	1038	45	1301	1153	1295	1316	1032	1318	1329
1167	1115	70	1004	1339	1340	1341	1081	1150	1294	1342	1343
1319	1086	1125	1018	1269	1118	1012	1214	1344	1129	79	1332
1061	1213	1111	1205	1305	1130	1252	1093	1315	1057	1037	1299

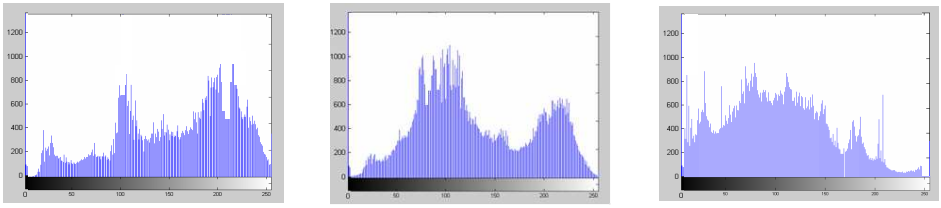
1083	1277	1062	1219	1345	1238	1346	1096	1221	1022	1103	1195
1257	1142	1050	1347	1274	1077	1088	1006	1058	1220	1046	1020
1171	1268	1116	1348	1278	1013	1188	1291	1237	1314	1296	1349
1350	1351	1109	1192	1352	1353	1189	1267	1229	1005	1155	1107
1243	1060	1008	1091	1128	1064	1023	1041	1293	1245	1248	1102
1113	1354	1263	1131	1355	1163	1356	1015	1039	1357	1247	1000
1053	1311	1358	1283	1049	1124	1065	1359	1360	1361	1281	1173
1254	1181	1114	1276	1191	1336	1104	1223	1184	1362	1079	1123
1226	1259	1137	1273	1112	1070	1363	1287	1035	1197	1364	1365
1140	1366	1367	1368	1127	1288	1369	1199	1244	1010	1133	1105
1024	1370	1371	1048	1180	1312	1317	1045	1139	1330	1177	1019
1040	1029	1055	1230	1225	1285	1275	1334	1372	1085	1036	1076
1206	1258	1208	1333	1028	1002	1154	1320	1186	1166	1161	1030
1135	1261	1095	1101	1067	1280	1136	1194	1097	1302	1260	1099
1034	1373	1172	1063	1374	1132	1094	1080	1337	1234	1265	1175
1066	1169	1047	1087	1246	1121	1375	1193	1068	1292	1376	1377
1138	1378	1325	1075	1326	1232	1203	1249	1092	1073	1379	1297
1152	1001	1071	1306	1026	1256	1242	1304	1380	1148	1156	1321
1201	1303	1021	1239	1210	1222	1381	1290	1251	1204	1025	1264
1382	1383	1084	1187	1228	1271	1146	1185	1078	1324	1384	1385
1215	1216	1176	1170	1100	1098	1212	1016	1031	1218	1209	1159
1196	1217	1178	1233	1241	1106	1162	1266	1141	1143	1090	1240
1250	1298	1328	1108	1307	1386	1262	1089	1387	1074	1027	1313
1388	1389	1056	1179	1009	1151	1119	1044	1255	1331	1134	1286
1272	1117	1033	1147	1231	1052	1149	1207	1309	1200	1164	1282
1390	1007	1160	1198	1082	1072	1308	1284	1253	1174	1069	1391
1202	1392	1182	1017	1270	1235	1236	1042	1043	1211	1322	1051
1183	1011	1120	1323	1158	1289	1279	1327	1110	1393	1014	1300
1059	1144										



(a) ***(b)*** ***(c)***



(d)



(e)

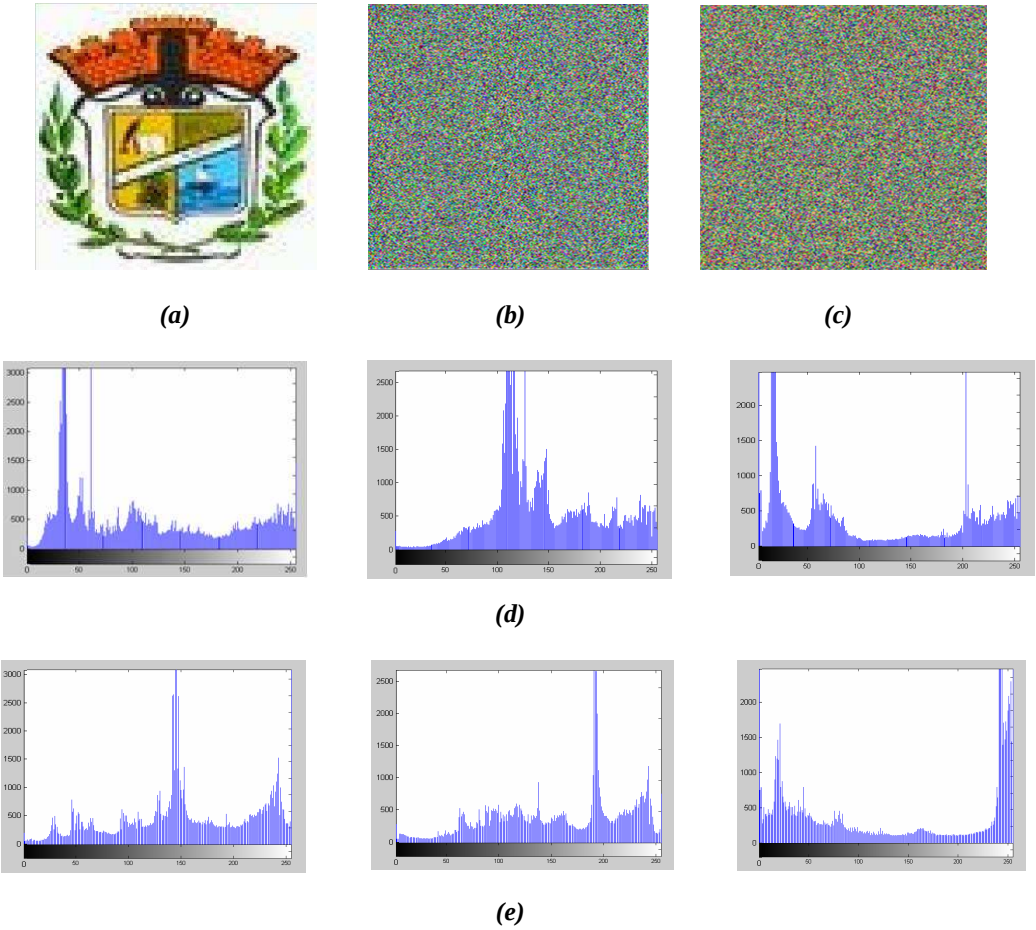
Figure IV.6. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

5	623	186	3	733	275	321	55	398	496	437	199
363	266	63	526	685	316	438	200	324	525	595	106
194	114	109	386	198	126	674	374	751	755	176	273
541	467	369	279	516	661	197	609	447	4	86	679
716	196	117	422	736	430	129	474	261	762	527	630
759	737	615	362	694	367	687	583	581	646	19	676
234	645	203	574	284	580	332	389	517	613	743	637
732	272	209	431	453	408	495	557	651	394	30	242
337	735	551	566	473	477	391	104	50	79	401	11
535	514	500	269	352	143	372	216	664	267	341	499
184	713	543	553	458	300	22	51	718	325	712	68
532	704	512	501	1	7	547	690	763	256	522	536
130	381	215	511	224	371	647	94	485	70	78	494
761	220	436	622	21	577	149	725	582	443	322	334
191	576	424	697	513	10	296	96	550	345	492	88
154	585	506	239	549	456	519	590	702	399	426	505
212	339	120	727	631	692	166	159	201	67	684	742
102	440	537	225	29	655	709	562	567	308	171	644
539	452	264	502	592	416	510	274	111	753	621	561
235	208	226	45	468	370	175	471	410	747	349	42
304	387	677	230	439	157	534	121	170	657	8	552
617	213	691	455	150	43	223	678	484	95	301	36
442	503	598	579	449	610	731	156	635	563	90	249
103	247	432	524	206	633	6	533	584	160	564	721
754	750	91	538	162	722	508	285	250	353	254	744
726	717	597	283	185	559	625	393	570	669	69	504
174	421	457	329	355	87	268	616	136	195	639	336
100	720	668	518	54	116	214	605	319	303	190	58
23	405	342	548	428	376	231	470	373	34	320	670
49	475	132	181	588	271	418	314	167	703	243	404
364	123	419	297	227	48	291	624	461	81	528	738
192	105	766	619	62	25	187	330	115	636	441	450
152	545	760	498	365	221	71	383	153	530	134	660
662	189	757	396	348	368	602	596	238	714	205	671
749	614	435	493	317	395	672	307	182	481	17	47
415	338	27	57	682	640	648	765	60	219	356	689
155	173	673	629	683	529	600	658	448	32	292	384
125	172	327	681	113	740	351	603	24	312	675	148
40	643	15	309	402	294	93	217	232	122	434	135
326	183	281	315	593	480	118	53	288	358	89	142
128	39	699	131	241	41	521	66	343	282	741	734
218	277	420	202	540	13	642	486	589	204	723	478
730	715	340	710	764	80	653	472	211	445	483	411
459	464	433	18	392	138	618	406	377	571	193	591
360	366	158	680	252	469	382	145	52	35	462	72
407	278	133	73	101	412	26	165	745	344	575	12
599	400	739	695	748	310	568	139	293	140	608	403
375	708	587	768	298	666	76	554	251	606	414	572
245	97	222	74	270	507	388	44	601	696	546	594
229	107	706	168	311	350	487	112	667	255	33	444
361	177	454	169	164	479	509	82	686	463	515	228
663	244	357	649	711	259	628	265	397	707	728	423
665	2	632	626	460	654	210	290	124	767	163	333
280	429	295	489	451	698	565	127	37	641	491	688
161	531	302	719	179	75	729	724	604	482	693	347
240	413	61	490	746	378	586	258	84	627	427	248
263	335	16	542	523	233	556	31	144	257	38	64
331	92	544	701	650	466	56	573	188	659	306	578
253	287	756	236	425	299	119	611	65	9	555	323
141	417	110	318	359	20	146	77	656	758	652	207
289	151	237	98	385	620	700	85	137	634	147	260
354	180	276	560	59	380	465	313	569	346	409	638
558	108	328	612	476	488	46	446	262	286	246	178
607	83	705	752	390	497	99	305	28	379	14	520

Clé de chiffrement de la deuxième version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

81	506	289	165	107	184	553	751	570	17	729	724
457	204	65	760	510	256	116	708	314	296	315	381
538	6	734	302	480	33	139	345	54	144	113	57
393	424	399	163	470	71	599	733	51	653	151	30
175	743	321	556	187	232	112	323	363	707	588	183
235	260	128	37	477	453	462	267	88	304	351	56
308	359	762	464	121	374	294	382	715	392	478	609
647	205	431	253	89	594	666	262	622	592	224	322
311	164	277	29	341	291	154	646	699	522	495	74
625	47	320	395	644	77	43	100	494	723	402	706
702	105	152	244	283	737	404	340	750	629	396	764
125	412	49	387	22	364	484	643	337	242	131	389
342	690	120	572	567	435	745	176	466	13	209	403
517	508	279	265	4	252	117	132	475	587	391	310
635	331	492	502	360	372	380	456	710	104	96	669
565	416	257	531	3	326	612	171	568	1	563	444
434	507	41	259	505	577	84	192	147	447	597	747
347	711	335	744	383	421	540	367	394	660	143	281
657	178	339	418	562	436	301	766	27	94	516	66
703	161	91	313	58	63	136	610	203	45	162	292
195	483	317	705	228	62	196	476	18	539	64	368
141	451	503	39	169	307	229	297	672	350	670	626
596	309	9	19	740	353	519	160	673	134	614	400
541	75	649	605	748	223	82	137	388	525	667	655
535	640	90	585	659	448	243	295	170	48	299	237
754	533	726	452	548	528	231	230	590	700	24	586
182	678	732	101	573	290	156	272	207	631	607	701
177	445	155	365	545	73	38	583	284	80	560	138
591	469	559	97	423	398	119	208	603	527	512	514
767	78	12	173	490	95	324	126	619	349	69	488
665	662	460	693	459	180	420	604	509	720	598	595
529	174	5	50	87	271	683	561	696	422	188	59
245	620	439	482	240	580	361	385	685	593	755	142
358	274	415	497	571	698	677	370	536	679	31	663
641	219	468	736	375	222	189	757	432	158	216	639
471	413	633	455	23	85	642	407	168	651	675	238
206	450	123	146	465	518	276	739	546	319	199	145
611	106	689	636	688	714	461	768	348	632	214	133
442	333	46	129	118	344	654	10	68	722	627	379
581	575	758	730	390	661	411	682	172	249	334	14
652	695	11	336	576	487	278	725	268	384	61	630
191	520	211	521	239	589	756	26	286	409	417	127
227	501	55	298	426	713	735	35	148	159	437	401
186	330	99	638	618	550	410	226	15	44	40	430
742	103	140	458	111	369	449	373	537	202	615	83
270	674	472	600	564	608	397	645	215	234	721	130
261	153	269	515	579	316	405	741	621	727	628	656
254	354	352	122	72	763	76	287	34	752	181	446
544	21	236	566	441	524	617	300	513	79	248	467
115	731	474	408	704	491	305	547	557	377	20	427
719	697	102	357	52	346	526	709	582	185	718	149
542	481	7	716	318	761	606	312	288	201	429	109
549	493	338	285	378	454	86	438	613	419	601	329
303	293	712	110	728	247	433	157	443	282	212	218
197	634	694	746	489	135	558	623	717	680	246	648
264	664	366	574	676	124	362	36	53	8	213	200
530	668	114	555	543	485	650	498	233	42	150	325
166	98	511	343	749	738	414	552	753	251	250	255
356	280	386	266	551	28	273	328	93	499	275	92
486	108	198	534	32	532	194	70	569	217	691	658
406	578	16	263	327	2	221	496	765	463	332	681
616	500	25	428	554	479	306	67	637	584	371	190
692	473	258	60	759	220	425	440	671	523	167	225
210	355	684	602	179	687	376	193	686	624	504	241



(a) ***(b)*** ***(c)***

(d)

(e)

Figure IV.7. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée*
(d) *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

92	39	9	38	13	16	45	76	98	47	24	8
83	40	3	78	80	90	61	26	68	89	53	52
19	31	50	18	12	14	54	64	2	46	58	65
35	71	5	28	74	36	44	73	79	99	96	62
63	94	29	15	37	11	75	10	30	43	7	51
91	55	33	66	57	23	81	41	70	56	77	88
4	95	25	1	69	86	22	82	97	67	42	59
20	87	32	49	72	48	27	425	652	573	441	543
468	129	748	420	738	513	353	453	569	700	185	525
707	143	512	264	237	198	755	658	487	221	741	557
484	647	227	257	708	162	179	579	197	724	60	326
629	757	548	220	413	115	266	258	120	443	218	639
253	669	673	260	712	202	345	430	225	561	284	449
481	109	294	661	248	552	710	354	433	644	372	630
161	684	648	21	476	432	110	489	331	503	403	111
507	363	637	680	17	734	556	588	438	692	666	367
753	723	613	287	34	604	505	168	273	267	421	564
239	542	742	382	642	471	508	278	614	333	359	234
136	764	285	765	545	672	259	114	214	572	170	520
147	458	502	352	244	659	423	688	537	175	386	706
609	678	328	149	272	462	393	567	379	6	144	330
523	145	562	417	119	761	767	226	478	355	726	518
737	722	583	127	440	238	397	429	339	247	173	450
296	635	254	378	341	213	488	134	409	292	731	427

399	494	193	280	314	93	704	123	469	148	634	657
303	344	501	705	404	733	171	627	662	551	311	470
743	510	401	157	554	346	491	340	698	256	240	347
125	611	727	589	358	568	374	325	631	307	361	745
308	756	746	164	369	131	306	448	640	245	565	118
586	668	167	580	313	553	758	189	84	124	350	262
645	703	676	574	540	255	348	575	515	286	128	664
398	366	531	113	217	456	300	681	224	223	593	699
691	422	585	506	402	434	641	270	528	446	426	696
301	656	690	302	529	151	682	204	735	571	416	663
165	108	514	435	444	689	190	290	158	602	140	335
299	582	174	677	516	616	203	736	104	454	536	625
279	577	396	495	312	750	310	349	559	205	315	141
728	269	192	250	208	695	718	709	740	618	687	587
550	283	322	142	209	282	199	187	276	336	653	229
321	176	730	626	714	342	327	219	599	139	608	112
752	566	603	581	390	210	466	373	650	576	152	447
233	163	384	196	242	362	768	180	188	570	461	601
126	216	395	365	497	309	751	201	230	739	595	177
178	249	133	605	766	116	207	394	667	318	281	194
530	600	117	376	235	636	623	646	85	732	598	555
437	241	693	334	222	100	293	612	406	184	492	563
560	547	534	679	407	215	535	351	674	332	475	670
271	212	275	320	558	749	721	418	181	186	633	371
701	591	385	517	533	467	638	686	490	590	364	375
154	606	683	388	762	759	200	474	729	649	405	360
500	166	655	389	343	538	628	493	524	596	483	594
527	172	243	549	754	304	473	415	477	103	597	615
464	295	725	368	412	195	277	231	713	632	297	480
211	442	459	620	532	660	651	319	410	544	485	419
445	744	101	252	191	121	298	486	265	439	182	463
150	452	392	457	400	146	106	747	526	357	370	665
317	482	763	377	720	451	578	621	455	356	324	521
130	288	228	156	323	122	504	643	246	522	496	619
436	697	694	675	431	316	584	232	607	107	381	671
719	261	428	411	408	498	105	159	717	610	539	715
153	268	546	617	387	102	160	414	685	716	465	460
338	511	391	622	289	291	479	138	702	305	711	424
654	155	499	169	329	137	183	251	383	132	592	472
206	263	541	135	760	509	274	624	337	380	519	236

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

528	192	38	202	98	315	156	142	97	339	304	457
569	700	485	538	200	292	180	544	443	684	130	287
352	689	571	468	536	101	169	244	639	353	215	24
735	501	545	266	628	575	359	745	653	1	694	535
411	37	9	342	491	497	146	278	620	88	533	80
151	380	298	137	521	508	409	61	123	191	728	662
558	115	398	514	454	50	323	541	429	410	686	488
228	232	247	559	378	234	181	261	233	345	243	643
758	387	384	33	8	761	412	346	371	695	750	40
308	477	128	179	487	493	294	604	11	504	129	710
68	39	196	739	725	753	43	63	187	73	26	749
401	578	162	176	404	395	699	302	140	270	157	272
563	341	249	81	381	92	652	58	673	289	67	712
547	768	94	322	172	218	241	69	721	438	618	645
242	723	611	556	635	688	297	193	641	474	553	62
654	25	502	310	301	148	269	428	377	70	106	592
672	617	258	439	708	658	34	737	274	407	113	112
279	12	209	119	326	531	207	677	584	42	282	330
199	237	268	648	251	731	447	338	744	685	126	557
216	610	621	483	328	467	205	674	184	259	44	210
649	532	75	600	114	625	473	290	334	83	596	752
256	434	470	82	650	425	141	482	763	534	116	201
354	385	96	549	601	726	576	717	283	173	730	212
707	525	522	373	452	281	348	117	227	318	589	565
208	663	214	505	711	609	276	93	526	31	767	23
636	687	667	372	570	303	581	369	252	510	343	546
540	219	177	418	182	722	760	495	171	656	255	78

329	691	608	293	580	690	229	27	262	513	554	223
296	22	152	432	57	333	159	211	347	516	665	675
727	676	124	450	713	299	564	86	679	4	757	226
311	335	393	402	161	217	702	102	741	54	260	397
637	213	118	548	230	32	364	622	382	111	254	379
568	107	174	165	366	195	417	153	759	512	697	764
367	696	742	76	597	394	435	158	480	361	103	149
120	448	706	755	716	178	166	520	602	537	422	740
682	577	327	357	475	168	560	585	598	436	35	413
110	189	414	250	127	552	358	332	403	190	668	41
587	424	122	588	567	391	612	17	376	416	631	291
355	175	307	222	456	154	340	66	427	586	331	238
542	84	370	484	295	566	183	471	437	236	529	583
593	52	459	319	614	271	449	613	131	698	55	6
313	19	280	60	550	325	349	59	46	714	396	671
453	45	605	245	337	105	85	350	469	167	664	400
669	351	623	660	197	720	591	594	224	16	246	51
421	562	692	2	724	478	388	442	666	53	3	320
419	659	132	91	383	423	235	121	71	462	461	481
543	20	147	160	515	415	316	87	527	203	300	754
286	138	681	288	15	284	579	511	143	704	524	65
509	632	198	574	64	163	29	455	231	139	624	733
638	539	765	275	309	680	472	263	108	321	463	607
640	693	523	135	630	619	701	389	683	627	732	170
709	99	603	420	426	312	441	374	440	392	747	77
719	136	661	317	756	368	406	48	476	489	305	748
109	715	465	615	285	496	375	155	21	446	519	408
561	18	498	220	734	336	5	530	188	762	492	399
464	13	518	555	499	766	433	431	204	458	273	751
248	74	89	479	186	100	705	506	551	500	405	572
28	267	14	743	444	164	150	240	430	362	655	257
616	265	657	185	494	703	644	642	56	134	646	390
445	104	678	647	729	718	363	466	306	599	629	517
360	10	507	746	225	736	595	47	253	651	194	95
582	633	490	486	451	344	606	738	7	634	277	386
503	144	324	30	670	36	365	125	145	356	79	264
314	590	90	626	460	573	206	133	49	239	72	221

Le tableau suivant résume les résultats relatifs au temps de calcul (temps de chiffrement), la taille de clé, la quantité de phéromone et l'efficacité dans le cas de chiffrement des différentes données.

			Taille donnée (éléments)	**Taille clé (bits)**	**Efficacité**	**Temps calcul (s)**	**Phéromone**
Données Texte	Texte1	Version-Chiff1	1084	15323	11784.0	5.63	9,06
		Version-Chiff2			12030.0	6.02	9,13
	Texte2	Version-Chiff1	1151	15323	14498.0	6.7	8,67
		Version-Chiff2			14712.0	6.58	8,65
Données Images	Lena	Version-Chiff1	131 X 131	7680	135362.0	2.16	2.75
		Version-Chiff2			130074.0	2.31	1.17
	Logo	Version-Chiff1	420 X 395	7680	244926.0	2.87	23.11
		Version-Chiff2			226188.0	2.84	21.81

Tableau IV.2. *Résultats obtenus par AntCrypt.*

IV.4. Discussion et évaluation des résultats

La première chose à remarquer, d'après les résultats résumés à travers le tableau IV.2, c'est que le temps de chiffrement des données texte est beaucoup plus grand que celui des données images malgré que la taille des données images est généralement plus grande en comparaison à celle des données texte. Cela revient au fait que la taille du vecteur codant les données texte (1393 éléments) est beaucoup plus grande que celle du vecteur codant les données images (768 éléments), ce qui nécessitera beaucoup plus de temps de calcul lors de l'application du processus de résolution à base de fourmis. Toutefois, le temps de chiffrement est raisonnable pour les deux types de données texte ou images.

Nous remarquons, aussi, que les temps de calcul des différentes versions d'une même donnée sont proches, car ils sont indépendants de la taille de la donnée à chiffrer (une différence apparaît quand la dissemblance entre les tailles de deux données de même type sera importante et qui représente un temps de lecture ou de codage). En effet, la taille du vecteur codant toutes les données de même type et soumises au chiffrement est la même (1393 éléments pour le cas de texte / 768 éléments pour le cas d'images), donc sa manipulation sur l'ensemble des générations prend le même temps. La toute petite différence correspond au temps de codage des données avant le lancement du processus de chiffrement. Toutefois, la légère différence entre le temps de chiffrement des deux versions d'une même donnée revient au caractère aléatoire exploité dans certaines phases de l'algorithme.

Pour le cas de chiffrement des deux données images Lena et Logo, nous constatons que le temps de chiffrement de l'image Logo est plus grand que celui de l'image Lena. La différence est due au fait que le temps de chiffrement est la somme des temps de lecture puis codage de l'image qui varie suivant la taille de l'image, plus le temps de manipulation du vecteur codant les solutions qui n'a aucune relation avec la taille de l'image. Donc, la différence en temps de chiffrement des deux images est justifiée.

La valeur de la phéromone dépend de la valeur d'efficacité maximale et de l'efficacité de la solution calculée. Il est bien clair que pour de grandes valeurs d'efficacité, une amélioration de la quantité de phéromone sera marquée. Prenons l'exemple du premier message où pour la première version chiffrée nous avons obtenu 2166 / 9,06 pour efficacité / phéromone. Cependant, pour la deuxième version chiffrée, nous avons eu 2184 / 9,13. Donc, l'amélioration de l'efficacité est certainement accompagnée d'une amélioration de la quantité de phéromone. Il sera utile de rappeler, ici, que l'évaporation est la cause de diminution de la quantité de phéromone.

En mesurant le taux de différence entre les deux versions chiffrées de l'image Lena et l'image originale (l'image Lena), le tableau suivant récapitule les résultats obtenus en termes de NPCR, MAE et MSE. Il est clair que les deux versions chiffrées sont presque totalement différentes de l'image originale. Autrement dit, les pixels formant les versions chiffrées sont presque tous changés, soit de positions, soit de nombres d'occurrences et de positions ; ce qui donnera lieu à un très bon niveau de confusion.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.9207	1.08	0.73
Lena-version chiffrée 2	0.9082	1.03	0.69

TableauIV.3. *Niveaux de confusion de AntCrypt.*

Maintenant comparant AntCrypt avec le plus rapide de nos trois algorithmes développés et présentés dans le précédent chapitre, bien sûre sur le plan de temps de calcul. Le tableau suivant rappelle les temps de convergence des deux algorithmes OEEA et AntCrypt. Ce dernier a montré un temps de calcul meilleur que celui de OEEA que ça soit pour chiffrer des données textes ou images malgré qu'aucune explication exacte ne peut être donnée du fait que les deux algorithmes utilisent un même mode de chiffrement (chiffrement à base d'occurrences), donc un même codage qui est exploité par les composants de la métaheuristique impliquée (opérateurs génétiques et mécanismes de la sélection naturelle pour OEEA et fourmis et mécanismes de manipulation de phéromone pour AntCrypt). Et vu que le temps de convergence de notre algorithme de chiffrement basé sur les mécanismes évolutionnaires (OEEA) est légèrement plus grand que celui basé sur les fourmis (AntCrypt), donc possible que les premiers mécanismes consomment plus de temps de calcul que les deuxièmes, mais aucune preuve n'est connue !

	Temps calcul (s)	
	Texte 1	Lena
AntCrypt	5.825	2.235
OEEA	6.54	3.37

Tableau IV.4. *Temps de calcul de AntCrypt en comparaison avec le temps de calcul de OEEA.*

Remarque :

Dans le tableau IV.4, les temps de calcul de l'algorithme AntCrypt pour les deux cas de données (Texte 1 et Lena), représentent la moyenne des temps de calcul des deux versions chiffrées (Voir tableau IV.2).

De même et vu que la méthode de chiffrement présentée dans ce chapitre (Cryptage à base de fourmis) et celle présentée dans le précédent chapitre (Cryptage évolutionnaire) appartiennent à une même catégorie mère qui est celle de métaheuristiques, donc toute les deux vont bénéficier de quelques caractéristiques cryptanalytiques en commun. Il s'agit de l'aspect non déterministe démontré à travers les résultats présentés ci-dessus où deux versions chiffrées différemment correspondent toutes les deux à une même donnée originale (voir figures IV.4, IV.5, IV.6 et IV.7), et d'une robustesse contre les types d'attaques les plus avancées (attaque exhaustive, attaque statistique, attaque fréquentielle et attaque différentielle) comme c'était démontré dans la section III.5 du chapitre précédent.

IV.5. Conclusion

ACO est une méthode stochastique qui requiert de plus en plus l'attention de la communauté scientifique. Cette métaheuristique a prouvé sa performance pour plusieurs problèmes d'optimisation combinatoire. En effet, l'utilisation des traces de phéromone permet d'exploiter l'expérience de recherche acquise par les fourmis et ainsi renforcer l'apprentissage pour la construction de solutions dans les itérations futures de l'algorithme. En même temps, l'information heuristique peut guider les fourmis vers les zones prometteuses de l'espace de recherche.

Ainsi et dans ce chapitre, nous avons présenté, interprété et discuté les résultats expérimentaux obtenus par l'application de notre algorithme de chiffrement proposé *AntCrypt* sur des données modèles texte et images. Nous sommes arrivés à fixer nos paramètres par expériences, en choisissant 40 générations et 12 fourmis en essayant de réaliser un compromis entre l'efficacité et le temps de chiffrement.

L'algorithme, ainsi développé et qui a été nommé *AntCrypt*, a été comparé avec nos algorithmes développés à base d'algorithmes évolutionnaires où il a montré de bonnes caractéristiques telles qu'un bon niveau confusionnel, un aspect non déterministe, une taille sécurisée de clés, lui offrant une bonne résistibilité contre les attaques les plus avancées. En plus le temps de convergence de cet algorithme est meilleur que celui du plus rapide de nos trois algorithmes proposés dans le précédent chapitre.

CHAPITRE V

CRYPTAGE TABOU DES DONNEES TEXTES ET IMAGES

V.1. Introduction

Nous nous sommes intéressés, dans les deux chapitres précédents, à la résolution du problème de cryptage de données texte et images par application de métaheuristiques à populations. Les approches que nous avons proposé sont caractérisées par leur qualité de confusion, leur temps d'exécution réduits pour certains d'eux (OEEA et AntCrypt) et leur robustesse face aux attaques avancées.

Dans ce chapitre, nous nous intéressons à l'exploitation d'une métaheuristique appartenant à une autre catégorie, les métaheuristiques à trajectoire. Il s'agit de la recherche tabou. Cette dernière s'appuie sur une recherche locale combinée à un mécanisme de prévention des cycles, grâce à un système de mémoire des mouvements précédemment appliqués ou des configurations visitées (la liste tabou).

De manière générale, une recherche locale démarre d'une solution initiale possible et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant. Un voisinage d'une certaine solution correspond à des éléments adjacents à cette solution dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration dans la solution courante ne puisse être faite.

Nous présentons, après la description de l'approche proposée, les résultats numériques obtenus afin de les comparer aux résultats obtenus par nos autres méthodes proposées (cryptage évolutionnaire et cryptage par colonies de fourmis) et certaines autres méthodes de la littérature.

V.2. Motivation

La recherche Tabou est une métaheuristique basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

L'originalité de la méthode de recherche tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objective f et l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum

quelconque par acceptation de la dégradation de la fonction objective, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste d'une certaine longueur, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

V.3. Algorithme proposé

À partir d'une solution initiale, le principe général de la recherche tabou est le suivant (voir Algorithme V.1). À chaque itération de la recherche, une partie ou l'ensemble des voisins de la solution est exploré, et un des mouvements parmi ceux de coût minimal est sélectionné. Ce mouvement est appliqué quel que soit son coût, à la condition de ne pas créer un cycle dans le processus de recherche locale à court terme (en menant à une configuration dont les caractéristiques sont stockées dans la liste tabou). Une exception est faite (critère d'aspiration) lorsque le mouvement permet d'atteindre une solution de meilleure qualité que la meilleure solution enregistrée à ce stade. La liste tabou est mise à jour à chaque itération de la recherche en fonction des mouvements choisis. Ce mécanisme permet de sortir des minima locaux en acceptant des mouvements détériorants et en empêchant parallèlement le retour immédiat à la solution de minimum local qui vient d'être quittée.

Algorithme V.1 *Schéma général d'un algorithme tabou*

```
Engendrer une configuration initiale s
s* ← s
T ← ∅ liste tabou
tant que condition d'arrêt non satisfaite faire
    m ← meilleur mouvement (i.e. minimisant f) parmi ceux non tabou
    ou ceux vérifiant un critère d'aspiration
    Modifier s en effectuant le mouvement m
    Mettre T à jour
    si f(s) < f(s*) alors
        s* ← s
    fin
fin
retourner s*
```

La succession des étapes suivantes montre plus de détails du schéma du mécanisme utilisé pour la construction d'une solution au problème étudié :

1) Détermination d'une solution initiale $S_{(0)}$ codant l'image originale.
2) Création puis la mise à jour d'une liste tabou qui mémorisera les déplacements effectués.
3) Choix aléatoire d'un nombre d'emplacements ou d'éléments de la solution courante ($i^{ème}$ solution) à déplacer.
4) Génération aléatoire d'un ensemble de voisins pour chacun des éléments choisis dans l'étape précédente.
5) Choix du meilleur voisin de l'ensemble généré.
6) Calcul tabou de la solution suivante $S_{(i+1)}$.
7) Création et la mise à jour d'une table de hachage qui comportera les dernières meilleures solutions obtenues.
8) Évaluation de la solution calculée $S_{(i+1)}$ obtenue à chaque itération.
9) Validation de la solution calculée $S_{(i+1)}$.
10) Vérification du critère d'arrêt de l'algorithme.

En ce qui suit, nous décrivons les principales étapes de l'algorithme de chiffrement tabou proposé et que nous l'avons appelé *TabuCrypt*.

V.3.1. Création de la solution initiale

La solution initiale exploitée par *TabuCrypt* s'obtient en codant les données originales suivant le même mode de codage adopté par *OEEA* ou *AntCrypt*. C'est d'ailleurs le même codage utilisé pour modéliser toutes les autres solutions.

V.3.2. Choix des éléments à déplacer

Les solutions sont représentées soit, par un vecteur englobant 1393 éléments pour le cas de manipulation de données texte soit, par un autre regroupant 768 éléments pour le cas de manipulation de données images. *TabuCrypt* cherche à réarranger aléatoirement le vecteur correspondant à la solution initiale codant la donnée originale en permutant les éléments entre eux. Toutefois, la taille du vecteur est assez grande (1393 éléments ou même 768 éléments), c'est pourquoi la manipulation un par un de ses éléments sur l'ensemble des itérations de l'algorithme nécessitera un grand temps de calcul. Ainsi, sauf un sous ensemble du vecteur globale sera manipuler pour déplacement. Cela permet, d'un coté, d'optimiser le temps de calcul et, d'un autre coté, de converger petit à petit vers la solution finale en évitant, ainsi, le problème de convergence prématurée.

V.3.3. Génération de voisinage

Le voisinage est le responsable sur le fait de pousser l'algorithme à l'amélioration de la qualité des résultats. Dans notre cas, il est généré de la manière suivante :

Pour chacun des éléments du sous-ensemble construit dans l'étape précédente, nous générons aléatoirement un ensemble de voisins candidats au déplacement avec l'élément en question. Lors de l'élection d'un voisin, les conditions suivantes doivent être vérifiées :

- ✓ Un voisin ne doit pas être élu plus qu'une fois, c'est-à-dire, tous les voisins de l'ensemble de voisins sont différents les uns des autres.
- ✓ Les voisins élus n'appartiennent pas dans la liste tabou.

Une fois l'ensemble des voisins construit, nous choisissons celui qui diffère le plus de l'élément à déplacer. Formellement, soit E_V l'ensemble de n voisins V, et E_i l'élément candidat au déplacement. La sélection du voisin v avec lequel l'élément E_i sera permuté se fait comme suit :

$$v = V_j \,/\, j = \overline{1,n} : Max\left(\left|E_i - V_j\right|\right) \qquad \text{(V.1)}$$

V.3.4. Mise à jour de la liste tabou

Dans notre cas et à une itération donnée, la liste tabou regroupe les voisins qui ont participé à des déplacements au cours des itérations précédentes avec l'un des éléments du sous-ensemble construit dans l'étape 2 décrite dans la section V.3.2. Ainsi, la mise à jour de cette liste (liste tabou) se fait à chaque fois qu'un élément sera permuté avec un voisin choisi suivant l'étape 3 décrite dans la section V.3.3 en ajoutant ce dernier à la liste des éléments tabous. A la fin de chaque itération, le contenu de la liste tabou sera effacé.

La politique utilisée pour rendre tabous certains éléments évite la modification des nombres d'occurrences originaux, puisque le processus de calcul de la solution ne doit pas modifier les éléments du vecteur initial mais plutôt il cherche à changer leur dispersion.

V.3.5. Calcul de solutions

Au cours d'une itération donnée, le calcul de la solution proposée à cette itération se base sur la solution calculée à l'itération précédente. En effet, des éléments du vecteur présentant la solution à l'itération précédente échangeront leurs positions avec celles des voisins désignés pour chacun d'eux. Cette nouvelle dispersion des éléments forme le nouveau vecteur solution. Il est à noter que les éléments non élus pour un déplacement garderont leurs positions sans modification.

V.3.6. Mise à jour de la table de hachage et évaluation des solutions

La table de hachage sert à mémoriser les différents points visités de l'espace de recherche en gardant les différentes solutions calculées sur l'ensemble des itérations. Le but sera d'interdire la recherche à revenir vers des points déjà visités à travers les précédentes itérations. Ainsi, une solution générée à la fin d'une certaine itération sera comparée au contenu de la table de hachage avant qu'elle soit soumise à une validation finale en vu de l'accepter comme intéressante solution à partir de laquelle la recherche peut continuer sur l'ensemble des itérations restantes. Si cette solution représente un nouveau point de l'espace non encore visité, elle sera ajoutée à la table de hachage pour quelle fera l'objet des futurs tests. Cette politique sert à échapper aux minima locaux et à favoriser l'exploration de nouveaux points de l'espace de solutions possibles.

La validation définitive d'une solution S_i calculée à une itération i donnée, consiste tout d'abord, à l'évaluer. Dans notre cas, l'évaluation se fait suivant la fonction d'évaluation illustrée par la formule suivante, notant que S_0 soit la solution initiale (voir étape 1).

$$F(S_i) = \sum_{j=1}^{l} \left| S_{i_j} - S_{0_j} \right| \qquad \text{(V.2)}$$

Où l : représente le nombre de valeurs possibles que peut prendre un élément d'une donnée (égale à 1393 pour le cas de données texte et 768 pour le cas de données images).

Une fois l'évaluation faite, la qualité de la solution en question sera comparée à celle de la solution calculée à l'itération précédente. Si elle est meilleure que la précédente, la recherche à travers la prochaine itération va se continuer à partir du point représenté par cette solution, sinon elle sera rejetée et la recherche continuera à partir de la solution calculée à l'itération précédente.

Ces points peuvent être résumés comme suit : à chaque itération la solution calculée est ajoutée à la table de hachage puis elle sera examinée selon les trois critères suivants :

1) Si la solution S_i obtenue n'est pas une nouvelle solution alors elle sera rejetée et les étapes de 2 à 6 sont à refaire.

2) Si la solution S_i obtenue est une nouvelle solution mais plus mauvaise que celle de l'itération *i-1* (S_{i-1}), de même elle est rejetée, mais sauvegardée dans la table de hachage, et les étapes de 2 à 6 sont à refaire.

3) Si la solution S_i obtenue est une nouvelle solution mais, cette fois ci, est meilleure que celle de l'itération (i-1), alors elle sera retenue pour continuer la recherche à partir de la prochaine itération et, ainsi, elle servira à calculer la solution suivante S_{i+1}. En même temps, elle sera mémorisée dans la table de hachage.

Remarque :
Dans notre algorithme, on considère comme *mouvement global* le passage d'une solution courante S_i vers une solution suivante S_{i+1}, en respectant les conditions citées précédemment. Toutefois, les *mouvements élémentaires* sont les déplacements des éléments et leurs voisins vers leurs positions mutuelles. Ainsi, un mouvement global émerge comme résultat des mouvements élémentaires.

V.3.7. Critère d'arrêt

Pour toute recherche itérative un critère d'arrêt est obligatoire pour éviter le problème de boucles infinies. Ce dernier qui doit être fixé à l'avance permet de déterminer le nombre de fois que la tâche à exécuter sera répétée.

Pour notre algorithme, nous avons choisi de fixer, expérimentalement, le nombre d'itérations. Ce choix influe directement sur longueur du temps de calcul et de la qualité de la solution construite.

V.3.8. Mécanismes avancés en recherche tabou

V.3.8.1. Intensification / Exploitation

L'intensification consiste à approfondir la recherche dans certaines régions de l'espace, identifiées comme susceptibles de contenir un optimum global.
Pour notre cas, les mouvements globaux en plus des mouvements élémentaires serviront à mieux intensifier la recherche. En effet, les mouvements élémentaires traduisent les déplacements des éléments d'un vecteur solution vers les positions de leurs voisins et vise versa. Ces voisins sont les éléments optimaux (les meilleurs voisins) de l'ensemble des voisins candidats au déplacement (voir étape 3). Donc, nous cherchons à exploiter les qualités des éléments. De même, les mouvements globaux cherchent à exploiter les qualités des solutions (les vecteurs d'éléments) en n'acceptant de poursuivre la recherche qu'à partir des solutions améliorantes lors du passage d'une itération à l'itération qui suit.

V.3.8.2. Diversification / Exploration

La diversification vise à utiliser des mouvements encore jamais réalisés afin d'explorer de nouvelles régions de l'espace de recherche et des régions éloignées du voisinage actuel. Dans notre cas, cette caractéristique est accomplie par la sélection aléatoire d'un nouveau voisinage non tabou pour chaque élément.

V.3.8.3. Critère d'aspiration

Le critère d'aspiration autorise un mouvement tabou sous certaines conditions. Il consiste à tester si la solution produite de statut tabou présente un coût inférieur ou de qualité meilleure que ceux de la meilleure solution trouvée jusqu'à présent. Si c'est le cas, le statut tabou de la solution est levé.

Dans notre cas, la notion de tabou est exploitée lors de la construction d'une certaine solution en interdisant les mouvements élémentaires vers les éléments voisins avec lesquels les éléments du sous-ensemble construit comme l'indique l'étape 2 ont échangé de positions. Ainsi, le mécanisme d'aspiration n'a pas de place dans notre méthode du fait que toute libération d'un élément de la liste tabou entraîne une modification des éléments des vecteurs et, par conséquent la modification des caractéristiques servira à calculer ultérieurement la donnée déchiffrée, tandis que notre but est, plutôt, modifier la répartition des éléments des vecteurs.

V.3.9. Déchiffrement

L'opération inverse au chiffrement est le déchiffrement qui permet de regénérer la donnée originale précédemment chiffrée par notre algorithme de chiffrement tabou *TabuCrypt*. Pour ce faire, ce processus exploite une information secrète calculée lors du chiffrement à coté de l'image chiffrée. Il s'agit d'une clé de session secrète. Elle représente les permutations des positions des l éléments (1393 éléments ou 768 éléments) du vecteur codant une certaine donnée à travers les différentes itérations. Une fois elle parviendra sans modification au destinataire approprié, elle servira à calculer la donnée originale.

V.3.10. Réglage des paramètres et résultats

Le schéma général de l'algorithme finalement implémentant les étapes de l'approche de cryptage tabou proposé est le suivant :

Algorithme V.2 *TabuCrypt*

Étape 1 : Générer une solution initiale de gain total G_0.
Étape 2 : Initialiser
- le nombre d'itération iter (à 1),
- la table de Hachage H (ajouter la solution initiale à H),
- la liste tabou T (vide).

Répéter les étapes de 3 à 11 jusqu'à iter = iterMax.
Étape 3 : Initialiser le compteur d'éléments (composants d'une solution)
Répéter les étapes de 4 à 6 jusqu'à
Étape 4 : Sélectionner aléatoirement un ensemble de voisins des éléments non tabous, puis choisir le meilleur de l'ensemble.
Étape 5 : Déclarer comme tabou le voisin choisi (voisin ajouté à T).
Étape 6 : Déplacer l'élément courant et le voisin choisi vers leurs positions mutuelles.
Étape 7 : Evaluer la nouvelle solution calculée pour mesurer son gain G_{iter}.
Étape 8 : Si : la solution calculée est déjà contenue dans H *alors :* Solution ignorée, *Sinon :* Ajouter la solution à H
Étape 9 : Si $G_{iter} > G_{iter-1}$ *alors* continuer le calcul à partir de $Solution_i$ *Sinon* Continuer le calcul à partir de la $solution_{iter-1}$
Étape 10 : Vider T.
Étape 11 : iter + +.

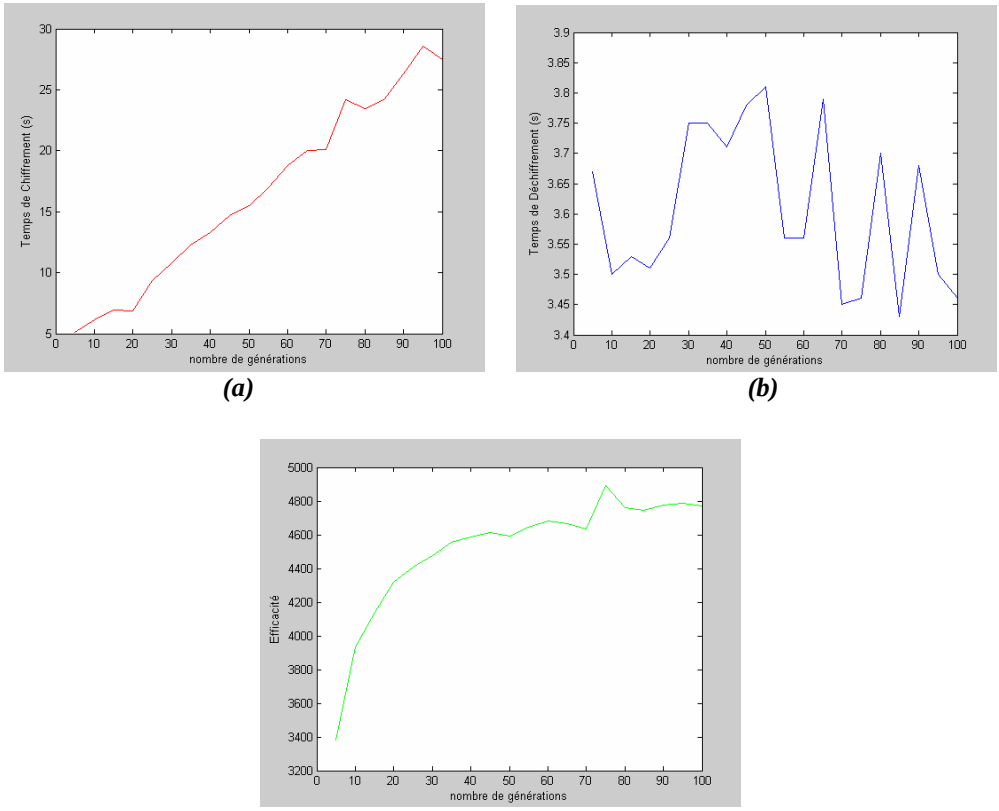
Nous précisons, dans cette section, les valeurs des paramètres de *TabuCrypt* ainsi que les résultats de son application sur les données modèles précédemment présentées (Texte1, Texte2, Image Lena et image Logo).

V.3.10.1. Réglage des paramètres

Les principaux paramètres de *TabuCrypt* concernent le nombre de générations ou d'itérations de l'algorithme et le nombre de voisins seront présentés.

Après *IterMax* itérations, fixée expérimentalement, la procédure de recherche tabou s'arrête et fournit la meilleure solution trouvée. La figure V.1 récapitule la moyenne des différentes valeurs de test et leurs impacts sur le temps de convergence et la qualité de la solution trouvée représentant une version chiffrée de l'image de test Lena.

D'après les résultats résumés dans la figure, ci-dessous, nous constatons que la meilleure efficacité a été obtenue pour un nombre de générations égale à 75 après un temps de chiffrement de 24.20 s et un temps de déchiffrement de 3.46 s. De même, les nombres de générations 80, 85, 90, 95 et 100 ont donné de bonnes valeurs d'efficacité (4760, 4744, 4777, 4789 et 4771, respectivement), mais leurs temps de chiffrement sont assez longs (23.46 s, 24.18 s, 26.29 s, 28.54 s et 27.48 s, respectivement).



(a) ***(b)*** ***(c)***

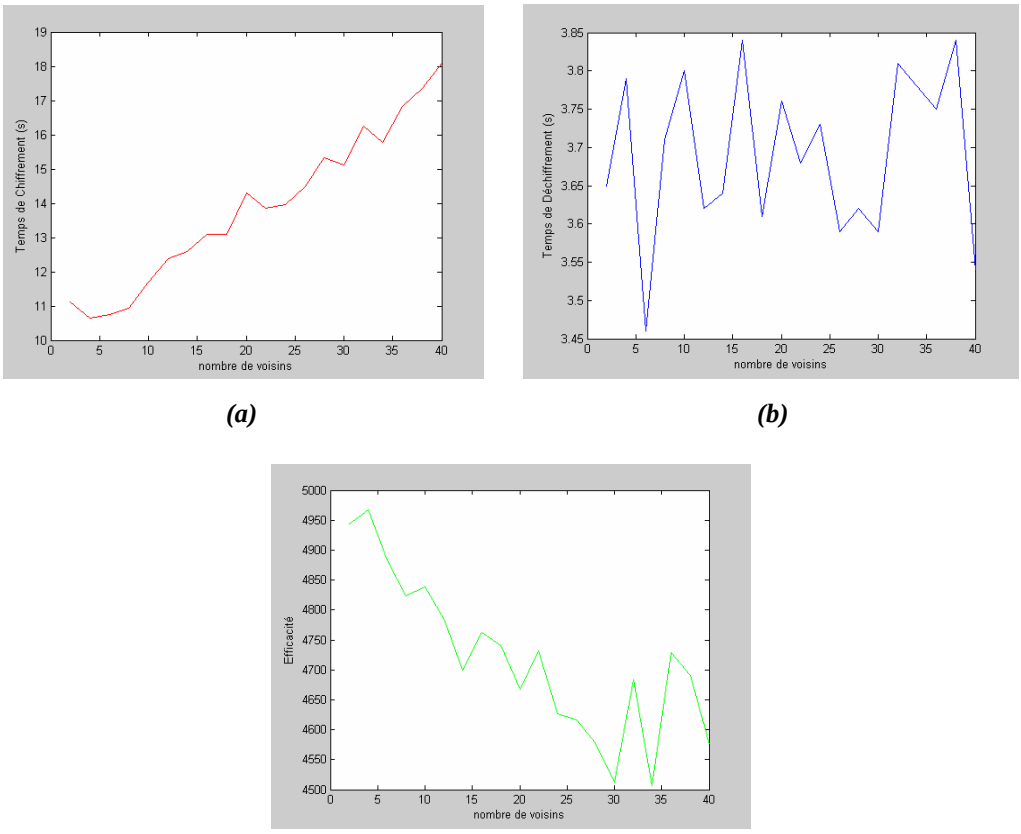
Figure V.1. *Résultats obtenus suivant les différentes valeurs de test de nombre d'itérations (générations) :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Toutefois, les résultats obtenus, pour le test avec 45 générations, sont acceptables que se soit sur le plan efficacité (4612) ou temps de chiffrement (14.70 s). D'autres valeurs de nombre de générations (55, 60, 65 et 70) ont donné des résultats d'efficacité de même ordre que celle du nombre de générations 45, mais après des temps de chiffrement relativement plus longs (29.93, 28.76, 20.04 et 20.12, respectivement). Ainsi, la valeur de *IterMax* a été fixée à 45 générations.

L'autre paramètre à régler est le nombre de voisin. Pour ce faire, nous avons utilisé le même mécanisme que celui utilisé pour régler le paramètre relatif au nombre d'itérations. Les résultats des différents tests sont résumés à travers la figure V.2 en donnant l'influence des différents nombres de voisins testés sur le temps de convergence et l'efficacité de la solution trouvée.

D'après le contenu des figures V.1 et V.2, la meilleure efficacité (4967) a été obtenue avec un nombre de voisins qui vaut 4 après un temps de chiffrement assez raisonnable et représentant le plus petit temps de calcul sur l'ensemble des tests effectués (10.64s).

Ainsi, nous avons opté à régler le paramètre relatif au nombre de voisins en lui attribuant la valeur 4.



(a) ***(b)***

(c)

Figure V.2. *Résultats obtenus suivant les différentes valeurs de test de nombre de voisins :* ***(a)*** *Temps de chiffrement,* ***(b)*** *Temps de déchiffrement,* ***(c)*** *Efficacité.*

Le tableau suivant (tableau V.1) résumé les valeurs de paramètres adoptés par *TabuCrypt.*

	Valeurs des paramètres de*TabuCrypt*
Nombre de voisins	4
Nombre de générations	45

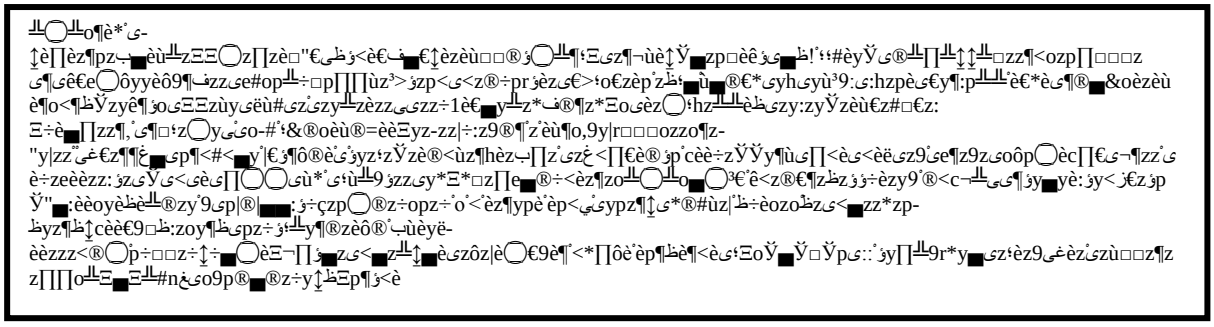
Tableau V.1. *Valeurs adoptées pour les paramètres de TabuCrypt.*

V.3.10.2. Résultats

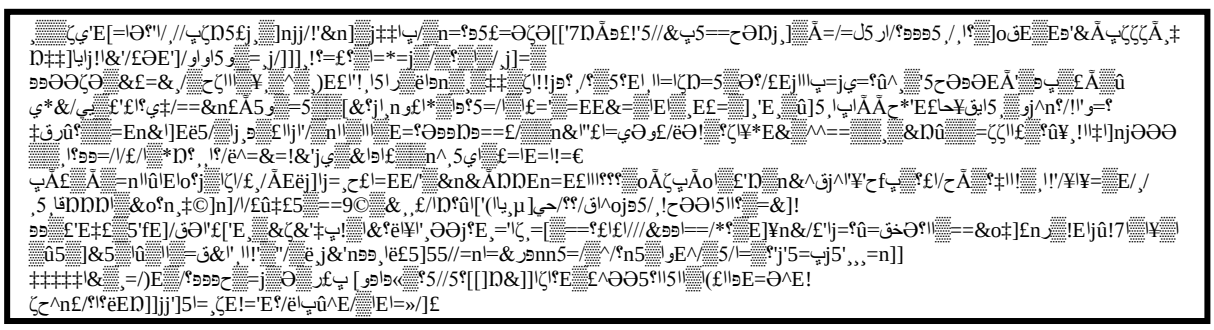
A ce niveau nous présentons les résultats de chiffrement par *TabuCrypt* des données tests. Deux versions chiffrées de chacune des données tests sont présentées accompagnées des clés générées dans chaque cas. Le tableau V.2 résume les temps de chiffrement et de déchiffrement ainsi que l'efficacité de chacun des exemples.

Indéniablement, avec l'essor fulgurant des nouvelles technologies, la cryptographie est omniprésente : Cartes bancaires, DVD, achats en ligne... et l'information devenue une denrée prisée qui doit être protégée loin aux yeux des inévitables curieux. Le grand public soit concerné et elle est devenue l'unique souci des grandes entreprises et des gouvernements. Et la question cruciale qui se pose : est ce que la protection totale des données est-elle une utopie ou une réalité?
En dépit de son antiquité et de son importante évolution, de la cryptographie classique à la cryptographie moderne à la cryptographie quantique, elle est toujours empêtrée dans ses limites et présente de nombreuses failles exploitables. A chaque apparition d'une nouvelle technique de chiffrement, des techniques de décryptage ont été développées ; toutes les techniques de chiffrement ont été décryptées plus ou moins rapidement. C'est une course-poursuite entre cryptographes et décrypteurs. En fait, les meilleurs systèmes de chiffrement sont comptés sur les bouts des doigts tel que : DES, IDEA, RSA, AES, PGP …

(a)



(b)



(c)

Figure V.3. (a) *Donnée originale Texte1,* ***(b)*** *Première version chiffrée,* ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte1 ($Taille_{Clé}$ = 1,8704 K octets)

728	898	353	668	901	875	906	437	425	40	739	524
797	890	877	786	357	654	889	258	566	884	883	899
399	721	896	133	905	483	410	895	735	117	900	876
878	303	479	907	706	879	100	813	91	453	152	634
874	186	597	902	480	341	99	42	736	438	456	893
794	344	881	267	887	474	617	191	885	56	892	886
897	377	601	904	34	98	672	880	903	673	496	760
373	882	894	888	347	463	750	891	520	248	465	571
624	116	573	744	866	646	926	68	1228	131	311	532
299	518	139	1134	1120	734	714	290	22	97	541	1047
1234	961	1110	819	52	153	255	1006	560	960	1077	942
1109	154	836	30	412	1119	1202	1037	1003	383	1238	335
392	1035	1241	119	448	409	1115	266	1036	857	971	194
713	912	422	378	500	782	274	1048	1102	651	1129	1091
810	696	1040	994	321	362	46	563	1148	149	339	858
503	328	110	770	1053	773	272	598	77	1087	1187	1121
551	418	225	801	1185	1082	955	557	737	506	464	1070
1240	168	187	843	796	82	867	450	200	764	190	982
1076	967	873	260	695	224	1018	207	1049	277	3	652
792	1014	1213	936	466	621	829	2	400	1085	38	23
1159	570	509	816	973	164	205	1201	112	1078	870	333
364	1212	350	250	102	536	172	629	705	178	381	722
283	1100	823	426	84	122	134	1128	591	361	798	24

1224	1111	179	1143	526	1219	1166	176	834	10	1221	1093
1062	1232	167	587	647	342	490	1041	1114	785	525	1020
473	1233	856	562	423	579	527	427	1194	215	653	242
1237	767	795	1153	862	952	998	644	1081	461	182	232
29	539	1025	1223	434	411	320	932	1204	499	11	546
1112	558	958	1160	486	467	1039	436	1245	1151	421	920
126	692	174	802	581	1066	852	355	1023	180	125	821
1104	129	265	1094	158	1101	824	871	555	1156	106	681
1145	645	803	1208	114	1068	1080	1222	269	996	1074	263
1064	1075	1184	213	699	123	312	1056	1130	175	468	754
380	1063	616	234	1200	768	865	1154	108	1069	631	698
360	1207	397	356	711	540	484	1180	1215	941	1220	596
604	1244	774	916	1050	749	420	169	1029	1242	424	376
1107	1178	820	352	372	118	443	818	701	1136	690	950
1196	93	724	543	1155	476	1034	963	196	978	1086	472
1230	781	610	273	1125	1137	531	120	554	264	969	763
165	407	390	1209	583	49	648	183	331	104	985	130
1126	1152	977	43	917	508	614	992	428	1067	582	983
987	454	497	230	530	285	374	1097	758	308	354	495
664	128	825	441	990	288	951	1131	1008	1004	632	121
752	249	105	682	1195	460	1162	981	746	297	228	700
1116	1095	296	83	1164	1198	528	502	27	762	769	1139
655	863	514	1028	276	811	1177	663	432	12	1002	716
826	96	59	458	1225	1181	136	416	830	1072	937	209
94	986	784	940	146	658	204	976	945	918	185	1012
643	611	1169	1142	1079	1173	793	457	1191	393	212	635
239	984	261	231	1010	1186	1203	576	1124	1140	404	738
991	33	1163	358	586	1098	324	968	1083	642	972	76
1013	1182	1092	751	16	88	817	957	569	135	1117	535
622	670	1009	291	832	329	28	979	72	236	1144	462
298	79	626	772	221	238	487	559	1236	556	946	633
702	69	235	1005	109	140	319	452	1054	155	1019	304
485	756	512	74	398	216	1210	925	1171	911	1243	842
997	478	1239	293	921	429	160	933	1088	431	145	1229
661	295	322	35	1055	1043	1016	201	1150	637	620	482
369	790	253	640	719	491	1061	1042	237	113	914	1122
64	348	171	727	233	254	522	206	1174	81	115	511
1193	928	1218	346	931	405	1	613	1214	394	709	592
359	307	553	708	850	163	999	166	39	1011	370	949
710	1113	1157	935	1022	860	147	78	948	198	1211	177
330	19	771	211	684	993	677	980	1206	927	723	804
1065	970	1044	1099	729	523	442	908	1217	1165	159	9
828	995	150	244	1231	15	325	137	1189	1205	844	691
922	964	564	89	281	243	671	218	657	1000	1172	939
1059	1032	683	649	783	989	757	606	385	965	470	954
389	195	959	806	1060	615	676	630	219	1170	1052	294
257	864	869	371	715	301	840	318	1118	1158	268	1027
627	435	1103	47	1015	1175	975	1192	313	1084	1132	529
67	787	953	1167	1135	924	919	1073	1031	1058	853	271
384	471	812	947	18	1161	765	567	934	217	66	759
545	475	8	1147	505	1001	1045	599	1141	1235	1176	584
284	1216	929	704	1089	943	501	717	445	966	693	605
259	1168	733	743	747	1138	550	336	343	53	814	1133
144	913	439	1051	872	1127	909	1024	459	915	17	504
944	1096	1046	73	726	1105	1026	956	1188	718	549	1146
48	910	494	189	222	1017	827	334	612	585	745	678
208	580	974	938	1123	396	1190	1030	367	55	337	366
962	246	21	4	777	1179	1149	1227	220	930	1226	1021
800	659	1183	731	202	1197	403	1090	988	533	379	1057
1106	1071	515	282	451	1038	574	1033	1199	226	1108	923
1007	280	111	544	406	61	748	1274	402	753	181	1374
143	306	868	1264	6	193	489	618	1351	667	1346	1375
1267	32	732	687	286	688	507	363	1308	859	95	575
855	1376	345	1358	65	1361	1257	565	44	1371	214	703
447	36	561	1329	799	203	625	720	278	1287	845	210
444	1280	1327	1303	414	1268	1273	846	477	446	1253	578
251	1377	822	229	1328	1295	847	419	1320	837	1378	831
1373	1362	538	1379	662	1254	1334	778	1331	58	45	602
87	287	854	839	1292	780	1322	309	619	401	279	588
849	1350	332	791	351	156	1247	1355	639	157	132	1380
1282	1270	1381	641	375	730	1300	498	388	594	519	1382
689	1313	1258	542	595	162	1383	493	275	252	5	1367
1348	368	776	227	1319	675	413	1338	20	1318	188	338
516	548	568	590	1294	415	1343	609	417	1356	54	1315
223	1323	1316	1357	685	1246	1384	433	41	815	1360	1296
755	1345	316	808	650	310	449	1284	623	1347	1302	101

1364	481	1293	142	170	517	680	1314	1324	1281	1385	317
1321	572	833	1261	141	62	697	305	1330	51	173	1312
669	148	1256	656	92	391	161	382	1251	600	665	805
686	1342	589	1369	534	349	1249	838	1332	1336	521	1272
387	1262	1340	1301	70	552	192	1386	1311	603	1277	766
440	789	638	292	37	1279	1291	1372	725	1387	247	851
1299	86	1388	674	848	1310	1359	57	138	1286	124	1285
1266	1248	809	1252	340	289	1389	775	488	71	694	607
31	1349	628	1263	1390	1255	365	1341	300	1275	262	1309
1250	761	25	50	85	408	256	577	835	841	742	1370
707	712	666	1283	1354	455	1317	386	430	1269	492	1339
14	660	1288	314	1368	1306	151	1353	807	90	1265	788
608	510	1391	1335	327	1304	741	7	1365	197	245	861
1297	1259	80	679	1333	75	1326	1352	779	26	199	270
547	537	636	1289	740	323	326	395	1298	60	1344	302
1392	1337	1271	593	127	63	184	13	513	103	1307	469
1290	1393	1260	1366	1305	1325	1278	1276	107	241	315	240
1363											

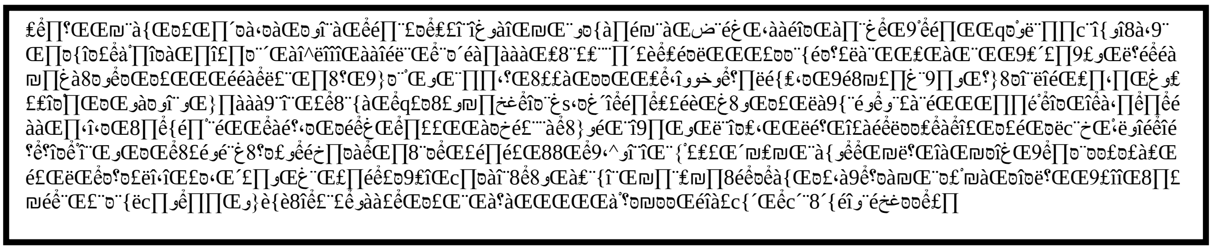
Clé de chiffrement de la deuxième version chiffrée de Texte1 (Taille$_{Clé}$ = 1,8704 K octets)

7	36	33	79	39	24	81	16	5	73	3	1
20	88	30	44	56	41	11	62	65	47	98	26
814	806	645	389	313	988	737	838	346	946	243	890
300	123	778	333	207	229	663	571	127	816	821	355
1220	1344	1025	1270	1147	1345	1020	1218	1338	1037	1085	1041
1016	1226	1114	1073	1346	1190	1195	1253	1227	90	40	70
15	95	6	53	87	34	91	25	82	76	45	13
93	14	55	28	37	17	72	66	85	61	22	64
21	51	8	77	35	78	86	38	10	42	18	99
31	60	2	27	54	92	94	80	19	71	69	29
67	46	84	89	12	9	96	4	57	58	63	49
59	75	50	23	83	48	526	275	178	752	965	810
453	272	436	898	463	245	321	586	113	506	208	419
708	183	640	451	376	195	372	414	942	895	238	302
443	631	474	712	434	917	277	392	187	467	607	557
845	647	688	188	52	97	630	680	865	449	583	164
262	105	634	317	639	119	928	464	714	802	797	932
823	822	953	684	529	694	270	991	482	154	956	978
440	135	197	670	280	351	987	248	881	779	933	327
606	214	662	695	292	456	909	608	158	285	491	877
311	771	395	891	938	803	563	612	561	963	715	501
447	153	498	134	792	220	215	811	874	427	283	418
538	68	32	43	74	687	345	661	924	284	486	359
839	495	115	893	732	103	307	442	305	675	600	136
157	603	106	417	122	918	441	785	930	982	181	466
255	758	765	794	817	271	854	597	494	900	219	198
484	315	194	218	923	382	667	309	901	408	120	558
204	772	336	761	470	266	519	996	240	582	795	969
168	142	541	559	540	853	564	578	936	916	788	882
274	861	527	365	651	367	707	665	162	435	469	929
589	979	939	290	753	755	776	584	656	269	391	416
860	700	786	514	344	566	492	319	993	139	384	790
256	475	252	437	244	145	276	544	374	598	373	828
258	652	870	685	784	318	801	927	353	109	180	324
689	858	910	108	940	508	525	137	749	697	975	296
489	650	503	329	101	570	326	850	716	581	246	627
403	782	375	556	985	949	126	907	422	179	282	221
161	944	981	604	517	349	212	783	138	906	835	873
594	545	880	655	462	967	539	457	203	528	717	952
341	316	366	813	591	836	299	769	926	951	547	904
848	844	840	800	744	576	819	995	412	413	147	588
812	424	356	404	554	869	619	186	390	150	692	825
740	852	343	601	831	431	312	989	535	481	377	702
764	286	855	682	976	261	515	768	750	254	696	883
654	169	304	618	889	727	710	805	362	799	360	621
543	530	288	428	379	118	587	531	423	833	426	210
342	241	905	787	572	439	560	954	748	653	402	773
394	980	448	649	223	488	760	415	368	959	611	913
287	493	569	729	253	767	132	322	536	479	553	476
648	410	815	964	562	458	323	217	385	830	144	234
173	617	643	143	660	628	461	551	473	193	337	257
644	698	673	293	104	872	757	334	843	826	510	452
289	107	573	213	251	459	516	512	110	998	438	242
974	971	314	471	669	550	620	465	148	580	593	777

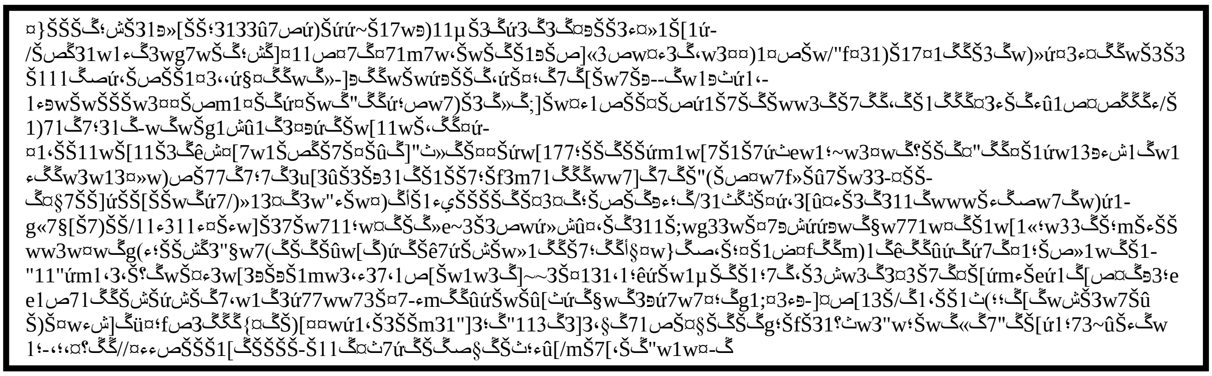
192	990	155	230	614	504	638	983	915	775	842	149
236	363	759	999	140	745	851	885	736	766	941	294
871	867	994	152	914	407	396	555	690	886	585	896
369	613	731	966	249	774	460	170	505	331	970	721
846	807	892	295	577	298	804	961	191	796	948	159
590	791	674	117	742	720	738	490	405	664	599	184
432	166	509	762	672	986	615	542	306	165	130	622
455	728	683	445	820	658	957	605	177	279	176	335
859	129	364	706	202	879	809	502	348	383	657	477
281	947	832	671	857	868	546	595	112	268	233	724
741	228	713	878	908	167	370	887	925	100	325	704
200	211	499	681	626	574	677	763	433	701	624	446
637	354	111	960	841	977	125	483	444	478	972	711
131	808	430	133	387	380	163	468	911	310	897	864
160	513	227	141	945	518	352	718	984	524	487	425
116	781	411	278	636	876	128	320	532	330	629	725
834	454	250	114	291	668	520	534	849	770	185	358
623	973	709	409	303	146	579	259	955	429	175	747
397	950	659	679	239	856	642	610	609	568	301	958
496	297	533	121	937	480	201	730	552	818	500	225
400	182	521	328	894	922	935	522	686	888	189	931
199	511	754	406	124	232	102	332	920	361	398	693
224	386	399	592	350	739	567	703	265	226	919	263
899	222	472	746	156	635	378	507	206	171	273	827
401	866	565	548	338	789	421	992	209	151	676	641
485	884	699	260	393	523	902	388	726	997	968	190
381	751	633	231	264	756	824	357	420	625	450	743
196	216	602	371	237	962	174	235	735	875	837	172
863	829	666	247	719	912	793	339	847	943	646	734
934	347	798	340	205	575	733	267	537	903	678	308
616	862	596	632	549	691	705	723	497	921	780	722
1229	1311	1126	1074	1105	1179	1155	1223	1023	1161	1175	1129
1184	1134	1305	1319	1088	1082	1347	1079	1281	1152	1144	1149
1288	1176	1186	1289	1254	1283	1348	1017	1159	1189	1349	1217
1131	1039	1350	1272	1064	1192	1188	1228	1115	1095	1264	1310
1058	1201	1198	1022	1014	1043	1108	1071	1351	1352	1027	1165
1151	1059	1028	1353	1141	1200	1143	1354	1247	1100	1029	1112
1005	1355	1257	1083	1194	1046	1356	1035	1024	1307	1101	1091
1278	1287	1092	1139	1120	1250	1007	1246	1225	1116	1051	1357
1244	1086	1209	1096	1090	1284	1358	1106	1127	1318	1309	1290
1193	1216	1076	1052	1259	1359	1117	1360	1361	1239	1047	1034
1003	1057	1277	1111	1093	1332	1301	1009	1183	1328	1178	1061
1099	1265	1215	1240	1362	1213	1269	1298	1049	1312	1363	1171
1263	1252	1364	1197	1077	1243	1365	1366	1145	1214	1142	1018
1315	1123	1146	1107	1060	1367	1113	1006	1160	1337	1249	1326
1011	1279	1368	1208	1196	1275	1056	1170	1110	1234	1335	1261
1294	1069	1031	1203	1063	1267	1258	1300	1202	1157	1241	1030
1150	1280	1206	1369	1181	1180	1021	1370	1158	1121	1097	1205
1128	1012	1293	1089	1341	1237	1187	1317	1102	1010	1068	1304
1371	1291	1372	1148	1153	1065	1256	1303	1306	1124	1295	1053
1013	1162	1268	1044	1373	1078	1032	1166	1177	1137	1026	1255
1374	1375	1133	1236	1191	1172	1048	1245	1282	1119	1296	1062
1130	1308	1376	1207	1299	1036	1377	1333	1136	1199	1378	1379
1292	1164	1045	1248	1343	1380	1314	1235	1321	1066	1232	1182
1271	1336	1320	1260	1251	1381	1382	1168	1286	1383	1384	1316
1322	1385	1042	1262	1173	1212	1038	1386	1040	1274	1154	1327
1273	1185	1387	1313	1210	1388	1019	1132	1033	1001	1156	1389
1122	1070	1098	1211	1055	1329	1323	1285	1072	1054	1325	1004
1302	1118	1231	1219	1204	1087	1222	1221	1125	1080	1342	1224
1015	1340	1390	1050	1330	1233	1094	1081	1067	1104	1174	1297
1391	1331	1075	1392	1339	1238	1230	1135	1324	1276	1163	1334
1103	1084	1002	1138	1167	1109	1242	1140	1008	1266	1169	1000
1393											

استخدم علم التشفير منذ القدم لارسال الرسائل المخفيه لاغراض سياسية وعسكرية في الحضارة الفرعونية والدولة الرومانية .لكن التشفير كعلم مؤسس وومنظم يدين لعالم التشفير الذي يزخر بهامات شامخة امتدت عبر تاريخ الحضارة وأسهمت في بناء هذا العِلم الشيق وهو (أبي يوسف يعقوب الكِندي).
إن الدافع لإخفاء المعلومة إذن كان عاملاً أساسياً في حسم الصراعات السياسية والعسكرية على مر تاريخ. وهو ما يفسر الأهمية القصوى التي طالما تمتعت بها فنون التشفير عبر العصور. الرغبة في إخفاء المعلومة أظهرت تقنيات شيقة تستحق الدراسة. وحيث أن تقنيات التشفير قد شهدت ولادة جديدة بملامح مختلفة كلياً بعد انضوائها تحت تطبيقات الحاسبات الإلكترونية والتي شهدت تطوراً باهراً بسبب عاملين أساسيين. أولهما مثلته الصراعات المسلحة التي شهدها العالم في القرن الأخير . العامل الثاني الذي قفز بعلوم الترميز لآفاق جديدة كان التطور الكبير في علم الحوسبة الإلكترونية. فالحواسيب لم تقدم فقط أنماطاً جديدة من تقنيات التشفير والترميز; لكنها عقدّت الأمر أكثر بقدرتها المتنامية على كسر الشفرات الصعبة وهو ما وضع مطوري الشفرات في تحدٍ دائم. تطور الحواسيب تضافر مع الانفتاح في الاتصالات ليقدما الخصوصية كخدمة مطلوبة على الصعيد الفردي بعدما كان الأمر مقصوراً في الماضي على المراسلات الرسمية أو السرية بطبيعة الحال.

(a)



(b)



(c)

Figure V.4. (a) *Donnée originale Texte2*, ***(b)*** *Première version chiffrée*, ***(c)*** *Deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Texte2 (Taille$_{Clé}$ = 1,8704 K octets)

8	2	4	7	6	9	1	5	811	910	899	896
878	885	622	892	767	914	357	901	895	297	771	211
70	851	189	890	897	694	880	913	662	78	908	900
906	210	170	889	879	904	833	407	208	300	498	306
259	891	871	858	469	698	543	87	903	368	52	884
266	835	642	886	92	909	905	882	883	405	527	371
915	658	12	893	113	172	457	898	888	633	887	242
877	881	414	376	911	556	912	907	902	163	648	281
894	876	949	701	174	1155	643	192	649	213	466	1103
514	1076	1022	660	624	3	632	430	623	944	925	770
785	796	67	1194	230	707	799	1177	1218	245	508	1007
179	484	1209	1223	549	1118	1037	1167	740	73	1160	248
193	1088	287	1200	304	131	175	544	511	428	1173	176
1176	931	983	815	951	370	32	955	826	53	945	754
168	863	1009	1219	692	532	459	273	1018	356	730	255
636	604	539	11	984	79	930	817	383	581	829	1148
1133	764	961	1068	1127	1140	1225	630	956	403	1143	1235
822	601	755	1217	535	1073	960	1137	203	325	545	1036
745	1112	971	1097	1168	1232	223	437	805	1062	1204	321
272	374	1080	260	748	86	1045	434	429	1157	675	994
145	1142	587	585	1229	1069	1044	267	386	471	1163	147
1102	1182	747	593	798	1054	1019	82	981	978	348	1066
928	669	1192	1065	1038	311	1196	943	1141	597	610	63
741	1085	1075	234	1063	962	850	682	1156	401	843	670
81	1058	84	1000	151	525	60	1110	225	501	969	552
341	446	486	985	1093	580	372	1187	998	517	65	778

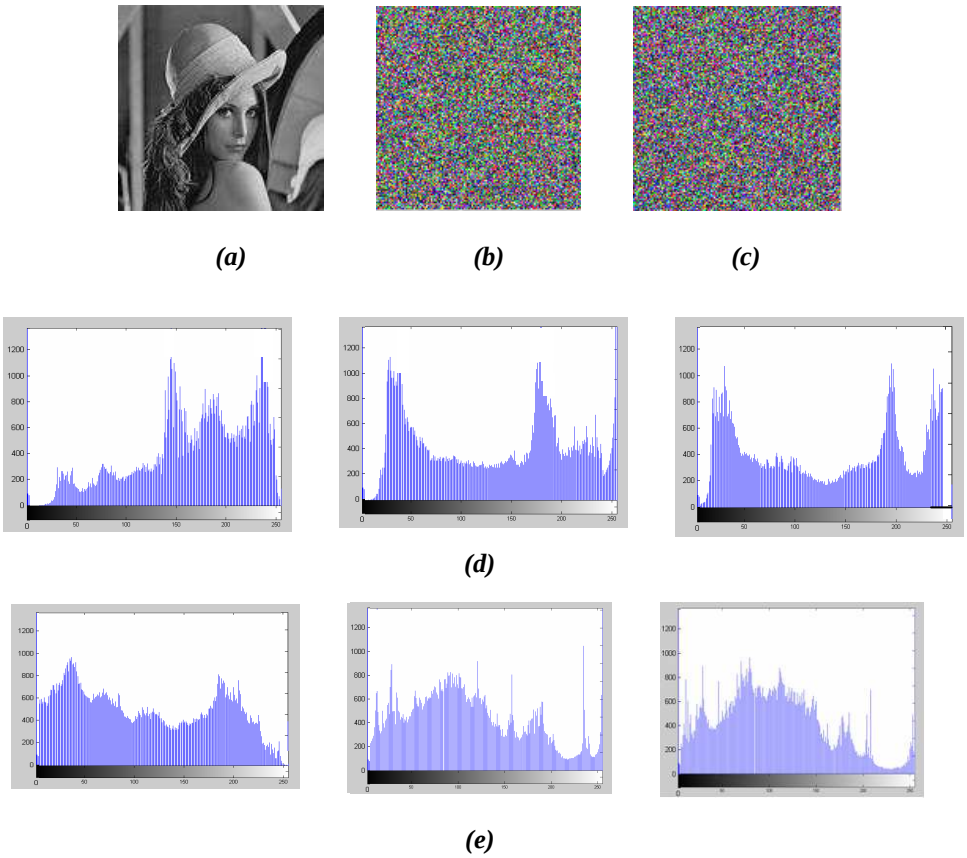
606	137	97	938	180	952	1230	90	35	1067	1084	1083
1014	463	237	814	650	319	935	379	315	309	46	676
1139	1060	1238	240	1152	645	689	846	221	1003	251	116
635	1215	182	28	467	460	171	495	377	674	80	1228
1027	392	452	54	502	957	1064	1079	228	733	932	359
609	997	647	834	1149	1004	751	166	923	830	1166	976
980	1198	1116	975	578	631	75	253	235	1095	1224	860
965	449	1109	43	979	150	763	202	271	252	394	59
516	1205	239	838	874	247	353	685	704	169	83	177
1012	389	114	39	1074	659	924	982	104	565	783	839
705	167	1222	987	656	292	57	93	194	1056	1165	282
861	500	526	564	34	411	922	1189	566	445	506	99
1119	793	257	637	398	1087	1122	1185	105	582	51	494
490	1024	199	207	696	782	596	521	412	138	50	1233
1030	993	1191	258	653	1023	1105	312	757	492	712	419
173	1055	406	948	21	713	119	378	402	929	1186	574
19	654	286	1098	327	246	852	1017	1039	183	592	936
1082	954	332	714	781	328	666	85	518	620	787	1033
20	920	347	750	546	756	1034	1181	1031	62	1035	497
22	522	519	346	1051	591	803	373	1239	1015	927	37
382	966	586	218	140	1124	410	1183	74	595	1180	942
841	1170	872	1125	742	1243	845	316	1104	106	554	159
802	967	1099	710	344	158	196	939	1195	1091	146	614
569	1002	562	1226	641	279	567	416	1178	100	1132	236
823	49	809	296	360	254	575	1013	1048	313	441	801
206	992	1193	157	48	723	1234	280	184	768	615	415
691	824	1213	762	743	753	187	233	404	262	479	1016
422	365	481	1171	661	837	323	963	686	772	605	1227
599	572	777	1072	443	717	1244	1135	1131	735	448	1047
109	657	941	1078	1071	531	865	794	1032	25	107	1100
238	621	120	334	71	795	1077	1001	1159	626	1161	451
342	23	699	1040	919	1145	510	336	715	274	739	468
1231	1172	1237	831	1150	1144	423	14	832	1188	31	1162
68	918	385	338	766	867	1050	528	216	305	577	524
972	709	1081	222	639	399	72	1212	953	1216	124	112
432	1203	1201	126	570	1146	1211	101	426	125	1136	178
209	706	1005	1174	1086	921	926	33	917	358	1011	161
977	450	42	1057	688	728	1111	847	219	456	322	970
197	999	1199	720	366	1175	201	627	1130	320	536	290
1236	995	229	868	478	1214	330	589	1008	625	355	690
968	310	973	337	488	1006	672	77	148	746	808	603
608	1070	343	667	553	354	752	989	946	520	568	628
1061	725	693	1089	110	1117	1151	326	629	1108	1092	294
855	759	16	792	1208	651	391	776	58	1164	256	800
263	916	1120	1179	424	579	547	1115	369	160	974	538
15	89	541	934	947	18	480	1049	1029	465	612	191
958	1028	986	1134	774	548	1138	1210	1197	937	1221	806
1096	542	613	1123	433	678	804	726	76	933	1090	571
1158	1043	959	1094	1041	10	1052	1106	1020	1169	724	1241
862	990	988	269	461	644	711	1153	421	602	677	684
1107	594	1206	1242	164	853	1021	475	277	1113	214	1025
1053	440	491	573	1207	1059	683	134	139	1046	318	864
285	1121	560	1026	1154	397	996	190	509	1190	350	702
1128	417	1042	438	950	673	1010	1114	515	991	665	810
91	869	964	493	188	395	108	844	152	769	335	827
1101	1129	88	435	64	472	1147	807	551	56	1126	719
1202	791	857	136	779	217	1240	444	1184	499	409	1220
668	224	940	299	1343	697	1344	616	442	95	1336	1334
1286	204	646	836	220	489	301	700	1313	288	820	561
773	1297	1345	156	607	473	476	1259	721	133	98	840
1346	351	464	559	1347	384	1348	1304	375	1349	563	1268
436	873	652	439	142	731	1264	314	111	681	393	611
784	1302	558	1270	1350	205	1351	165	13	303	361	1269
1352	734	618	381	732	363	1353	1354	664	557	380	1294
513	94	530	727	1291	244	1355	1338	367	425	454	758
333	1356	1300	1357	1248	339	1342	1341	588	1282	250	317
760	122	1308	512	1358	69	1299	129	102	1251	1359	534
331	788	1298	396	1360	268	264	103	427	482	1325	1307
523	1361	329	1324	1362	1363	1290	1364	859	455	155	663
1340	1333	819	261	1365	1366	1329	1328	786	293	284	1367
154	1331	270	1265	849	132	27	135	729	130	749	505
1368	231	638	1369	390	1370	345	540	1371	1278	1293	276
1372	1250	308	737	17	1373	241	298	362	1262	198	127
447	153	474	1272	1252	825	349	121	1315	1303	215	47
1337	1305	249	1260	186	775	671	1320	387	1374	291	870
289	1375	400	41	227	1289	144	30	55	1339	744	1316

1296	128	1319	504	1376	1245	1377	529	1378	848	431	485
1255	418	1254	708	352	1267	212	232	640	1246	1314	458
1379	722	1327	1380	634	181	118	797	123	590	1280	679
1381	1326	1258	195	1271	507	38	1382	66	1279	453	1256
1309	1383	738	483	340	655	736	533	1312	1384	420	364
842	149	780	1263	1332	1275	1274	283	816	1284	1330	1266
866	1385	117	598	818	703	200	716	408	1285	854	576
295	1386	1292	115	695	1249	1318	40	185	302	1306	1283
1288	45	487	1323	875	143	600	477	583	1281	1253	550
1387	278	619	828	324	1257	813	812	24	790	503	496
36	1388	1389	243	96	1301	1321	718	555	789	1247	388
26	1261	821	1276	162	761	1322	687	275	765	1390	1287
617	1310	141	265	44	1277	226	1391	1392	1311	29	1393
1317	856	584	1295	680	537	462	307	413	1273	61	1335
470											

Clé de chiffrement de la deuxième version chiffrée de Texte2 ($Taille_{Clé}$ = 1,8704 K octets)

681	257	54	272	127	416	212	193	126	445	404	584
108	735	917	617	694	270	386	241	702	562	898	175
381	459	904	737	596	689	130	228	326	836	460	289
37	958	638	703	356	817	741	466	971	858	1	909
688	524	53	13	448	624	634	199	369	807	116	686
102	121	204	490	397	187	670	650	522	163	256	950
868	720	153	510	660	580	68	427	698	546	523	900
621	306	312	331	721	488	314	243	350	313	452	325
844	986	82	499	494	48	12	991	526	453	479	910
978	56	409	607	172	240	620	627	391	779	15	641
174	931	71	90	55	265	963	947	981	60	84	250
95	39	977	513	748	220	235	516	507	916	401	190
360	213	362	725	447	333	105	491	857	78	881	383
89	934	705	999	123	426	231	293	323	91	943	557
803	846	324	945	791	718	826	903	396	258	839	602
713	83	859	38	639	411	400	201	359	545	487	92
138	766	879	799	346	558	928	864	49	961	365	519
149	146	371	18	281	158	430	684	277	885	754	58
374	435	269	318	358	850	335	953	567	443	970	899
169	719	291	789	809	613	432	595	275	882	247	347
62	282	851	685	97	775	151	814	601	384	439	771
980	343	552	598	106	853	541	192	612	993	687	154
271	461	496	125	708	776	948	744	939	375	232	952
286	927	675	671	483	575	373	455	155	305	421	761
731	279	869	288	642	933	788	367	122	676	46	997
35	31	830	902	873	480	736	402	751	477	336	654
450	704	696	295	236	534	244	944	988	629	230	861
341	100	433	906	786	390	750	905	307	42	351	659
714	301	395	34	205	549	77	438	215	285	454	662
871	883	949	884	165	573	935	398	728	112	888	5
985	304	412	440	505	514	219	292	920	131	965	72
349	509	833	287	157	707	308	47	472	810	492	144
339	489	734	139	233	223	474	261	533	209	987	656
912	994	475	911	968	98	772	506	553	214	610	469
132	202	159	568	925	983	938	237	224	669	777	693
538	964	891	746	431	464	604	227	722	756	773	554
50	527	142	252	529	334	171	712	465	437	515	255
874	57	758	540	161	760	733	503	792	27	486	532
822	385	462	234	407	299	583	210	446	88	544	757
436	319	700	109	478	614	393	732	246	599	556	316
682	753	767	70	586	423	796	361	570	793	177	913
73	8	414	30	372	81	709	429	456	79	64	936
508	878	578	63	780	329	442	135	110	457	597	226
870	512	876	458	812	866	266	942	765	768	302	26
330	69	76	537	724	907	3	946	608	500	561	872
4	424	23	535	865	181	120	493	539	315	160	93
589	588	611	701	200	216	661	531	419	115	677	273
399	982	380	188	890	382	24	377	749	655	194	922
673	87	653	823	267	740	85	221	44	582	310	189
813	956	834	695	995	366	410	889	600	353	140	425
590	783	838	908	672	185	819	806	918	501	892	816
955	229	930	128	778	536	542	413	560	484	559	504
975	99	941	186	867	420	984	476	518	66	606	622
405	976	141	937	592	797	378	631	485	211	33	566
666	521	723	28	635	297	957	441	7	683	251	992
626	511	101	591	21	665	716	636	996	551	548	274
585	363	979	332	96	117	609	249	129	923	644	710

637 517 738 51 43 357 969 563 222 203 322 547
470 860 344 798 355 863 248 628 921 845 841 184
847 502 65 565 134 886 848 951 940 471 594 406
774 818 663 45 468 14 645 972 303 959 770 337
855 260 124 752 824 623 619 574 451 781 962 9
825 368 498 640 196 428 877 473 166 197 463 354
415 764 118 815 587 739 276 183 67 321 94 298
564 577 136 785 897 801 444 36 808 364 895 657
820 759 179 311 495 658 147 742 376 691 16 579
167 896 370 434 625 804 467 571 805 745 387 379
664 6 2 690 652 795 967 300 422 726 827 837
572 191 831 61 150 901 225 706 924 327 729 593
320 253 17 152 543 667 954 715 254 893 697 137
263 180 11 143 74 990 19 278 875 763 119 345
605 264 408 22 576 699 802 497 309 854 929 86
680 678 842 168 482 113 394 649 679 727 296 782
835 787 615 618 338 389 25 743 692 52 550 843
114 919 207 828 80 40 284 178 730 10 238 966
242 569 794 832 103 755 392 182 75 481 849 647
821 259 974 104 294 880 418 973 603 674 998 530
616 59 32 262 348 651 195 20 133 340 894 632
156 388 403 643 829 176 581 960 29 245 198 217
762 711 717 41 239 862 747 668 148 914 555 206
449 926 280 352 417 111 932 852 342 784 633 173
887 989 162 107 218 170 328 646 208 790 915 528
290 811 769 630 648 317 840 520 283 145 164 268
525 856 800 1319 1015 1163 1032 1115 1334 1035 1004 1255
1286 1101 1309 1186 1325 1012 1335 1304 1336 1136 1337 1107
1172 1113 1249 1190 1227 1058 1010 1251 1111 1023 1338 1127
1302 1114 1033 1100 1228 1128 1112 1031 1020 1299 1109 1311
1339 1226 1081 1202 1076 1054 1295 1340 1069 1341 1198 1216
1342 1183 1343 1002 1148 1070 1176 1328 1303 1003 1121 1029
1277 1229 1089 1230 1256 1091 1175 1060 1042 1119 1182 1188
1074 1050 1132 1224 1333 1170 1078 1294 1168 1344 1149 1320
1239 1162 1203 1223 1052 1310 1269 1233 1210 1083 1265 1307
1275 1298 1045 1326 1345 1346 1272 1048 1317 1347 1261 1030
1263 1049 1285 1165 1348 1276 1349 1350 1027 1201 1235 1262
1189 1351 1313 1005 1063 1017 1195 1000 1150 1352 1016 1353
1062 1305 1130 1024 1354 1221 1073 1355 1179 1126 1356 1205
1161 1244 1240 1001 1129 1022 1197 1284 1087 1080 1006 1291
1137 1147 1152 1055 1181 1268 1252 1250 1327 1053 1160 1044
1134 1357 1072 1037 1151 1040 1225 1093 1008 1358 1056 1192
1246 1359 1271 1360 1047 1315 1090 1361 1013 1156 1362 1204
1157 1041 1082 1096 1184 1105 1123 1329 1363 1281 1097 1118
1241 1215 1364 1146 1288 1258 1365 1131 1366 1034 1092 1297
1300 1098 1367 1280 1368 1248 1018 1142 1245 1120 1177 1211
1267 1369 1173 1370 1371 1057 1372 1122 1214 1159 1237 1138
1021 1038 1166 1217 1064 1180 1208 1283 1106 1065 1185 1373
1278 1314 1374 1200 1322 1071 1220 1219 1316 1145 1169 1222
1375 1293 1067 1059 1104 1019 1260 1242 1257 1133 1376 1125
1043 1377 1167 1378 1379 1308 1140 1079 1085 1077 1232 1103
1380 1290 1381 1154 1331 1124 1266 1199 1253 1116 1270 1209
1158 1036 1102 1046 1382 1306 1095 1007 1088 1231 1108 1193
1238 1206 1039 1236 1187 1274 1383 1282 1094 1171 1384 1075
1264 1084 1254 1196 1259 1164 1155 1025 1385 1178 1386 1068
1086 1387 1318 1243 1207 1026 1321 1324 1323 1028 1213 1061
1218 1191 1212 1330 1388 1066 1332 1234 1117 1051 1153 1301
1273 1099 1389 1390 1144 1139 1143 1009 1247 1287 1014 1135
1391 1279 1174 1141 1392 1312 1110 1292 1296 1289 1011 1194
1393



(a) ***(b)*** ***(c)***

(d)

(e)

Figure V.5. (a) *Image test Lena,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Lena ($Taille_{Clé}$ = 0,9375 K octets) :

369	423	529	171	397	412	474	15	689	64	112	390
267	21	118	304	6	28	574	465	356	338	717	638
339	110	243	458	508	450	361	608	635	115	593	749
151	742	575	716	428	109	486	584	294	722	521	141
87	695	373	225	452	602	679	13	208	85	468	634
709	298	70	297	514	601	439	359	320	251	222	292
319	765	456	614	142	645	625	288	383	217	125	476
396	718	274	236	729	483	567	114	487	492	303	542
426	628	675	713	296	505	370	730	409	395	764	641
669	460	337	264	230	204	623	358	293	418	150	520
134	328	469	762	440	533	651	551	535	286	438	462
103	355	347	63	158	16	561	205	410	643	703	432
519	693	247	620	346	40	736	366	129	652	164	327
371	506	363	154	241	447	434	624	585	163	748	376
380	144	407	59	143	667	252	10	536	548	766	360
335	455	311	497	200	466	249	751	35	119	165	341
36	741	398	448	656	18	65	233	734	17	531	512
595	660	104	586	81	435	299	101	92	309	454	436
710	408	636	441	278	662	463	411	76	668	321	280
353	648	1	139	473	185	674	732	155	569	140	579
352	739	698	306	253	577	248	735	515	342	285	61
219	138	568	705	658	427	511	532	210	357	307	153
633	711	98	433	619	691	60	20	137	733	485	324
659	419	719	684	269	392	754	68	259	284	467	295
48	300	224	541	600	715	630	613	172	66	425	558
604	38	692	34	557	263	258	384	51	317	375	111
325	146	701	472	388	607	289	94	644	377	745	580
500	598	496	240	443	594	364	554	74	637	187	712
655	747	576	145	47	501	417	490	622	162	491	626
578	699	750	14	128	493	282	226	459	351	589	209
582	545	740	596	147	275	121	362	470	53	193	291
333	290	179	523	527	42	30	402	738	525	113	374
664	646	507	609	685	756	189	700	148	260	654	464

191	350	378	176	257	538	599	603	11	702	372	571
237	12	96	170	192	653	88	416	540	25	180	522
642	166	72	246	194	707	680	33	639	159	215	75
728	227	516	681	344	90	99	559	127	513	265	7
23	666	704	406	281	135	108	499	723	126	287	5
169	152	678	168	83	564	250	256	24	266	657	555
69	708	345	494	271	238	726	393	534	132	385	587
686	647	368	203	3	481	482	332	199	167	687	560
539	495	743	44	583	91	610	690	336	31	261	403
379	255	725	498	323	386	198	572	220	484	73	322
677	606	340	272	343	58	503	381	26	421	517	761
79	676	334	348	760	49	663	235	597	566	313	627
39	414	618	518	479	55	752	186	694	563	649	175
223	133	549	420	277	510	727	305	391	228	632	404
196	206	84	52	184	755	212	581	763	429	107	697
724	590	77	737	445	71	682	502	509	120	254	413
758	640	149	616	43	195	242	239	211	611	453	757
218	86	19	310	387	2	105	117	424	244	504	116
449	591	8	229	605	399	629	182	57	461	314	588
177	234	190	102	106	714	122	617	612	216	673	312
431	326	621	80	382	721	93	45	156	665	553	767
475	174	329	394	415	547	188	331	543	365	308	46
67	670	330	157	354	302	389	480	367	232	768	27
207	422	130	405	316	318	451	82	29	37	100	530
315	552	161	650	592	537	550	124	173	56	565	631
41	471	131	201	245	54	202	444	556	544	720	270
32	688	759	442	279	478	136	197	123	273	671	437
262	457	672	268	349	95	50	488	706	546	183	221
178	160	528	696	62	9	214	615	573	489	430	97
744	22	753	524	446	213	746	731	181	231	283	661
570	401	276	78	400	4	562	526	683	301	89	477

Clé de chiffrement de la deuxième version chiffrée de Lena (Taille$_{Clé}$ = 0,9375 K octets) :

156	453	55	623	217	430	383	574	180	389	257	147
405	685	168	355	721	754	738	227	187	474	143	551
127	578	506	528	477	713	723	319	702	417	384	67
512	575	469	301	86	237	703	204	366	2	268	247
340	452	642	21	12	343	128	214	390	220	262	76
495	281	286	66	397	169	445	170	219	24	48	531
261	338	478	282	635	44	191	209	735	323	410	514
255	234	43	54	516	291	97	344	720	695	129	332
461	117	522	270	89	537	298	161	515	346	758	737
253	35	75	739	240	719	131	27	548	98	455	195
78	683	710	178	595	387	400	403	633	353	350	250
687	37	462	235	119	53	200	385	192	530	179	419
392	488	61	457	45	125	1	576	58	312	552	509
546	706	276	241	617	73	585	681	751	753	517	677
399	91	473	557	701	26	302	637	632	32	565	755
135	539	47	266	110	505	17	631	647	727	303	577
427	411	493	23	743	458	259	122	285	114	489	544
182	72	582	194	264	120	439	279	669	718	292	207
101	407	446	158	475	536	284	630	215	724	25	256
172	622	650	734	87	638	450	521	416	665	762	519
700	251	491	335	287	608	742	34	426	104	370	290
7	315	77	480	330	486	183	619	730	19	331	141
599	555	229	202	359	731	757	273	504	308	167	155
230	542	654	304	165	423	360	733	705	662	648	763
28	433	92	664	694	716	175	760	656	190	171	415
263	627	470	51	100	589	4	707	311	148	88	696
16	466	498	590	111	224	363	468	239	408	162	621
36	333	373	626	561	113	341	70	717	511	379	3
371	109	508	378	463	673	693	221	367	30	388	8
212	750	748	667	374	33	358	661	289	116	447	674
566	173	364	747	314	401	500	185	328	140	634	198
676	449	527	490	260	106	588	79	672	68	759	502
591	142	443	94	420	581	698	108	471	572	296	188
436	38	395	437	545	614	163	525	605	704	121	160
587	318	294	764	307	615	107	761	337	151	422	604
326	69	412	223	300	213	472	249	678	690	580	658
765	11	57	573	99	124	184	523	402	501	226	479
649	361	137	193	618	181	174	39	248	299	579	484
62	670	345	351	636	382	347	454	41	118	644	675
74	620	745	218	252	610	265	438	280	564	60	362
404	503	271	712	418	569	550	406	236	13	600	603

96	82	547	126	568	369	199	534	602	612	485	596
365	31	441	203	682	49	216	729	85	456	722	154
641	258	254	524	668	483	616	435	598	317	532	549
768	613	46	465	628	680	460	225	297	186	246	645
321	133	459	520	196	584	145	413	59	692	231	375
688	102	444	210	293	322	643	714	274	242	339	201
749	65	325	767	691	736	726	424	766	601	144	376
349	159	429	130	583	233	18	529	709	646	625	208
269	380	377	64	594	487	451	244	639	305	176	711
20	112	756	653	71	684	464	157	29	563	189	606
553	651	699	232	50	352	138	728	497	467	152	310
103	10	507	327	715	586	245	434	15	14	746	640
205	81	56	541	526	309	177	393	134	663	562	741
725	368	597	391	394	348	272	288	238	660	40	153
136	533	295	396	571	744	84	146	342	425	164	629
123	689	567	52	320	708	166	740	316	275	559	611
481	686	267	657	93	556	197	752	386	5	336	494
139	554	428	90	83	732	313	592	671	243	448	431
518	132	95	496	277	607	535	80	306	22	540	381
570	372	6	334	652	356	211	659	149	543	499	421
228	278	492	513	357	329	679	432	482	510	115	655
9	624	283	63	150	666	354	324	476	697	409	609
440	414	593	105	442	560	558	206	222	42	538	398

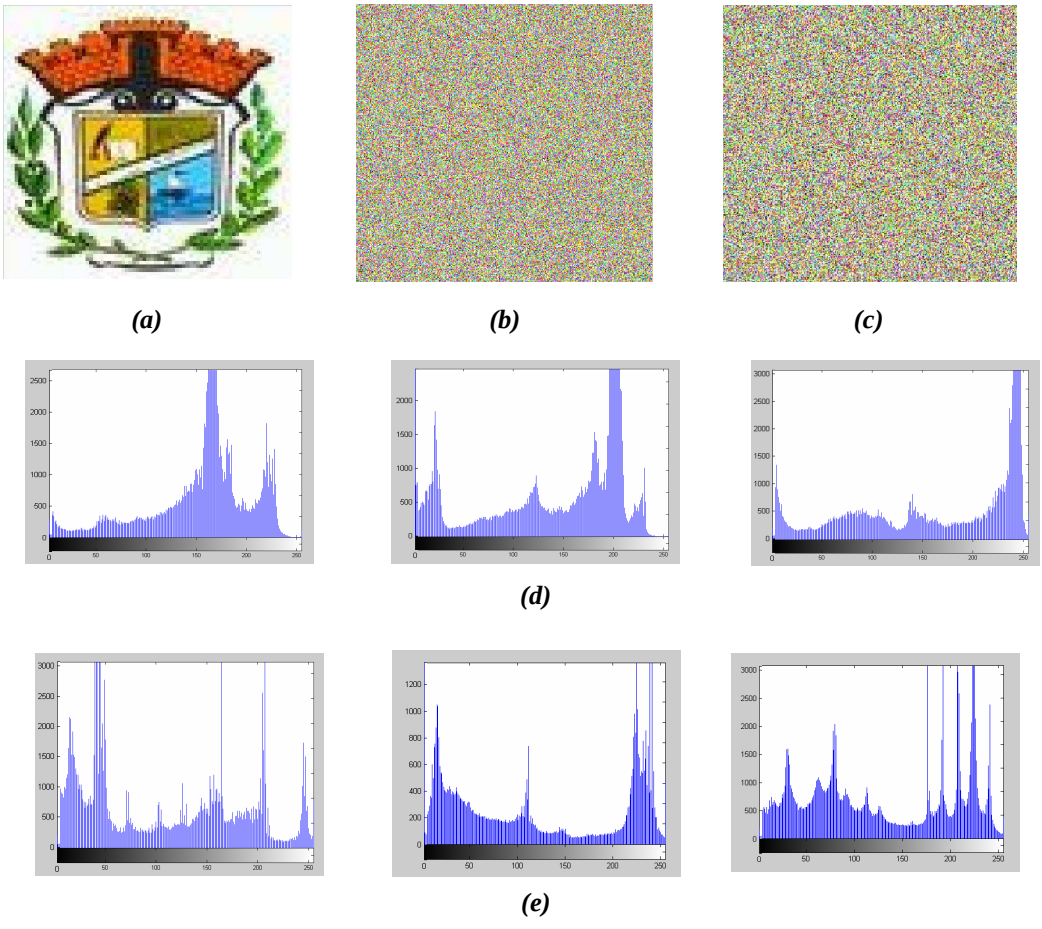


Figure V.6. (a) *Image test Logo,* ***(b)*** *première version chiffrée,* ***(c)*** *deuxième version chiffrée,* ***(d)*** *Histogrammes de la première version chiffrée,* ***(e)*** *Histogrammes de la deuxième version chiffrée.*

Clé de chiffrement de la première version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

65	9	12	62	15	21	94	7	28	11	81	3
61	36	78	70	91	25	56	97	85	4	30	24
83	32	57	40	54	31	63	8	95	26	75	55
19	96	20	38	49	92	47	44	1	98	52	89
35	67	6	72	69	53	74	60	14	43	84	23
22	82	88	73	90	76	39	13	59	33	58	68
41	50	86	46	5	51	27	99	79	10	71	66
87	48	16	2	45	64	42	80	17	34	77	29
37	93	128	625	414	134	459	456	642	446	447	205
635	230	755	387	503	179	646	371	252	744	662	574
575	475	768	734	494	638	389	441	312	571	445	214
367	308	498	643	659	705	175	676	237	530	674	370
527	626	415	330	333	472	600	238	339	137	730	257
262	144	660	139	217	323	739	329	578	420	618	735
463	199	729	207	340	336	409	551	295	296	477	222
468	276	151	258	725	178	764	245	132	366	525	305
196	717	363	484	658	136	342	397	408	198	598	443
589	455	256	239	287	216	181	290	338	111	334	751
345	392	560	697	355	585	418	18	610	162	534	154
467	241	250	694	464	461	614	500	743	170	244	741
404	259	211	645	318	650	593	442	187	282	639	288
648	267	655	220	573	436	552	344	164	225	727	165
226	667	693	416	116	265	536	532	686	522	471	587
316	260	486	514	488	268	601	628	767	365	382	314
528	195	493	430	108	227	212	145	194	753	716	675
547	152	517	374	182	539	356	677	724	760	377	434
546	119	702	204	590	481	394	114	509	395	301	419
473	490	656	728	168	583	538	426	745	518	149	141
131	279	644	140	580	701	325	348	341	285	203	298
651	263	723	235	462	691	622	722	129	437	354	163
335	398	647	122	469	603	553	737	269	511	101	240
110	361	400	402	317	186	733	277	766	142	264	172
399	613	703	588	231	315	106	669	531	380	233	180
319	504	275	113	425	153	372	748	424	281	157	234
249	581	466	200	289	533	411	630	579	223	221	376
679	569	596	188	421	758	519	146	427	243	123	331
671	508	678	206	304	364	273	440	561	324	297	713
670	274	118	576	715	611	752	620	379	489	369	750
343	278	103	479	621	485	224	526	453	229	604	352
765	653	413	714	499	169	595	762	746	465	410	403
537	100	609	327	117	683	617	570	740	492	373	417
310	505	155	543	246	390	689	102	747	710	594	616
497	381	406	756	563	548	487	632	757	720	512	636
749	474	631	294	433	476	347	661	197	718	393	709
577	174	712	665	592	328	599	435	311	375	719	385
280	480	299	261	292	423	242	159	283	696	542	761
641	349	582	458	565	171	432	431	731	520	732	597
541	605	666	209	368	202	612	337	680	156	272	652
619	633	657	135	125	150	148	506	255	495	104	313
535	540	401	521	412	320	554	634	572	684	654	332
523	627	449	690	183	460	251	682	602	664	742	685
470	309	161	350	515	266	452	640	501	444	567	606
502	429	322	483	711	454	428	726	357	544	378	177
184	291	407	763	624	591	201	218	510	391	556	124
478	293	566	557	623	629	189	721	321	219	270	232
208	302	708	608	586	362	213	388	672	707	248	568
307	271	405	130	115	663	185	253	143	681	698	545
176	422	754	695	450	396	668	109	439	210	607	105
692	673	759	637	112	516	699	121	584	286	736	358
107	147	326	353	562	228	127	513	215	687	555	491
700	649	529	190	738	191	507	138	615	704	383	524
496	193	359	133	384	564	306	550	438	173	236	386
549	167	346	120	448	457	126	300	558	192	254	451
559	482	706	303	284	160	247	688	166	351	158	360

Clé de chiffrement de la deuxième version chiffrée de Logo ($Taille_{Clé}$ = 0,9375 K octets) :

498	116	41	239	486	387	497	89	154	262	303	25
129	485	272	506	489	496	446	502	195	127	234	350
240	282	491	338	501	304	256	484	29	209	382	212
238	512	511	494	335	166	495	79	437	381	299	499
480	508	505	481	358	73	482	483	492	392	427	359
507	269	510	236	487	493	243	372	167	231	182	37

94	106	490	87	333	500	71	64	16	504	488	160
503	438	97	227	1	317	336	331	324	76	150	478
509	90	203	716	401	386	194	442	477	328	529	4
762	474	132	3	709	407	735	376	86	596	2	48
651	228	704	200	345	380	201	626	18	5	23	24
385	660	531	732	145	248	562	677	373	663	725	648
736	712	453	84	365	52	569	128	170	183	526	754
43	294	746	146	224	83	112	745	434	448	764	582
400	541	436	635	339	597	390	632	717	26	738	551
383	721	237	93	156	144	463	623	375	627	727	130
152	173	728	731	281	722	220	68	707	656	153	204
193	680	252	54	95	550	429	465	759	70	542	34
55	528	149	414	763	590	232	729	218	646	639	600
458	49	472	57	678	630	664	374	593	533	750	585
190	344	384	117	631	676	539	205	208	300	577	657
697	196	696	394	141	689	131	559	445	513	99	69
669	706	693	578	32	142	756	247	169	768	100	742
217	290	59	264	393	755	330	701	703	332	743	726
266	444	221	549	199	163	659	370	270	250	621	702
268	614	740	552	21	96	435	766	245	622	326	28
278	417	316	752	15	343	733	634	430	618	311	181
340	189	325	532	557	624	765	636	747	9	470	571
140	595	341	538	178	710	308	402	296	573	63	271
125	610	553	449	215	699	20	411	519	263	692	357
690	162	464	439	471	126	666	105	242	720	147	681
604	420	12	523	714	615	739	11	35	155	184	136
192	426	118	318	546	705	368	580	65	253	283	186
682	460	53	670	719	399	724	440	749	364	450	222
45	202	561	121	389	19	109	164	47	261	223	246
77	558	408	82	476	179	423	535	56	616	8	139
38	760	80	406	521	405	527	298	441	249	454	396
723	447	708	748	91	348	684	120	180	715	314	210
617	655	751	658	683	312	391	168	607	61	598	33
404	424	673	570	601	418	295	642	346	613	694	643
159	661	111	257	50	327	287	674	589	289	39	62
219	285	443	143	30	640	274	462	757	362	124	310
397	753	60	259	214	254	216	81	92	412	548	574
555	293	276	534	695	667	671	605	473	302	665	176
433	594	255	265	591	587	516	456	267	206	518	455
369	319	306	98	431	603	292	395	291	378	602	103
191	638	107	688	40	174	581	356	85	102	469	679
711	347	46	213	409	119	138	566	547	371	342	377
211	628	641	165	554	609	425	78	421	413	108	575
525	612	568	10	280	114	349	520	536	468	517	652
451	649	88	197	198	416	226	110	134	157	185	172
592	556	337	415	66	524	734	75	241	279	113	698
744	654	315	422	351	27	515	320	687	611	686	700
297	572	653	650	361	543	629	761	14	475	619	175
586	51	579	288	355	540	466	637	244	730	606	564
353	36	691	235	230	563	379	398	44	522	567	576
7	403	718	388	363	685	565	758	599	329	133	273
137	459	229	122	625	584	17	275	322	251	187	367
410	334	645	560	158	123	633	583	305	101	151	741
428	537	457	452	313	354	207	737	277	366	675	233
258	309	148	620	479	647	58	161	352	767	323	22
42	432	467	713	588	225	668	177	321	419	104	6
360	188	644	301	135	74	72	461	672	31	307	286
171	67	260	284	530	545	608	544	115	13	662	514

Le tableau suivant (tableau V.2) représente une récapitulation des résultats de chiffrement des données tests présentées ci-dessus :

			Taille donnée (éléments)	Taille clé (bits)	Efficacité	Temps chiffrement (s)	Temps déchiffrement (s)
Données texte	Texte1	Version-Chiff1	1084	15323	15784.0	28.35	2,06
		Version-Chiff2			15030.0	28.02	2,13
	Texte2	Version-Chiff1	1151	15323	19498.0	29.16	2,67
		Version-Chiff2			18712.0	28.97.	2,65
Données images	Lena	Version-Chiff1	131 X 131	7680	139362.0	14.62	3.7
		Version-Chiff2			134074.0	14.21	4.01
	Logo	Version-Chiff1	420 X 395	7680	246426.0	15.06	3.11
		Version-Chiff2			225688.0	15.28	3.81

Tableau V.2. *Résultats obtenus par TabuCrypt.*

V.4. Discussion et évaluation des résultats

La première chose à remarquer d'après les résultats présentés dans la section précédente, est que le temps de chiffrement est proportionnellement dépendant de la taille de l'image à chiffrer. En effet, le temps de chiffrement de l'image Lena (34.826 secondes en moyenne) est plus petit par rapport au temps de chiffrement de l'image Logo (35.59 secondes en moyenne), du fait que la première est plus petite que la deuxième. La même remarque est valable pour le cas des deux données texte Texte1 et Texte2. Cette différence en temps revient au fait que le temps de chiffrement englobe le temps de lecture et de codage de la donnée à chiffrer en plus du temps de calcul, proprement dit, de la solution (temps de la recherche tabou). Le premier (temps de lecture et de codage) est strictement dépendant de la taille de la donnée à chiffrer, tandis que ce n'est pas le cas pour le deuxième (temps du calcul tabou). Donc, c'est surtout le temps de lecture et codage qui fait la différence en temps de chiffrement global d'une donnée par rapport à une autre.

De même, du coté du temps de déchiffrement c'est la phase de décodage de la donnée chiffrée qui fait la différence en temps de calcul.

Maintenant et en comparaison avec nos algorithmes développés et présentés précédemment à travers les deux précédents chapitres, le temps de calcul de ce dernier algorithme se trouve meilleur que le temps de calcul de certains de ces algorithmes et plus mauvais que le temps de calcul de certains autres algorithmes. La figure V.7 positionne le temps de calcul de l'algorithme *TabuCrypt* parmi nos quatre algorithmes PosESecL1, PosESecL2, OEEA et *AntCrypt*.

D'après le résumé de temps de calcul de nos cinq algorithmes développés par utilisation des trois métaheuristiques d'algorithmes évolutionnaires (PosESecL1, PosESecL2 et OEEA), de colonies de fourmis (*AntCrypt*) et de recherche taboue (*TabuCrypt*), nous constatons que *TabuCrypt* possède un temps de calcul meilleur que celui de PosESecL1 et de PosESecL2 mais plus grand que celui de OEEA et celui de *AntCrypt*.

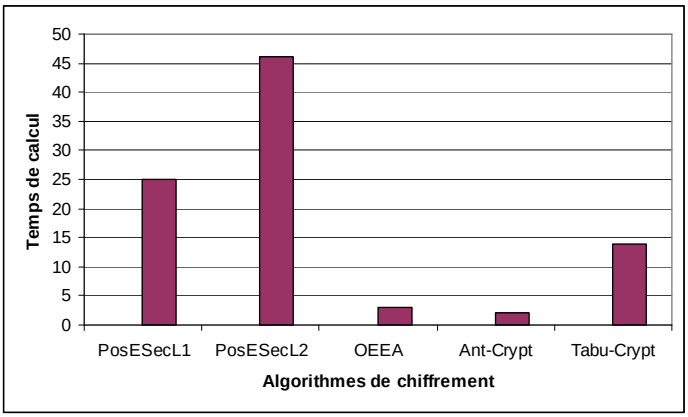


Figure V.7. *Comparaison des temps de calcul.*

Et comme *TabuCrypt* est un algorithme bâti par exploitation d'une métaheuristique, et nous l'avons expliqué précédemment, l'adaptabilité et l'efficacité d'application de telles méthodes dans un tel domaine à cause de leur large utilisation de l'aléatoire a été la source de notre motivation d'application de métaheuristique. L'algorithme développé sera, ainsi, doté d'un grand pouvoir confusionnel pour compliquer le plus possible la tâche de cryptanalyse.

En effet, l'aspect non déterministe innové à travers nos approches proposées, et par conséquent celle, ici, présentée, représente l'un des points forts de l'algorithme *TabuCrypt* tout comme nos autres algorithmes leur assurant, ainsi, une résistibilité contre les attaques différentielles. Ceci est assuré par une élection aléatoire, sur l'ensemble des générations, des voisins des éléments codant une solution intermédiaire pour calculer une autre solution intermédiaire ou une solution finale qui soit satisfaisante.
De même, l'attaque statistique sera presque impossible à appliquer grâce à ce dernier point en plus du fait que les données originales et leurs version chiffrées sur une instance, seront presque toutes différentes comme le montre le tableau suivant (tableau V.3) résumant les valeurs d'application des mesures de similarité NPCR, MAE et MSE.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.8927	0.97	0.68
Lena-version chiffrée 2	0.9104	1.05	0.71

Tableau V.3. *Niveaux de confusion de Tabu-Crypt.*

De son tour, l'attaque exhaustive sera mise à l'écart grâce à la taille de clé qui est largement sécurisée et qui est de taille égale à 15323 bits dans le cas de manipulation de données texte et qui égale à 7680 bits dans le cas de manipulation de données images.

Les attaques fréquentielle sont aussi pénalisées du fait, d'un coté, que ces dernières étudient les fréquences d'apparition des caractères dans un message (précisément un texte) écrit suivant une langue naturelle, alors que notre algorithme *TabuCrypt* peut traiter toutes sortes de message constitué de caractères du code Unicode. D'un autre coté, le message chiffré peut être constitué de caractères complètement ou partiellement différents de ceux constituant le message en clair.

V.5. Conclusion

La recherche Tabou est une méthode d'optimisation qui utilise la notion de voisinage d'une solution. Cette méthode permet, à partir d'une solution initiale, de visiter un ensemble de voisins. Ces voisins seront évalués et permettent l'évolution, à travers des règles spécifiques, vers une solution finale.

Ainsi et dans ce chapitre, nous avons présenté notre nouvel algorithme de chiffrement tabou, *TabuCrypt*, qui opère suivant le mode de chiffrement à base d'occurrence expliqué et validé dans le troisième chapitre.

Nous avons présenté en détails les différentes étapes du processus développé et nous l'avons testé sur différentes données test. Ainsi, notre objectif qui se résume en l'exploitation d'un voisinage en recherche locale de type taboue pour la manipulation de données texte et images, est atteint dans certains cotés vu les qualités de la méthode tabou (simplicité du principe, performance…) en donnant lieu à des résultats de bon niveau de confusion. Toutefois, le temps de convergence du processus de résolution est relativement lent par rapport à certains de nos précédents algorithmes proposés : OEEA et *AntCrypt*. Cela revient à l'aspect parallèle encapsulé à travers le principe soit des AEs ou des algorithmes de colonies de fourmis.

Conclusion et perspectives

Au cours de cette thèse nous avons proposé une nouvelle approche de cryptage de données texte et images par exploitation de métaheuristiques parmi lesquelles, nous nous sommes intéressés à celle inspirée du monde biologique et désignée par « approches biomimétiques ». Il s'agit des algorithmes évolutionnaires inspirés des principes de l'évolution naturelle des espèces. Par la suite, nous avons abordé une approche inspirée de modèles d'organisation naturels observés dans les sociétés animales, en particulier, dans une colonie de fourmis. Ces deux méthodes exploitées appartiennent à la catégorie des métaheuristiques à population. De même et pour l'autre catégorie qui est celle des métaheuristiques à trajectoire, nous sommes aussi arrivés à proposer un autre algorithme de cryptage utilisant le principe d'une recherche tabou.

Ainsi et après avoir étudié un panel assez diversifié des techniques de cryptage et avoir analysé les problèmes de sécurisation du transfert, nous avons élaboré nos propositions de stratégies de résolution d'un tel problème. Autrement dit et afin de détailler nos apports, notre thèse a été organisée autour de deux parties dont l'objectif de la première est de situer le lecteur dans le contexte de notre travail pour lui permettre de suivre les démarches réalisées dans cette étude. Dans la deuxième, nous avons détaillé la conception, la réalisation et l'évaluation de l'approche de résolution proposée.

Notre première contribution a porté sur l'élaboration et la conception de nouveaux algorithmes de chiffrement efficaces et robustes qui traitent le problème de taille de clés leur permettant, ainsi, d'être loin des attaques exhaustives.

Pour ce faire, nous avons commencé par une étude approfondie du problème tout en essayant de le ramener en un problème d'optimisation par une formulation adéquate aux métaheuristiques à appliquer : métaheuristique des algorithmes évolutionnaires, métaheuristique des algorithmes de colonies de fourmis et celle de recherche tabou. Pour la première, les obstacles à franchir étaient surtout la définition d'un codage adéquat de solutions. Dans ce sens, deux modes de codage différents ont été proposés : codage à base de positions et un deuxième à base d'occurrences donnant naissance, ainsi, à trois algorithmes de chiffrement, PosESecL1, PosESecL2 et OEEA de qualités différentes dont celui opérant suivant le mode de chiffrement par occurrences (OEEA) était le plus efficace et le plus robuste.

L'application de la deuxième métaheuristique a démontré l'efficacité du codage à base d'occurrences. À ce stade aussi, il a fallu définir avec précaution certains paramètres tels que la fonction d'évaluation, le mécanisme de manipulation de phéromone (incrémentation et évaporation) ainsi que celui de génération de la clé de chiffrement qui doit doter l'algorithme développé, *AntCrypt*, d'une résistance contre les attaques exhaustives vu que cet algorithme est de type symétrique.

De même, l'exploitation de la troisième métaheuristique a, aussi, abouti au développement d'un cinquième algorithme de chiffrement, *TabuCrypt*. Cela a nécessité la projection des composants d'une telle métaheuristique sur le problème de cryptage, à savoir la

liste tabou, stratégie de choix de voisins, table de hachage et les mécanismes avancés en recherche tabou (Intensification, diversification et aspiration). Les résultats étaient satisfaisants que ce soit sur le plan cryptographique que cryptanalytique, toutefois, il nécessite plus de temps de calcul que certains de nos autres algorithmes (*OEEA* et *AntCrypt*).

Notre deuxième contribution était l'innovation d'algorithmes de chiffrement non déterministes en se bénéficiant de l'aspect aléatoire de la sélection naturelle ou du déplacement des fourmis. En effet, l'aléatoire représente l'ennemi des cryptanalystes. En appliquant toutes les attaques redoutables, il est donc quasiment impossible de parvenir à l'information initiale ou au moins d'établir une relation entre les données chiffrées puisqu'on obtient toujours des versions chiffrées différentes à chaque application de l'algorithme même lors d'un chiffrement sur plusieurs instances d'une même donnée originale.

En fait, à travers notre étude nous sommes arrivés à présenter les métaheuristiques comme des méthodes de résolution approchées se voulant simples et adaptables à tout type de problèmes. Leur capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la solution trouvée. Du point de vue de la recherche opérationnelle, cet inconvénient n'est pas toujours un problème, tout spécialement quand une seule approximation de la solution optimale est recherchée. En nous situant dans ce cas, nous aurions pu mettre en évidence l'avantage de ces nouvelles approches pour résoudre le problème de cryptage de données texte ou images.

Toutefois et malgré que nos algorithmes développés soient de bonne qualité cryptographique, plusieurs autres idées et applications en relation avec ce présent travail pour le compléter, restent à concrétiser. En premier lieu, il est urgent d'aborder le problème de métaréglage [Dréo, 2004], non seulement pour régler automatiquement les paramètres des métaheuristiques, mais aussi pour maîtriser leurs fonctionnements.

De même et malgré que les algorithmes évolutionnaires ou même les algorithmes de colonies de fourmis permettent d'obtenir des solutions pas toujours optimales, mais possiblement satisfaisantes à de nombreux problèmes complexes, leur puissance d'optimisation est liée à la puissance sous-jacente des machines les exécutant.

Comme ce sont des algorithmes inspirés de la nature, ils sont fortement parallélisables et il se trouve que le récent développement des cartes graphiques, dédiées au calcul de rendu 3D dans un premier temps, puis généralisées au milieu des années 2000 avec les *General Purpose Graphic Processing Units (GPGPU)*, permet d'utiliser le grand nombre de cœurs de calcul qu'elles comportent pour autre chose que du rendu graphique.

Dans [Mait, 2011], l'auteur détaille et étudie comment les algorithmes évolutionnaires peuvent bénéficier de ces nouvelles architectures, avec leurs particularités. Plusieurs algorithmes ont été présentés, qui permettent l'utilisation efficace de ces processeurs pour l'évolution artificielle, que ce soit pour les algorithmes génétiques et stratégies d'évolution, mais aussi pour la programmation génétique. Différentes stratégies de parallélisation ont été utilisées, dont l'une nécessitant le développement d'un opérateur parallèle de réduction de la population. L'adaptation des paramètres de ces algorithmes pour permettre le portage de ces derniers sur GPGPU est détaillée, ainsi que certains points impactant directement les performances de ces portages.

Ainsi, de tels schémas peuvent être envisagés pour une parallélisation de nos algorithmes développés surtout PosESecL2 possédant un bon niveau confusionnel mais souffrant d'une lenteur de calcul vu la grande taille de chromosomes manipulés codant des données à chiffrer de grandes tailles.

D’un autre coté et vu que nos algorithmes de cryptage développés sont de type symétrique, une méthode de communication sécurisée de clé de chiffrement secrète peut être envisagée lors de la manipulation de données images. Il s’agit de marquer l'image cryptée dans des régions spécifiques, ou des régions déterminées pour l'utilisateur. Nous pouvons également utiliser des méthodes de marquage (watermarking) sans perte pour insérer la clef cryptée dans l'image cryptée. Il serait souhaitable de développer des évaluations sur la robustesse concernant les attaques. Ainsi, nous souhaitons faire des études concernant la cryptanalyse de nos méthodes pour classifier son niveau de sécurité face aux attaquants.

Par ailleurs, nos perspectives de recherche à long terme s’orientent vers l’étude et le développement d’autres méthodologies de cryptage de données. Nous nous intéresserons, en particulier à :

- La métaheuristique des automates cellulaires inventée par Ulam [Ulam, 1950] et Von Neumann [VonN, 1966] et basée sur le concept de la vie artificielle. Ce concept s’appuie sur des règles extrêmement simples permettant de construire des structures très complexes et esthétiques, d’où sa ressemblance avec le principe d’auto-organisation en biologie animale, notamment avec les mécanismes de la colonie de fourmis. Ces règles peuvent stipuler, par exemple, qu'une naissance nécessite un certain rassemblement de population, que les cellules ne peuvent survivre à un trop grand isolement et qu'une trop forte concentration les étouffe. Donc, nous comptons adapter ces règles au problème de cryptage où de génération en génération, l’application de ces règles permettra l’émergence d’une solution au problème.

- L’approche des « hyper-heuristiques » introduite dans [Cowl, 2000], [Cowl, 2002], [Burk, 2003], [Burk, 2005] et qui consiste à travailler dans un espace de recherche composé de métaheuristiques afin d’optimiser le choix de la métaheuristique à utiliser pour un problème donné. Il s’agit, donc, de mettre en confrontation un ensemble de métaheuristiques différentes pour en sélectionner la plus optimale.

REFERENCES BIBLIOGRAPHIQUES

[Aart, 1997] **Aarts, E.H.L. & Lenstra, J.K. (1997).** Local search in combinatorial optimization. John Wiley & Sons.

[Ayas, 2004] **Ayas, J. & André Viau, M. (2004).** La recherche Tabou. Notes de cours.

[Barn, 2004] **Barni, M. & Bartolini, F. (2004).** Watermarking Systems Engineering : Enabling Digital Assets Security and Other Applications. ISBN: 0-8247-4806-9, printed on acid-free paper.

[Barg, 2005] **Bargeton, A. & Devèze, B. (2005).** Les colonies de fourmis : apprentissage coopératif pour le problème du voyageur de commerce. Projet ANIMAT, Université Pierre et Marie Curie.

[Baud, 2005a] **Baudry, B., Fleurey, F., Jézéquel, J-M, & Le Traon, Y. (2005).** Automatic test, case optimization : A bacteriologic algorithm. IEEE software, 22(2):76-82.

[Baud, 2005b] **Baudry, B. (2005).** Assemblage testable et validation de composant. Thèse PhD, Université de Rennes 1.

[Baud, 2005c] **Baudry, B., Fleurey, F., Jézéquel, J-M, & Le Traon, Y. (2005).** from genetic to bacteriological algorithms for mutation-based testing : Research articles, Softw. Test. Verif Relia. 15 (2):73-96.

[Beck, 1992] **Beckers, R., Deneubourg, J. L., & Goss, S. (1992).** Trails and U-Turns in the Selection of a Path by the Ant Lasius Niger. J. Theor. Biol., 159 :397-415.

[Biha, 1991] **Biham, E. et Shamir, A. (1991).** Differential Cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1):3_72.

[Bona, 1994] **Bonabeau, E., & Theraulaz, G. (1994).** Intelligence collective. Edition Hermes Sciences, Paris, France.

[Burk, 2003] **Burke, E.K., Silva, J.D. Landa & Soubeiga, E. (2003).** Hyperheuristic Approaches for Multiobjective Optimisation. In Proceedings of the Fifth Metaheuristics International Conference, MIC 2003, August, Kyoto, Japan.

[Burk, 2005] **Burke, E. K., Silva, D. Landa & Soubeiga, E. (2005).** Multi-objective Hyperheuristic Approaches for Space Allocation and Timetabling. In Meta-heuristics: Progress as Real Problem Solvers (Eds. Baraki, T., Nonobe, K. & Yagiura, M.), pp.129-158, Springer.

[Camp, 2010] **Campesato, J-B. (2010).** Introduction à la physique quantique du xxie siècle : La cryptographie quantique. Cours UEL.

[Cler, 2004] **Clerc, M. (2004).** L'optimisation par essaims particulaires. Hermès Science.

[Coll, 2002] **Collette, Y. & Siarry, P. (2002).** Optimisation multiobjectif. Eyrolles.

[Cost, 2006] **Costanzo, A., Luong, T. & Guillaume, V.M. (2006).** Les métaheuristiques en optimisation combinatoire.

[Cowl, 2000] **Cowling, P., Kendall, G. & Soubeiga, E. (2000)**. A Hyperheuristic Approach to Scheduling a Sales Summit. Practice and Theory of Automated Timetabling III : Third International Conference (PATAT 2000) (Eds. Burke, E.K. & Erben, W.), August 2000, Konstanz, Germany, pp. 176-190.

[Cowl, 2002] **Cowling, P., Kendall, G. & Soubeiga, E. (2002).** Hyperheuristics: A Tool for Rapid Prototyping in Scheduling and Optimisation. sApplications of Evolutionary Computing: Proceeding of Evo Workshops 2002 (Eds. Cagoni,

S., Gottlieb, J., Hart, E., Middendorf, M. & Goenther, R.), April 3-4, Kinsale, Ireland, pp. 1-10.

[Davi, 1985] **Davis L, (1985).** Applying Adaptive Algorithms to Epistatic Domains. Proceedings of the International Joint Conference on Artificial Intelligence, pp162-164.

[DeJo, 2007] **De Jong, K. (2007**). Parameter Setting in EAs: a 30 Year Perspective, Studies in Computational Intelligence (SCI) 54, 1–18, Springer-Verlag Berlin Heidelberg.

[Dene, 1977] **Deneubourg, J. (1977).** Application de l'ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites. Insectes Sociaux, pp 117-130.

[Dene, 1983] **Deneubourg, J.L., Pasteels, J.M. & Verhaeghe, J.C. (1983).** Probabilistic behaviour in ants : a strategy of errors. Journal of Theoretical Biology, 105, pp.259-271.

[Dene, 1989] **Deneubourg, J.L. & Goss, S. (1989).** Collective patterns and decision-making. Ethology and Evolution, p 295-311.

[Dori, 1992] **Dorigo, M. (1992).** Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[Dori, 1999] **Dorigo, M., & Di Caro, G. (1999).** The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw Hill, London, UK, pages 11–32.

[Dori, 2003] **Dorigo, M. & Stutzle, T. (2003).** Handbook of Metaheuristics. Volume 57 of International series in operations research and management science, chapter The Ant Colony Optimization Metaheuristics : Algorithms, Applications and Advances. Kluwer Academic Publishers, Boston Hardbound.

[Dori, 2004] **Dorigo, M. & Stützle, T. (2004).** Ant colony optimization. MIT Press.

[Dréo, 2003] **Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2003).** Métaheuristiques pour l'optimisation difficile. Eyrolles.

[Dréo, 2004] **Dréo, J. (2004).** Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical. Thèse de Doctorat, Université de Paris 12, France, 166 p.

[Dupa, 2004] **Dupas, R. (2004).** Amélioration de performance des systèmes de production : apport des algorithmes évolutionnistes aux problèmes d'ordonnancement cycliques et flexibles. HDR, Université d'Artois.

[Eibe, 2007] **Eiben, A.E., Michalewicz, Z. Schoenauer, M. & Smith, J.E. (2007).** Parameter Control in Evolutionary Algorithms, Studies in Computational Intelligence (SCI) 54, 19–46, Springer-Verlag Berlin Heidelberg.

[Foge, 1966] **Fogel, L., Owens, A., & Walsh, M. (1966).** Artificial Intelligence Through Simulated Evolution. Wiley, J., Chichester, UK. In [Jour, 2003].

[Gant, 2001] **Ganteaut, A. et Lévy, F. (2001).** La cryptologie moderne. L'Armement, 73:76_83.

[Gher, 2004] **Ghernaouti-Hélie, S. (2004).** Sécurité informatique et réseaux, Editions Dunod.

[Glov, 1986a] **Glover, F. (1986).** Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 13 : 533-549.

[Glov, 1986b] **Glover, F. (1986).** Future paths for integer programming and links to artificial intelligence. Computing and operating research, tome 13.

[Glov, 1989] **Glover, F. (1989, 1990).** Tabu search, Part1 and Part2. ORSA journal on computing, tome 1 et tome 2.

[Glov, 1997] **Glover, F. & Laguna, M. (1997).** Tabu Search. Kluwer, Boston.

[Gold, 1989] **Goldberg, D.E. (1989).** Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, New York.

[Goss, 1989] **Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989).** Self-Organized Shortcuts in the Argentine Ant. Naturwissenchaften, 76 :579-581.

[Gren, 1986] **Grenfenslette J.J. (1986).** Optimization of control parameters for genetic algorithms. IEEE translation on system Man and cybernetics, Vol 16 №1, pp122-128.

[Gury, 2004] **Gury, S. & Rémond, N. (2004).** Cryptologie : ElGamal d'après Diffie-Hellman.

[Hert, 2003] **Hertz, A. & Widmer, M. (2003).** Guidelines for the use of meta-heuristics in combinatorial optimization. European Journal of Operational Research, tome 151.

[Holl, 1975] **Holland, J. (1975).** Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.

[Huff, 1952] **Huffman, D.A. (1952).** A method for the construction of minimum-redundancy codes. Proceedings of the I.R.E., pp 1098-1102. http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf

[JinK,1999] **Jin-Kao, H., Galinier, P. & Habib, M. (1999).** Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. Revue d'Intelligence Artificielle, Vol : No. 1999.

[Jour, 2003] **Jourdan, L. (2003).** Métaheuristiques pour l'extraction de connaissances : application à la génomique. Thèse de doctorat, U.S.T.L.

[Kamm, 2006] **Kammarti, R. (2006).** Approches évolutionnaires pour la résolution du 1-PDPTW statique et dynamique. Thèse de doctorat, délivrée conjointement par l'école centrale de lille et l'université des sciences et technologies de lille.

[Katz, 2000] **Katzenbeisser, S. & Petitcolas, A.P. (2000).** Information hiding techniques for steganography and digital watermarking, Artech House, Boston-London.

[Kerc, 1883] **Kerckhoffs, A. (1883).** La cryptographie militaire. Journal des sciences militaires, Janvier 1883.

[Koba, 1990] **Kobayashi, M. (1990).** DigitalWatermarking : Historical Roots. Technical report, IBM Research, Tokyo Research Laboratory. Japan.

[Koza, 1992] **Koza, J. (1992).** Genetic Programming. Cambridge, MA: MIT Press.

[Lafo, 2006] **Lafourcade, P. (2006).** Vérification de protocoles cryptographiques en présence de théories équationnelles. Mémoire d'HDR, École Normale Supérieur de Cachan, Cachan, France.

[Lapo, 1996] **Laporte, G. & Osman, I.H. (1996).** Metaheuristics in combinatorial optimization, Annals of Operations Research 63. J.C. Baltzer Science Publishers, Basel, Switzerland.

[Laur, 1978] **Lauriere, J.L. (1978).** A language and a program for stating and solving combinatorial problems, Artificial Intelligence 10 : 29-127, 1978.

[Lawr, 1987] **Lawrence, D. (1987).** Genetic Algorithm and Simulated Annealing. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

[Laye, 2010] **Layeb, A.S. (2010).** Utilisation des Approches d'Optimisation Combinatoire pour la Vérification des Applications Temps Réel. Thèse de Doctorat, Université de Constantine, Algérie.

[Lepr, 2000] **Leprévost, F. (2000).** Les standards cryptographiques du XXIe siècle : AES et IEEE-P1363. Gazette des Mathématiciens - n°85.

[LinS, 1973] **Lin, S. & Kernighan, B.W. (1973).** An efficient heuristic for the traveling-salesman problem. Operations Research 21 : 498-516.

[Lobo, 2004] **Lobo, F.G. & Goldberg, D.E. (2004**). The parameter-less genetic algorithm in practice. Information Sciences, 167:217–232.

[Lobo, 2007] **Lobo, F.G. & Lima, C.F. (2007**). Parameter Setting in Evolutionary Algorithms : Adaptive Population Sizing Schemes in Genetic Algorithms. Springer Berlin / Heidelberg.

[Loid, 2005] **Loidreau, P. (2005).** Introduction à la cryptographie. LinuxFocus, Article № 243.

[Luke, 2010] **Luke. S. (2010).** Essentials of Metaheuristics.

[Magn, 2001] **Magnin, V. (2001).** Optimisation et algorithmes génétiques. Cyber-cour gratuit de l'EUDIL.
https://iris.univ-lille1.fr/dspace/bitstream/1908/499/2/

[Mait, 2011] **Maitre, O. (2011).** Parallélisation d'algorithmes évolutionnaires sur carte GPGPU. Thèse de Doctorat, Université de Strasbourg, France.

[Mats, 1994] **Matsui, M. (1994).** Linear cryptanalysis method for DES cipher. Advances in Cryptology**,** EUROCRYPT'93, volume 765 de Lecture Notes in Computer Science, Springer-Verlag.

[Mene, 1996] **Menezes, A.J., Van Oorschot, P.C. & Vanstone, S.A. (1996).** Handbook of Applied Cryptography. CRC Press.

[Mesg, 1999] **Mesghouni, K. (1999).** Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de production. Thèse de doctorat, Ecole centrale de lille et l'université de lille1.

[Mich, 1992] **Michalewicz, Z. (1992).** Genetic Algorithms + data structures = evolution programs, Springer Verlag, Berlin.

[Mich, 1996] **Michalewicz, Z. (1996).** Genetic Algorithms + Data Structures = Evolution Programs. 3rd Edition, Springer Verlag, Berlin.

[Mich, 2007] **Michalewicz, Z. & Schmidt, M. (2007**). Parameter Control in Practice, Studies in Computational Intelligence (SCI) 54. 277–294, Springer-Verlag Berlin Heidelberg.

[Monm, 2000] **Monmarché, N. (2000).** Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation. Thèse de Doctorat, Université de Tours, France.

[Nave, 2002] **Navez, P. & Van Assche, G. (2002).** Une transmission sécurisée : la cryptographie quantique.

[Omar, 2006] **Omary, F. (2006).** Application des algorithmes évolutionnistes à la cryptographie. Mémoire d'HDR, Université Mohamed V- Agdal, Rabat, Marroc.

[Omar, 2007] **Omary, F., Tragha, A.R, Bellaachia, A.G & Mouloudi, A.A. (2007).** Design and Evaluation of Two Symmetrical Evolutionist-Based Ciphering Algorithms. IJCSNS, International Journal of Computer Science and Network Security, VOL.7 No.2.

[Osma, 1996] **Osman, I.H. & Kelly, J.P. (1996).** Meta-heuristics: theory and applications. Kluwers Academic Publishers, Boston.

[Pear, 1984] **Pearl, J. (1984).** Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading.

[Prab, 1998] **Prabhu, D., Buckles, B. P., & Petry, F. E. (1998).** Genetic algorithms for scene interpretation from prototypical semantic description. Submitted to IEEE Transactions on Evolutionary Computation. 22 p.

[Preu, 2007] **Preuss, M. & Bartz-Beielstein, T. (2007**). Sequential Parameter Optimization Applied to Self-Adaptation for Binary-Coded Evolutionary Algorithms.

Studies in Computational Intelligence (SCI) **54**, 91–119, Springer-Verlag Berlin Heidelberg.

[Radc, 1995] **Radcliff, N.J. & Surry, P.D. (1995).** Fundamental limitations on search algorithms: evolutionary computing in perspective. Lecture Notes in Computer Science 1000, Springer-Verlag.

[Reev, 1993] **Reeves, C.R. (1993).** Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publications, Oxford.

[Rech, 2003] **Rechenberg, I. (2003).** Evolutions Strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart. In [Jour, 2003].

[Renn, 2000] **Rennard, J-P. (2000).** Introduction aux algorithmes génétiques. http://www.rennard.org/alife/french/gavintr.html

[Sema, 2009a] **Semassel, M.L, Souici, I. & Seridi, H. (2009).** Extension et Parallèlisation d'un Algorithme de Chiffrement Évolutionnaire basé Occurrences. CDROM, Conférence international sur l'Informatique et ses Applications, CIIA'09, 03-04 mai, Saida, Algérie. http://www.informatik.uni-trier.de/~ley/db/conf/ciia/ciia2009.html/.

[Sema, 2009b] **Semassel, M.L, Souici, I. & Seridi, H. (2009).** Extension et parallélisation d'un algorithme de chiffrement évolutionnaire basé occurrences. Colloque International Nouvelles Techniques Immuno-Cognitives dans les Réseaux Informatique, NTICRI'09, 10-11 Mai, Oran, Algérie.

[Shan, 1949] **Shannon, C. (1949).** Communication Theory of Secrecy Systems. Bell Systems Technical Journal.

[Soui, 2008a] **Souici, I., Seridi, H. & Aissaoui, M. Z. (2008).** Nouvel Algorithme de Chiffrement Evolutionniste ACEO (Algorithme de chiffrement Evolutionniste basé Occurrences)", Revue des Sciences, Technologie & Développement, N° 4, pp. 44, 56. ISSN : 1112 7309, http://www.andru.gov.dz/

[Soui, 2008b] **Souici, I., Aissaoui, Z. & Seridi, H. (2008).** Conception et Evaluation d'un Nouvel Algorithme Crypto-évolutionnaire (SSA). COSI'08. Colloque sur l'Optimisation et les Systèmes d'Information. Tizi-Ouzou, Algérie.

[Soui, 2009] **Souici, I., Benhamza, K. & Seridi, H. (2009).** Nouvel Algorithme Crypto-évolutionnaire (ACEO) : Conception et Evaluation. CDROM, 6ème Conférence sur le Génie Electrique (CGE'06), 13-14 Avril, Ecole Militaire Polytechnique, Bordj El Bahri, Algérie. http://www.emp.edu.dz/Manif_Scientifique/CGE06/CGE06.htm.

[Soui, 2010a] **Souici, I., Seridi, H., & Akdag, H. (2010).** Images evolutionary encryption. Premier Congrès International sur les modèles, Optimisation et Sécurité des Systèmes, ICMOSS'2010, 29-31 Mai, Tiaret, Algérie.

[Soui, 2010b] **Souici, I., Seridi, H., & Akdag, H. (2010).** Images encryption by the use of evolutionary algorithms. International Conference on Electrical Engineering, Electronics and Automatic'10, ICEEA'2010, 2-3 Novembre, Bejaia, Algérie.

[Soui, 2011a] **Souici, I & Seridi, H. (2011).** Une méthode rapide et efficace pour le cryptage évolutionnaire d'images. 8ème Colloque International sur l'Optimisation et les Systèmes d'Information, COSI'11, 24-27 avril, Guelma, Algérie. http://www.isima.fr/cosi/cosi2011/.

[Soui, 2011b] **Souici, I., Seridi, H. & Akdag, H. (2011).** Images Encrypton by the Use of Evolutionary algorithms. Analog Integrated Circuits and Signal Processing, Springer, Volume 69, Issue 1, pp 49-58, ISSN 1573-1979, Springer. http://www.springerlink.com/content/y17861290161146m/.

[Ster, 2004] **Stern, J., Granboulan, L., Nguyen, P. & Pointcheval, D. (2004).** Conception et preuves d'algorithmes Cryptographiques. Cours de magistère M.M.F.A.I, École normale supérieure.

[Stin, 1996] **Stinson, D. (1996).** Cryptographie, théorie et pratique. International Thomson Publishing, France.

[Tail, 1991] **Taillard, E.D. (1991).** Robust taboo search for the quadratic assignment problem. Parallel computing. Tome 17, p443-455.

[Tail, 1995] **Taillard, E.D. (1995).** Comparison of iterative searches techniques for the quadratic assignment problem. Tome 3, numéro 2, p87-105.

[Tail, 1999] **Taillard, E.D. (1999).** Some efficient heuristic methods for the flow shop sequencing problem. European journal of operational research, tome 47, numéro 1, p65-74.

[Talb, 1999] **Talbi, E.G. (1999).** Métaheuristiques pour l'optimisation combinatoire multi□objectif: Etat de l'art. CNET, PE: 98□757.33.

[Ulam, 1950] **Ulam, S. (1950).** Random Processes and Transformations. In: S. Ulam, Sets, Numbers and Universes, Cambridge: MIT Press, pp. 326-337, 1974.

[VobS, 1993] **Voß, S. (1993).** Intelligent Search. Manuscript, TU Darmstadt.

[VobS, 1999] **Voß, S., Martello, S., Osman, I.H & Roucairol, C. (1999).** Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston.

[VonN, 1966] **Von Neumann, J. (1966).** Theory of Self-Reproducing Automata. University of Illinois Press.

[Wolp, 1997] **Wolpert, D.H. & Macready, W. G. (1997).** No free lunch theorems for optimization. IEEE Transactions On Evolutionary Computation 1(1) : 67-82.

[Zimm, 2005] **Zimmermann, P. (2005).** Cryptographie asymétrique : signature et RSA en détail. 6ème cours du module : Introduction à la cryptologie, Master M1 Informatique, 2005.
http://www.loria.fr/~zimmerma/cours/id12-2005-6.pdf

[www1] http://wapedia.mobi/fr/Cryptanalyse

[www2] http://www.arrobe.fr/docs/cryptologie.pdf

[www3] http://www.bibmath.net/crypto/moderne/des.php3

[www4] http://www.scryptosystems.com/whitepapers_securite.htm

[www5] http://www.uqtr.ca/~delisle/Crypto/prives/blocs_idea.php

[www6] http://www.bibmath.net/crypto/moderne/rsa.php3

[www7] http://www.apprendre-en-ligne.net/crypto/moderne/pgp.html

[www8] news:fr.misc.cryptologie

[www9] http://fr.wikipedia.org/wiki/Cryptographie_quantique

[www10] www.google.com_url_q=http//ufrsciencestech.u-bourgogne.fr_master2009Tabou.

[www11] http: //ufrsciencestech.u-bourgogne.fr /master1/ mi1tc5/ CM2009/ Genetiques/ Genetique_1.pdf

[www12] http://www.irisa.fr/triskell/home_html.

REFERENCES BIBLIOGRAPHIQUES

[Aart, 1997] **Aarts, E.H.L. & Lenstra, J.K. (1997).** Local search in combinatorial optimization. John Wiley & Sons.

[Ayas, 2004] **Ayas, J. & André Viau, M. (2004).** La recherche Tabou. Notes de cours.

[Barn, 2004] **Barni, M. & Bartolini, F. (2004).** Watermarking Systems Engineering : Enabling Digital Assets Security and Other Applications. ISBN: 0-8247-4806-9, printed on acid-free paper.

[Barg, 2005] **Bargeton, A. & Devèze, B. (2005).** Les colonies de fourmis : apprentissage coopératif pour le problème du voyageur de commerce. Projet ANIMAT, Université Pierre et Marie Curie.

[Baud, 2005a] **Baudry, B., Fleurey, F., Jézéquel, J-M, & Le Traon, Y. (2005).** Automatic test, case optimization : A bacteriologic algorithm. IEEE software, 22(2):76-82.

[Baud, 2005b] **Baudry, B. (2005).** Assemblage testable et validation de composant. Thèse PhD, Université de Rennes 1.

[Baud, 2005c] **Baudry, B., Fleurey, F., Jézéquel, J-M, & Le Traon, Y. (2005).** from genetic to bacteriological algorithms for mutation-based testing : Research articles, Softw. Test. Verif Relia. 15 (2):73-96.

[Beck, 1992] **Beckers, R., Deneubourg, J. L., & Goss, S. (1992).** Trails and U-Turns in the Selection of a Path by the Ant Lasius Niger. J. Theor. Biol., 159 :397-415.

[Biha, 1991] **Biham, E. et Shamir, A. (1991).** Differential Cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1):3_72.

[Bona, 1994] **Bonabeau, E., & Theraulaz, G. (1994).** Intelligence collective. Edition Hermes Sciences, Paris, France.

[Burk, 2003] **Burke, E.K., Silva, J.D. Landa & Soubeiga, E. (2003).** Hyperheuristic Approaches for Multiobjective Optimisation. In Proceedings of the Fifth Metaheuristics International Conference, MIC 2003, August, Kyoto, Japan.

[Burk, 2005] **Burke, E. K., Silva, D. Landa & Soubeiga, E. (2005).** Multi-objective Hyperheuristic Approaches for Space Allocation and Timetabling. In Meta-heuristics: Progress as Real Problem Solvers (Eds. Baraki, T., Nonobe, K. & Yagiura, M.), pp.129-158, Springer.

[Camp, 2010] **Campesato, J-B. (2010).** Introduction à la physique quantique du xxie siècle : La cryptographie quantique. Cours UEL.

[Cler, 2004] **Clerc, M. (2004).** L'optimisation par essaims particulaires. Hermès Science.

[Coll, 2002] **Collette, Y. & Siarry, P. (2002).** Optimisation multiobjectif. Eyrolles.

[Cost, 2006] **Costanzo, A., Luong, T. & Guillaume, V.M. (2006).** Les métaheuristiques en optimisation combinatoire.

[Cowl, 2000] **Cowling, P., Kendall, G. & Soubeiga, E. (2000)**. A Hyperheuristic Approach to Scheduling a Sales Summit. Practice and Theory of Automated Timetabling III : Third International Conference (PATAT 2000) (Eds. Burke, E.K. & Erben, W.), August 2000, Konstanz, Germany, pp. 176-190.

[Cowl, 2002] **Cowling, P., Kendall, G. & Soubeiga, E. (2002).** Hyperheuristics: A Tool for Rapid Prototyping in Scheduling and Optimisation. sApplications of Evolutionary Computing: Proceeding of Evo Workshops 2002 (Eds. Cagoni,

S., Gottlieb, J., Hart, E., Middendorf, M. & Goenther, R.), April 3-4, Kinsale, Ireland, pp. 1-10.

[Davi, 1985] **Davis L, (1985).** Applying Adaptive Algorithms to Epistatic Domains. Proceedings of the International Joint Conference on Artificial Intelligence, pp162-164.

[DeJo, 2007] **De Jong, K. (2007**). Parameter Setting in EAs: a 30 Year Perspective, Studies in Computational Intelligence (SCI) 54, 1–18, Springer-Verlag Berlin Heidelberg.

[Dene, 1977] **Deneubourg, J. (1977).** Application de l'ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites. Insectes Sociaux, pp 117-130.

[Dene, 1983] **Deneubourg, J.L., Pasteels, J.M. & Verhaeghe, J.C. (1983).** Probabilistic behaviour in ants : a strategy of errors. Journal of Theoretical Biology, 105, pp.259-271.

[Dene, 1989] **Deneubourg, J.L. & Goss, S. (1989).** Collective patterns and decision-making. Ethology and Evolution, p 295-311.

[Dori, 1992] **Dorigo, M. (1992).** Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[Dori, 1999] **Dorigo, M., & Di Caro, G. (1999).** The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw Hill, London, UK, pages 11–32.

[Dori, 2003] **Dorigo, M. & Stutzle, T. (2003).** Handbook of Metaheuristics. Volume 57 of International series in operations research and management science, chapter The Ant Colony Optimization Metaheuristics : Algorithms, Applications and Advances. Kluwer Academic Publishers, Boston Hardbound.

[Dori, 2004] **Dorigo, M. & Stützle, T. (2004).** Ant colony optimization. MIT Press.

[Dréo, 2003] **Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2003).** Métaheuristiques pour l'optimisation difficile. Eyrolles.

[Dréo, 2004] **Dréo, J. (2004).** Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical. Thèse de Doctorat, Université de Paris 12, France, 166 p.

[Dupa, 2004] **Dupas, R. (2004).** Amélioration de performance des systèmes de production : apport des algorithmes évolutionnistes aux problèmes d'ordonnancement cycliques et flexibles. HDR, Université d'Artois.

[Eibe, 2007] **Eiben, A.E., Michalewicz, Z. Schoenauer, M. & Smith, J.E. (2007).** Parameter Control in Evolutionary Algorithms, Studies in Computational Intelligence (SCI) 54, 19–46, Springer-Verlag Berlin Heidelberg.

[Foge, 1966] **Fogel, L., Owens, A., & Walsh, M. (1966).** Artificial Intelligence Through Simulated Evolution. Wiley, J., Chichester, UK. In [Jour, 2003].

[Gant, 2001] **Ganteaut, A. et Lévy, F. (2001).** La cryptologie moderne. L'Armement, 73:76_83.

[Gher, 2004] **Ghernaouti-Hélie, S. (2004).** Sécurité informatique et réseaux, Editions Dunod.

[Glov, 1986a] **Glover, F. (1986).** Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 13 : 533-549.

[Glov, 1986b] **Glover, F. (1986).** Future paths for integer programming and links to artificial intelligence. Computing and operating research, tome 13.

[Glov, 1989] **Glover, F. (1989, 1990).** Tabu search, Part1 and Part2. ORSA journal on computing, tome 1 et tome 2.

[Glov, 1997] **Glover, F. & Laguna, M. (1997).** Tabu Search. Kluwer, Boston.

[Gold, 1989] **Goldberg, D.E. (1989).** Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, New York.

[Goss, 1989] **Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989).** Self-Organized Shortcuts in the Argentine Ant. Naturwissenchaften, 76 :579-581.

[Gren, 1986] **Grenfenslette J.J. (1986).** Optimization of control parameters for genetic algorithms. IEEE translation on system Man and cybernetics, Vol 16 №1, pp122-128.

[Gury, 2004] **Gury, S. & Rémond, N. (2004).** Cryptologie : ElGamal d'après Diffie-Hellman.

[Hert, 2003] **Hertz, A. & Widmer, M. (2003).** Guidelines for the use of meta-heuristics in combinatorial optimization. European Journal of Operational Research, tome 151.

[Holl, 1975] **Holland, J. (1975).** Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.

[Huff, 1952] **Huffman, D.A. (1952).** A method for the construction of minimum-redundancy codes. Proceedings of the I.R.E., pp 1098-1102. http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf

[JinK,1999] **Jin-Kao, H., Galinier, P. & Habib, M. (1999).** Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. Revue d'Intelligence Artificielle, Vol : No. 1999.

[Jour, 2003] **Jourdan, L. (2003).** Métaheuristiques pour l'extraction de connaissances : application à la génomique. Thèse de doctorat, U.S.T.L.

[Kamm, 2006] **Kammarti, R. (2006).** Approches évolutionnaires pour la résolution du 1-PDPTW statique et dynamique. Thèse de doctorat, délivrée conjointement par l'école centrale de lille et l'université des sciences et technologies de lille.

[Katz, 2000] **Katzenbeisser, S. & Petitcolas, A.P. (2000).** Information hiding techniques for steganography and digital watermarking, Artech House, Boston-London.

[Kerc, 1883] **Kerckhoffs, A. (1883).** La cryptographie militaire. Journal des sciences militaires, Janvier 1883.

[Koba, 1990] **Kobayashi, M. (1990).** DigitalWatermarking : Historical Roots. Technical report, IBM Research, Tokyo Research Laboratory. Japan.

[Koza, 1992] **Koza, J. (1992).** Genetic Programming. Cambridge, MA: MIT Press.

[Lafo, 2006] **Lafourcade, P. (2006).** Vérification de protocoles cryptographiques en présence de théories équationnelles. Mémoire d'HDR, École Normale Supérieur de Cachan, Cachan, France.

[Lapo, 1996] **Laporte, G. & Osman, I.H. (1996).** Metaheuristics in combinatorial optimization, Annals of Operations Research 63. J.C. Baltzer Science Publishers, Basel, Switzerland.

[Laur, 1978] **Lauriere, J.L. (1978).** A language and a program for stating and solving combinatorial problems, Artificial Intelligence 10 : 29-127, 1978.

[Lawr, 1987] **Lawrence, D. (1987).** Genetic Algorithm and Simulated Annealing. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

[Laye, 2010] **Layeb, A.S. (2010).** Utilisation des Approches d'Optimisation Combinatoire pour la Vérification des Applications Temps Réel. Thèse de Doctorat, Université de Constantine, Algérie.

[Lepr, 2000] **Leprévost, F. (2000).** Les standards cryptographiques du XXIe siècle : AES et IEEE-P1363. Gazette des Mathématiciens - n°85.

[LinS, 1973] **Lin, S. & Kernighan, B.W. (1973).** An efficient heuristic for the traveling-salesman problem. Operations Research 21 : 498-516.

[Lobo, 2004] **Lobo, F.G. & Goldberg, D.E. (2004**). The parameter-less genetic algorithm in practice. Information Sciences, 167:217–232.

[Lobo, 2007] **Lobo, F.G. & Lima, C.F. (2007**). Parameter Setting in Evolutionary Algorithms : Adaptive Population Sizing Schemes in Genetic Algorithms. Springer Berlin / Heidelberg.

[Loid, 2005] **Loidreau, P. (2005).** Introduction à la cryptographie. LinuxFocus, Article № 243.

[Luke, 2010] **Luke. S. (2010).** Essentials of Metaheuristics.

[Magn, 2001] **Magnin, V. (2001).** Optimisation et algorithmes génétiques. Cyber-cour gratuit de l'EUDIL.
https://iris.univ-lille1.fr/dspace/bitstream/1908/499/2/

[Mait, 2011] **Maitre, O. (2011).** Parallélisation d'algorithmes évolutionnaires sur carte GPGPU. Thèse de Doctorat, Université de Strasbourg, France.

[Mats, 1994] **Matsui, M. (1994).** Linear cryptanalysis method for DES cipher. Advances in Cryptology**,** EUROCRYPT'93, volume 765 de Lecture Notes in Computer Science, Springer-Verlag.

[Mene, 1996] **Menezes, A.J., Van Oorschot, P.C. & Vanstone, S.A. (1996).** Handbook of Applied Cryptography. CRC Press.

[Mesg, 1999] **Mesghouni, K. (1999).** Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de production. Thèse de doctorat, Ecole centrale de lille et l'université de lille1.

[Mich, 1992] **Michalewicz, Z. (1992).** Genetic Algorithms + data structures = evolution programs, Springer Verlag, Berlin.

[Mich, 1996] **Michalewicz, Z. (1996).** Genetic Algorithms + Data Structures = Evolution Programs. 3rd Edition, Springer Verlag, Berlin.

[Mich, 2007] **Michalewicz, Z. & Schmidt, M. (2007**). Parameter Control in Practice, Studies in Computational Intelligence (SCI) 54. 277–294, Springer-Verlag Berlin Heidelberg.

[Monm, 2000] **Monmarché, N. (2000).** Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation. Thèse de Doctorat, Université de Tours, France.

[Nave, 2002] **Navez, P. & Van Assche, G. (2002).** Une transmission sécurisée : la cryptographie quantique.

[Omar, 2006] **Omary, F. (2006).** Application des algorithmes évolutionnistes à la cryptographie. Mémoire d'HDR, Université Mohamed V- Agdal, Rabat, Marroc.

[Omar, 2007] **Omary, F., Tragha, A.R, Bellaachia, A.G & Mouloudi, A.A. (2007).** Design and Evaluation of Two Symmetrical Evolutionist-Based Ciphering Algorithms. IJCSNS, International Journal of Computer Science and Network Security, VOL.7 No.2.

[Osma, 1996] **Osman, I.H. & Kelly, J.P. (1996).** Meta-heuristics: theory and applications. Kluwers Academic Publishers, Boston.

[Pear, 1984] **Pearl, J. (1984).** Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading.

[Prab, 1998] **Prabhu, D., Buckles, B. P., & Petry, F. E. (1998).** Genetic algorithms for scene interpretation from prototypical semantic description. Submitted to IEEE Transactions on Evolutionary Computation. 22 p.

[Preu, 2007] **Preuss, M. & Bartz-Beielstein, T. (2007**). Sequential Parameter Optimization Applied to Self-Adaptation for Binary-Coded Evolutionary Algorithms.

Studies in Computational Intelligence (SCI) **54**, 91–119, Springer-Verlag Berlin Heidelberg.

[Radc, 1995] **Radcliff, N.J. & Surry, P.D. (1995).** Fundamental limitations on search algorithms: evolutionary computing in perspective. Lecture Notes in Computer Science 1000, Springer-Verlag.

[Reev, 1993] **Reeves, C.R. (1993).** Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publications, Oxford.

[Rech, 2003] **Rechenberg, I. (2003).** Evolutions Strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart. In [Jour, 2003].

[Renn, 2000] **Rennard, J-P. (2000).** Introduction aux algorithmes génétiques. http://www.rennard.org/alife/french/gavintr.html

[Sema, 2009a] **Semassel, M.L, Souici, I. & Seridi, H. (2009).** Extension et Parallèlisation d'un Algorithme de Chiffrement Évolutionnaire basé Occurrences. CDROM, Conférence international sur l'Informatique et ses Applications, CIIA'09, 03-04 mai, Saida, Algérie. http://www.informatik.uni-trier.de/~ley/db/conf/ciia/ciia2009.html/.

[Sema, 2009b] **Semassel, M.L, Souici, I. & Seridi, H. (2009).** Extension et parallélisation d'un algorithme de chiffrement évolutionnaire basé occurrences. Colloque International Nouvelles Techniques Immuno-Cognitives dans les Réseaux Informatique, NTICRI'09, 10-11 Mai, Oran, Algérie.

[Shan, 1949] **Shannon, C. (1949).** Communication Theory of Secrecy Systems. Bell Systems Technical Journal.

[Soui, 2008a] **Souici, I., Seridi, H. & Aissaoui, M. Z. (2008).** Nouvel Algorithme de Chiffrement Evolutionniste ACEO (Algorithme de chiffrement Evolutionniste basé Occurrences)", Revue des Sciences, Technologie & Développement, N° 4, pp. 44, 56. ISSN : 1112 7309, http://www.andru.gov.dz/

[Soui, 2008b] **Souici, I., Aissaoui, Z. & Seridi, H. (2008).** Conception et Evaluation d'un Nouvel Algorithme Crypto-évolutionnaire (SSA). COSI'08. Colloque sur l'Optimisation et les Systèmes d'Information. Tizi-Ouzou, Algérie.

[Soui, 2009] **Souici, I., Benhamza, K. & Seridi, H. (2009).** Nouvel Algorithme Crypto-évolutionnaire (ACEO) : Conception et Evaluation. CDROM, 6ème Conférence sur le Génie Electrique (CGE'06), 13-14 Avril, Ecole Militaire Polytechnique, Bordj El Bahri, Algérie. http://www.emp.edu.dz/Manif_Scientifique/CGE06/CGE06.htm.

[Soui, 2010a] **Souici, I., Seridi, H., & Akdag, H. (2010).** Images evolutionary encryption. Premier Congrès International sur les modèles, Optimisation et Sécurité des Systèmes, ICMOSS'2010, 29-31 Mai, Tiaret, Algérie.

[Soui, 2010b] **Souici, I., Seridi, H., & Akdag, H. (2010).** Images encryption by the use of evolutionary algorithms. International Conference on Electrical Engineering, Electronics and Automatic'10, ICEEA'2010, 2-3 Novembre, Bejaia, Algérie.

[Soui, 2011a] **Souici, I & Seridi, H. (2011).** Une méthode rapide et efficace pour le cryptage évolutionnaire d'images. 8ème Colloque International sur l'Optimisation et les Systèmes d'Information, COSI'11, 24-27 avril, Guelma, Algérie. http://www.isima.fr/cosi/cosi2011/.

[Soui, 2011b] **Souici, I., Seridi, H. & Akdag, H. (2011).** Images Encrypton by the Use of Evolutionary algorithms. Analog Integrated Circuits and Signal Processing, Springer, Volume 69, Issue 1, pp 49-58, ISSN 1573-1979, Springer. http://www.springerlink.com/content/y17861290161146m/.

[Ster, 2004] **Stern, J., Granboulan, L., Nguyen, P. & Pointcheval, D. (2004).** Conception et preuves d'algorithmes Cryptographiques. Cours de magistère M.M.F.A.I, École normale supérieure.

[Stin, 1996] **Stinson, D. (1996).** Cryptographie, théorie et pratique. International Thomson Publishing, France.

[Tail, 1991] **Taillard, E.D. (1991).** Robust taboo search for the quadratic assignment problem. Parallel computing. Tome 17, p443-455.

[Tail, 1995] **Taillard, E.D. (1995).** Comparison of iterative searches techniques for the quadratic assignment problem. Tome 3, numéro 2, p87-105.

[Tail, 1999] **Taillard, E.D. (1999).** Some efficient heuristic methods for the flow shop sequencing problem. European journal of operational research, tome 47, numéro 1, p65-74.

[Talb, 1999] **Talbi, E.G. (1999).** Métaheuristiques pour l'optimisation combinatoire multi□objectif: Etat de l'art. CNET, PE: 98□757.33.

[Ulam, 1950] **Ulam, S. (1950).** Random Processes and Transformations. In: S. Ulam, Sets, Numbers and Universes, Cambridge: MIT Press, pp. 326-337, 1974.

[VobS, 1993] **Voß, S. (1993).** Intelligent Search. Manuscript, TU Darmstadt.

[VobS, 1999] **Voß, S., Martello, S., Osman, I.H & Roucairol, C. (1999).** Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston.

[VonN, 1966] **Von Neumann, J. (1966).** Theory of Self-Reproducing Automata. University of Illinois Press.

[Wolp, 1997] **Wolpert, D.H. & Macready, W. G. (1997).** No free lunch theorems for optimization. IEEE Transactions On Evolutionary Computation 1(1) : 67-82.

[Zimm, 2005] **Zimmermann, P. (2005).** Cryptographie asymétrique : signature et RSA en détail. 6ème cours du module : Introduction à la cryptologie, Master M1 Informatique, 2005.
http://www.loria.fr/~zimmerma/cours/id12-2005-6.pdf

[www1] http://wapedia.mobi/fr/Cryptanalyse

[www2] http://www.arrobe.fr/docs/cryptologie.pdf

[www3] http://www.bibmath.net/crypto/moderne/des.php3

[www4] http://www.scryptosystems.com/whitepapers_securite.htm

[www5] http://www.uqtr.ca/~delisle/Crypto/prives/blocs_idea.php

[www6] http://www.bibmath.net/crypto/moderne/rsa.php3

[www7] http://www.apprendre-en-ligne.net/crypto/moderne/pgp.html

[www8] news:fr.misc.cryptologie

[www9] http://fr.wikipedia.org/wiki/Cryptographie_quantique

[www10] www.google.com_url_q=http//ufrsciencestech.u-bourgogne.fr_master2009Tabou.

[www11] http: //ufrsciencestech.u-bourgogne.fr /master1/ mi1tc5/ CM2009/ Genetiques/ Genetique_1.pdf

[www12] http://www.irisa.fr/triskell/home_html.

REFERENCES BIBLIOGRAPHIQUES

[Aart, 1997] **Aarts, E.H.L. & Lenstra, J.K. (1997).** Local search in combinatorial optimization. John Wiley & Sons.

[Ayas, 2004] **Ayas, J. & André Viau, M. (2004).** La recherche Tabou. Notes de cours.

[Barn, 2004] **Barni, M. & Bartolini, F. (2004).** Watermarking Systems Engineering : Enabling Digital Assets Security and Other Applications. ISBN: 0-8247-4806-9, printed on acid-free paper.

[Barg, 2005] **Bargeton, A. & Devèze, B. (2005).** Les colonies de fourmis : apprentissage coopératif pour le problème du voyageur de commerce. Projet ANIMAT, Université Pierre et Marie Curie.

[Baud, 2005a] **Baudry, B., Fleurey, F., Jézéquel, J-M, & Le Traon, Y. (2005).** Automatic test, case optimization : A bacteriologic algorithm. IEEE software, 22(2):76-82.

[Baud, 2005b] **Baudry, B. (2005).** Assemblage testable et validation de composant. Thèse PhD, Université de Rennes 1.

[Baud, 2005c] **Baudry, B., Fleurey, F., Jézéquel, J-M, & Le Traon, Y. (2005).** from genetic to bacteriological algorithms for mutation-based testing : Research articles, Softw. Test. Verif Relia. 15 (2):73-96.

[Beck, 1992] **Beckers, R., Deneubourg, J. L., & Goss, S. (1992).** Trails and U-Turns in the Selection of a Path by the Ant Lasius Niger. J. Theor. Biol., 159 :397-415.

[Biha, 1991] **Biham, E. et Shamir, A. (1991).** Differential Cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1):3_72.

[Bona, 1994] **Bonabeau, E., & Theraulaz, G. (1994).** Intelligence collective. Edition Hermes Sciences, Paris, France.

[Burk, 2003] **Burke, E.K., Silva, J.D. Landa & Soubeiga, E. (2003).** Hyperheuristic Approaches for Multiobjective Optimisation. In Proceedings of the Fifth Metaheuristics International Conference, MIC 2003, August, Kyoto, Japan.

[Burk, 2005] **Burke, E. K., Silva, D. Landa & Soubeiga, E. (2005).** Multi-objective Hyperheuristic Approaches for Space Allocation and Timetabling. In Meta-heuristics: Progress as Real Problem Solvers (Eds. Baraki, T., Nonobe, K. & Yagiura, M.), pp.129-158, Springer.

[Camp, 2010] **Campesato, J-B. (2010).** Introduction à la physique quantique du xxie siècle : La cryptographie quantique. Cours UEL.

[Cler, 2004] **Clerc, M. (2004).** L'optimisation par essaims particulaires. Hermès Science.

[Coll, 2002] **Collette, Y. & Siarry, P. (2002).** Optimisation multiobjectif. Eyrolles.

[Cost, 2006] **Costanzo, A., Luong, T. & Guillaume, V.M. (2006).** Les métaheuristiques en optimisation combinatoire.

[Cowl, 2000] **Cowling, P., Kendall, G. & Soubeiga, E. (2000)**. A Hyperheuristic Approach to Scheduling a Sales Summit. Practice and Theory of Automated Timetabling III : Third International Conference (PATAT 2000) (Eds. Burke, E.K. & Erben, W.), August 2000, Konstanz, Germany, pp. 176-190.

[Cowl, 2002] **Cowling, P., Kendall, G. & Soubeiga, E. (2002).** Hyperheuristics: A Tool for Rapid Prototyping in Scheduling and Optimisation. sApplications of Evolutionary Computing: Proceeding of Evo Workshops 2002 (Eds. Cagoni,

S., Gottlieb, J., Hart, E., Middendorf, M. & Goenther, R.), April 3-4, Kinsale, Ireland, pp. 1-10.

[Davi, 1985] **Davis L, (1985).** Applying Adaptive Algorithms to Epistatic Domains. Proceedings of the International Joint Conference on Artificial Intelligence, pp162-164.

[DeJo, 2007] **De Jong, K. (2007**). Parameter Setting in EAs: a 30 Year Perspective, Studies in Computational Intelligence (SCI) 54, 1–18, Springer-Verlag Berlin Heidelberg.

[Dene, 1977] **Deneubourg, J. (1977).** Application de l'ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites. Insectes Sociaux, pp 117-130.

[Dene, 1983] **Deneubourg, J.L., Pasteels, J.M. & Verhaeghe, J.C. (1983).** Probabilistic behaviour in ants : a strategy of errors. Journal of Theoretical Biology, 105, pp.259-271.

[Dene, 1989] **Deneubourg, J.L. & Goss, S. (1989).** Collective patterns and decision-making. Ethology and Evolution, p 295-311.

[Dori, 1992] **Dorigo, M. (1992).** Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[Dori, 1999] **Dorigo, M., & Di Caro, G. (1999).** The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw Hill, London, UK, pages 11–32.

[Dori, 2003] **Dorigo, M. & Stutzle, T. (2003).** Handbook of Metaheuristics. Volume 57 of International series in operations research and management science, chapter The Ant Colony Optimization Metaheuristics : Algorithms, Applications and Advances. Kluwer Academic Publishers, Boston Hardbound.

[Dori, 2004] **Dorigo, M. & Stützle, T. (2004).** Ant colony optimization. MIT Press.

[Dréo, 2003] **Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2003).** Métaheuristiques pour l'optimisation difficile. Eyrolles.

[Dréo, 2004] **Dréo, J. (2004).** Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical. Thèse de Doctorat, Université de Paris 12, France, 166 p.

[Dupa, 2004] **Dupas, R. (2004).** Amélioration de performance des systèmes de production : apport des algorithmes évolutionnistes aux problèmes d'ordonnancement cycliques et flexibles. HDR, Université d'Artois.

[Eibe, 2007] **Eiben, A.E., Michalewicz, Z. Schoenauer, M. & Smith, J.E. (2007).** Parameter Control in Evolutionary Algorithms, Studies in Computational Intelligence (SCI) 54, 19–46, Springer-Verlag Berlin Heidelberg.

[Foge, 1966] **Fogel, L., Owens, A., & Walsh, M. (1966).** Artificial Intelligence Through Simulated Evolution. Wiley, J., Chichester, UK. In [Jour, 2003].

[Gant, 2001] **Ganteaut, A. et Lévy, F. (2001).** La cryptologie moderne. L'Armement, 73:76_83.

[Gher, 2004] **Ghernaouti-Hélie, S. (2004).** Sécurité informatique et réseaux, Editions Dunod.

[Glov, 1986a] **Glover, F. (1986).** Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 13 : 533-549.

[Glov, 1986b] **Glover, F. (1986).** Future paths for integer programming and links to artificial intelligence. Computing and operating research, tome 13.

[Glov, 1989] **Glover, F. (1989, 1990).** Tabu search, Part1 and Part2. ORSA journal on computing, tome 1 et tome 2.

[Glov, 1997] **Glover, F. & Laguna, M. (1997).** Tabu Search. Kluwer, Boston.

[Gold, 1989] **Goldberg, D.E. (1989).** Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, New York.

[Goss, 1989] **Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989).** Self-Organized Shortcuts in the Argentine Ant. Naturwissenchaften, 76 :579-581.

[Gren, 1986] **Grenfenslette J.J. (1986).** Optimization of control parameters for genetic algorithms. IEEE translation on system Man and cybernetics, Vol 16 №1, pp122-128.

[Gury, 2004] **Gury, S. & Rémond, N. (2004).** Cryptologie : ElGamal d'après Diffie-Hellman.

[Hert, 2003] **Hertz, A. & Widmer, M. (2003).** Guidelines for the use of meta-heuristics in combinatorial optimization. European Journal of Operational Research, tome 151.

[Holl, 1975] **Holland, J. (1975).** Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.

[Huff, 1952] **Huffman, D.A. (1952).** A method for the construction of minimum-redundancy codes. Proceedings of the I.R.E., pp 1098-1102. http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf

[JinK,1999] **Jin-Kao, H., Galinier, P. & Habib, M. (1999).** Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. Revue d'Intelligence Artificielle, Vol : No. 1999.

[Jour, 2003] **Jourdan, L. (2003).** Métaheuristiques pour l'extraction de connaissances : application à la génomique. Thèse de doctorat, U.S.T.L.

[Kamm, 2006] **Kammarti, R. (2006).** Approches évolutionnaires pour la résolution du 1-PDPTW statique et dynamique. Thèse de doctorat, délivrée conjointement par l'école centrale de lille et l'université des sciences et technologies de lille.

[Katz, 2000] **Katzenbeisser, S. & Petitcolas, A.P. (2000).** Information hiding techniques for steganography and digital watermarking, Artech House, Boston-London.

[Kerc, 1883] **Kerckhoffs, A. (1883).** La cryptographie militaire. Journal des sciences militaires, Janvier 1883.

[Koba, 1990] **Kobayashi, M. (1990).** DigitalWatermarking : Historical Roots. Technical report, IBM Research, Tokyo Research Laboratory. Japan.

[Koza, 1992] **Koza, J. (1992).** Genetic Programming. Cambridge, MA: MIT Press.

[Lafo, 2006] **Lafourcade, P. (2006).** Vérification de protocoles cryptographiques en présence de théories équationnelles. Mémoire d'HDR, École Normale Supérieur de Cachan, Cachan, France.

[Lapo, 1996] **Laporte, G. & Osman, I.H. (1996).** Metaheuristics in combinatorial optimization, Annals of Operations Research 63. J.C. Baltzer Science Publishers, Basel, Switzerland.

[Laur, 1978] **Lauriere, J.L. (1978).** A language and a program for stating and solving combinatorial problems, Artificial Intelligence 10 : 29-127, 1978.

[Lawr, 1987] **Lawrence, D. (1987).** Genetic Algorithm and Simulated Annealing. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

[Laye, 2010] **Layeb, A.S. (2010).** Utilisation des Approches d'Optimisation Combinatoire pour la Vérification des Applications Temps Réel. Thèse de Doctorat, Université de Constantine, Algérie.

[Lepr, 2000] **Leprévost, F. (2000).** Les standards cryptographiques du XXIe siècle : AES et IEEE-P1363. Gazette des Mathématiciens - n°85.

[LinS, 1973] **Lin, S. & Kernighan, B.W. (1973).** An efficient heuristic for the traveling-salesman problem. Operations Research 21 : 498-516.

[Lobo, 2004] **Lobo, F.G. & Goldberg, D.E. (2004**). The parameter-less genetic algorithm in practice. Information Sciences, 167:217–232.

[Lobo, 2007] **Lobo, F.G. & Lima, C.F. (2007**). Parameter Setting in Evolutionary Algorithms : Adaptive Population Sizing Schemes in Genetic Algorithms. Springer Berlin / Heidelberg.

[Loid, 2005] **Loidreau, P. (2005).** Introduction à la cryptographie. LinuxFocus, Article № 243.

[Luke, 2010] **Luke. S. (2010).** Essentials of Metaheuristics.

[Magn, 2001] **Magnin, V. (2001).** Optimisation et algorithmes génétiques. Cyber-cour gratuit de l'EUDIL.
https://iris.univ-lille1.fr/dspace/bitstream/1908/499/2/

[Mait, 2011] **Maitre, O. (2011).** Parallélisation d'algorithmes évolutionnaires sur carte GPGPU. Thèse de Doctorat, Université de Strasbourg, France.

[Mats, 1994] **Matsui, M. (1994).** Linear cryptanalysis method for DES cipher. Advances in Cryptology**,** EUROCRYPT'93, volume 765 de Lecture Notes in Computer Science, Springer-Verlag.

[Mene, 1996] **Menezes, A.J., Van Oorschot, P.C. & Vanstone, S.A. (1996).** Handbook of Applied Cryptography. CRC Press.

[Mesg, 1999] **Mesghouni, K. (1999).** Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de production. Thèse de doctorat, Ecole centrale de lille et l'université de lille1.

[Mich, 1992] **Michalewicz, Z. (1992).** Genetic Algorithms + data structures = evolution programs, Springer Verlag, Berlin.

[Mich, 1996] **Michalewicz, Z. (1996).** Genetic Algorithms + Data Structures = Evolution Programs. 3rd Edition, Springer Verlag, Berlin.

[Mich, 2007] **Michalewicz, Z. & Schmidt, M. (2007**). Parameter Control in Practice, Studies in Computational Intelligence (SCI) 54. 277–294, Springer-Verlag Berlin Heidelberg.

[Monm, 2000] **Monmarché, N. (2000).** Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation. Thèse de Doctorat, Université de Tours, France.

[Nave, 2002] **Navez, P. & Van Assche, G. (2002).** Une transmission sécurisée : la cryptographie quantique.

[Omar, 2006] **Omary, F. (2006).** Application des algorithmes évolutionnistes à la cryptographie. Mémoire d'HDR, Université Mohamed V- Agdal, Rabat, Marroc.

[Omar, 2007] **Omary, F., Tragha, A.R, Bellaachia, A.G & Mouloudi, A.A. (2007).** Design and Evaluation of Two Symmetrical Evolutionist-Based Ciphering Algorithms. IJCSNS, International Journal of Computer Science and Network Security, VOL.7 No.2.

[Osma, 1996] **Osman, I.H. & Kelly, J.P. (1996).** Meta-heuristics: theory and applications. Kluwers Academic Publishers, Boston.

[Pear, 1984] **Pearl, J. (1984).** Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, Reading.

[Prab, 1998] **Prabhu, D., Buckles, B. P., & Petry, F. E. (1998).** Genetic algorithms for scene interpretation from prototypical semantic description. Submitted to IEEE Transactions on Evolutionary Computation. 22 p.

[Preu, 2007] **Preuss, M. & Bartz-Beielstein, T. (2007**). Sequential Parameter Optimization Applied to Self-Adaptation for Binary-Coded Evolutionary Algorithms.

Studies in Computational Intelligence (SCI) **54**, 91–119, Springer-Verlag Berlin Heidelberg.

[Radc, 1995] **Radcliff, N.J. & Surry, P.D. (1995).** Fundamental limitations on search algorithms: evolutionary computing in perspective. Lecture Notes in Computer Science 1000, Springer-Verlag.

[Reev, 1993] **Reeves, C.R. (1993).** Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publications, Oxford.

[Rech, 2003] **Rechenberg, I. (2003).** Evolutions Strategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart. In [Jour, 2003].

[Renn, 2000] **Rennard, J-P. (2000).** Introduction aux algorithmes génétiques. http://www.rennard.org/alife/french/gavintr.html

[Sema, 2009a] **Semassel, M.L, Souici, I. & Seridi, H. (2009).** Extension et Parallèlisation d'un Algorithme de Chiffrement Évolutionnaire basé Occurrences. CDROM, Conférence international sur l'Informatique et ses Applications, CIIA'09, 03-04 mai, Saida, Algérie. http://www.informatik.uni-trier.de/~ley/db/conf/ciia/ciia2009.html/.

[Sema, 2009b] **Semassel, M.L, Souici, I. & Seridi, H. (2009).** Extension et parallélisation d'un algorithme de chiffrement évolutionnaire basé occurrences. Colloque International Nouvelles Techniques Immuno-Cognitives dans les Réseaux Informatique, NTICRI'09, 10-11 Mai, Oran, Algérie.

[Shan, 1949] **Shannon, C. (1949).** Communication Theory of Secrecy Systems. Bell Systems Technical Journal.

[Soui, 2008a] **Souici, I., Seridi, H. & Aissaoui, M. Z. (2008).** Nouvel Algorithme de Chiffrement Evolutionniste ACEO (Algorithme de chiffrement Evolutionniste basé Occurrences)", Revue des Sciences, Technologie & Développement, N° 4, pp. 44, 56. ISSN : 1112 7309, http://www.andru.gov.dz/

[Soui, 2008b] **Souici, I., Aissaoui, Z. & Seridi, H. (2008).** Conception et Evaluation d'un Nouvel Algorithme Crypto-évolutionnaire (SSA). COSI'08. Colloque sur l'Optimisation et les Systèmes d'Information. Tizi-Ouzou, Algérie.

[Soui, 2009] **Souici, I., Benhamza, K. & Seridi, H. (2009).** Nouvel Algorithme Crypto-évolutionnaire (ACEO) : Conception et Evaluation. CDROM, 6ème Conférence sur le Génie Electrique (CGE'06), 13-14 Avril, Ecole Militaire Polytechnique, Bordj El Bahri, Algérie. http://www.emp.edu.dz/Manif_Scientifique/CGE06/CGE06.htm.

[Soui, 2010a] **Souici, I., Seridi, H., & Akdag, H. (2010).** Images evolutionary encryption. Premier Congrès International sur les modèles, Optimisation et Sécurité des Systèmes, ICMOSS'2010, 29-31 Mai, Tiaret, Algérie.

[Soui, 2010b] **Souici, I., Seridi, H., & Akdag, H. (2010).** Images encryption by the use of evolutionary algorithms. International Conference on Electrical Engineering, Electronics and Automatic'10, ICEEA'2010, 2-3 Novembre, Bejaia, Algérie.

[Soui, 2011a] **Souici, I & Seridi, H. (2011).** Une méthode rapide et efficace pour le cryptage évolutionnaire d'images. 8ème Colloque International sur l'Optimisation et les Systèmes d'Information, COSI'11, 24-27 avril, Guelma, Algérie. http://www.isima.fr/cosi/cosi2011/.

[Soui, 2011b] **Souici, I., Seridi, H. & Akdag, H. (2011).** Images Encrypton by the Use of Evolutionary algorithms. Analog Integrated Circuits and Signal Processing, Springer, Volume 69, Issue 1, pp 49-58, ISSN 1573-1979, Springer. http://www.springerlink.com/content/y17861290161146m/.

[Ster, 2004] **Stern, J., Granboulan, L., Nguyen, P. & Pointcheval, D. (2004).** Conception et preuves d'algorithmes Cryptographiques. Cours de magistère M.M.F.A.I, École normale supérieure.

[Stin, 1996] **Stinson, D. (1996).** Cryptographie, théorie et pratique. International Thomson Publishing, France.

[Tail, 1991] **Taillard, E.D. (1991).** Robust taboo search for the quadratic assignment problem. Parallel computing. Tome 17, p443-455.

[Tail, 1995] **Taillard, E.D. (1995).** Comparison of iterative searches techniques for the quadratic assignment problem. Tome 3, numéro 2, p87-105.

[Tail, 1999] **Taillard, E.D. (1999).** Some efficient heuristic methods for the flow shop sequencing problem. European journal of operational research, tome 47, numéro 1, p65-74.

[Talb, 1999] **Talbi, E.G. (1999).** Métaheuristiques pour l'optimisation combinatoire multi□objectif: Etat de l'art. CNET, PE: 98□757.33.

[Ulam, 1950] **Ulam, S. (1950).** Random Processes and Transformations. In: S. Ulam, Sets, Numbers and Universes, Cambridge: MIT Press, pp. 326-337, 1974.

[VobS, 1993] **Voß, S. (1993).** Intelligent Search. Manuscript, TU Darmstadt.

[VobS, 1999] **Voß, S., Martello, S., Osman, I.H & Roucairol, C. (1999).** Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston.

[VonN, 1966] **Von Neumann, J. (1966).** Theory of Self-Reproducing Automata. University of Illinois Press.

[Wolp, 1997] **Wolpert, D.H. & Macready, W. G. (1997).** No free lunch theorems for optimization. IEEE Transactions On Evolutionary Computation 1(1) : 67-82.

[Zimm, 2005] **Zimmermann, P. (2005).** Cryptographie asymétrique : signature et RSA en détail. 6ème cours du module : Introduction à la cryptologie, Master M1 Informatique, 2005.
http://www.loria.fr/~zimmerma/cours/id12-2005-6.pdf

[www1] http://wapedia.mobi/fr/Cryptanalyse

[www2] http://www.arrobe.fr/docs/cryptologie.pdf

[www3] http://www.bibmath.net/crypto/moderne/des.php3

[www4] http://www.scryptosystems.com/whitepapers_securite.htm

[www5] http://www.uqtr.ca/~delisle/Crypto/prives/blocs_idea.php

[www6] http://www.bibmath.net/crypto/moderne/rsa.php3

[www7] http://www.apprendre-en-ligne.net/crypto/moderne/pgp.html

[www8] news:fr.misc.cryptologie

[www9] http://fr.wikipedia.org/wiki/Cryptographie_quantique

[www10] www.google.com_url_q=http//ufrsciencestech.u-bourgogne.fr_master2009Tabou.

[www11] http: //ufrsciencestech.u-bourgogne.fr /master1/ mi1tc5/ CM2009/ Genetiques/ Genetique_1.pdf

[www12] http://www.irisa.fr/triskell/home_html.