وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار- عنابة

Année: 2019

Faculté: Sciences de l'ingéniorat
Département: Electronique

MEMOIRE
Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé

ETUDE DE QUELQUES VARIANTES DE CHIFFREMENT D'IMAGE PAR RSA

Domaine : Sciences et Technologie

Filière : Télécommunications

Spécialité : Réseaux et Télécommunications

Par: Mr. DIA Aly Seydou

DEVANT Le JURY

Président	**: Mr. T. HAFS**	**M.C.B**	**Université Annaba**
Directeur de mémoire	**: Mr. M. KADDECHE**	**Pr.**	**Université Annaba**
Examinateur	**: Mr. N. DOGHMANE**	**Pr.**	**Université Annaba**
Examinateur	**: Yahi**	**MCA**	**Université Annaba**

Dédicaces

En la mémoire de Trospar Ndudzo

Etudiant Zimbabwéen en Master 2 Electrotechnique

Assassiné le 10/02/2019 à Annaba, victime de la barbarie humaine.

Remerciements

Je remercie en premier lieu ALLAH le tout puissant, le très miséricordieux qui m'a permis d'atteindre ce niveau d'études et grâce à qui toute chose se réalise.

Deuxièmement je tiens à remercier ma famille, mes parents, amis et proches qui m'ont toujours soutenu depuis l'enfance et qui m'ont toujours accompagné par leurs prières et leurs attachements à ma réussite.

En troisièmement lieu je remercie l'ensemble du corps professoral du département d'électronique de l'Université Badji Mokhtar d'Annaba qui n'ont jamais ménagé aucun effort pour notre formation ; plus particulièrement mon encadreur Monsieur le Professeur KADDECHE Mohamed pour l'orientation, et l'aide précieuse qu'il m'a offert durant toute l'année et aussi pour sa constante disponibilité, qu'il trouve dans ce travail un hommage vivant à l'endroit de sa haute personnalité.

Mention toute particulière à mes professeurs Mr DOGHMANE Noureddine, Mr HAFS Toufik, et Mr KOUADRIA Nasreddine, puissent ces quelques lignes traduire l'immense expression de ma gratitude éternelle à leurs égards respectifs.

Enfin mes très sincères remerciement aux autorités de l'Algérie, qui à travers la coopération algério-malienne m'ont permis d'effectuer mes études supérieures dans un cadre idyllique. Je suis fier et honoré d'être diplômé de l'Université Badji Mokhtar d'Annaba.

Résumé :

Le transfert de données importantes ne peut se faire avec un risque et doit donc être protéger. Pour protéger ces types de communication, il faut crypter ces données. Malgré toutes ses évolutions et le développement de nouveaux algorithmes cryptographiques, beaucoup d'entre eux ne résistent pas aux attaques.

Ainsi, notre travail de mémoire consiste au chiffrement par l'algorithme RSA d'images. L'image numérique est représentée par les niveaux de gris allant de 0 à 255 (8 bits), c'est-à-dire travaillant en modulo 256. Il est impossible de trouver deux nombres premiers tel que leur produit est 256. Pour pallier aux conditions d'application de cet algorithme de chiffrement (Déchiffrement), nous proposons quelques méthodes.

Les méthodes proposées ont été évalués et comparés entre elles et avec les principaux algorithmes standards de cryptage. Une de ces méthodes proposées donne de bons résultats de chiffrement relatifs soit à leur EQM ou à leur PSNR.

Mots clés : Cryptage, RSA, Image numérique, Algorithme Bezout-Euclide, EQM, PSNR

Abstract

The purpose of modern cryptography is to address more generally the problems of communications security and to provide a number of security services such as the confidentiality, integrity and authenticity of transmitted data. Modern cryptography consists of two main parts: Symmetric cryptography and asymmetric cryptography.

In this paper, we present one of the modern cryptographic techniques called RSA Protocol. The simulation of this algorithm is applied to different images from databases.

The digital image is represented by gray levels ranging from 0 to 255 (8 bits), that is, working in modulo 256. It is impossible to find two prime numbers such that their product is 256. To overcome the conditions of application of the RSA encryption algorithm to image, we propose some methods.

Keywords : RSA, Encryption, Picture, EQM, PNSR

Liste des Abréviations

- **RSA :** Du nom de ses inventeurs ron **R**ivest, adi **S**hamir et len **A**ldeman.
- **PSNR** : Peak Signal to Noise Ratio.
- **EQM** : Ecart Quadratique Moyenne.
- **RVB :** Rouge, Vert, Bleu.
- **DPI :** Dot Per Inch.
- **PPP :** Point Par Pouce.
- **SVG** : Scalable Vector Graphics.
- **ODF** : Open Document Format.
- **BMP :** BitMaP.
- **TIFF :** Tagged Image File Format.
- **RLE :** Run length encoding.
- **LZW :** Lempel Ziv Welch.
- **GIF :** Graphic Interchange Format.
- **PNG :** Portable Network Graphics.
- **JPEG :** Joint Picture Expert Group.
- **ROT13 :** ROTation de 13 lettres.
- **DES :** Data Encryption Standard.
- **I.B.M :** International Business Machines.
- **N.S.A :** National Security Agency.
- **PGCD :** Plus Grand Commun Diviseur.

Liste des Figures

Liste des Tableaux

Sommaire

Table des matières

Introduction Générale

INTRODUCTION GENERALE

Avec le développement des technologies d'Internet et de la communication, la transmission des images joue un rôle très important dans la transmission de l'information. Cependant, la sécurité de l'information est un sujet sensible pour la recherche, le cryptage est l'une des meilleures alternatives qui s'est avérée efficace tout au long de l'histoire pour assurer la confidentialité et la sécurité de l'information.

Le crypto système introduit par Rivest Shamir Adleman ou RSA est un algorithme asymétrique de cryptographie à clé publique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet.

Cependant, lors de son application aux images, un problème se pose est que le nombre 256 (Octet, 8 bits) qui est le modulo du chiffrement et déchiffrement du RSA n'est pas décomposable en produit de deux nombres premiers entre eux. C'est-à-dire, trouver deux nombre p et q tel p*q = 256 avec p et q premiers entre eux.

Dans ce mémoire, nous proposons quelques approches de chiffrement d'images qui est un chiffrement par blocs.

La première idée est d'utiliser le premier nombre inférieur à 256 et tel qu'il est décomposable en deux nombres p et q premiers entre eux. **Le nombre est N1 = 253 et l**es nombres premiers p et q tels que leur produit est égal à N1 sont : p1 = 11 et ; q1 = 23. Dans ce cas le chiffrement et déchiffrement de bloc d'un octet se fait en modulo 253.

La seconde proposition est l'utilisation le premier nombre supérieur à 256 et qui est décomposable entre deux nombres p et q premiers entre eux. **Le nombre est N2 = 259 et** les nombres premiers p2 et q2 tels que leur produit est égal à N2 sont : p2 = 07 ; q2 = 37. Dans ce cas le chiffrement et déchiffrement de bloc d'un octet se fait en modulo 259.

Les troisième et quatrième approche sont une combinaison des résultats données par les méthodes une et deux.

- La moyenne des deux approches précédentes (moyenne entre les chiffrés)
- La valeur la plus proche de 256 en restant inférieure à 256 des résultats donnés par les chiffrements des deux approches précédentes.

Enfin pour valider notre travail, les valeurs des EQM, PSNR entre l'image originale et celle chiffrée donnés par simulation seront comparées à d'autres travaux de recherches effectués.

Nous avons structuré notre mémoire en quatre chapitres :

Le premier chapitre est une brève présentation sur les techniques de cryptographie classiques et modernes ainsi que leurs classifications.

Dans le deuxième chapitre nous abordons les notions relatives aux images numériques (définition, numérisation, différents formats etc.) et quelques critères d'évaluation de la qualité d'images.

Au troisième chapitre nous détaillons la méthode utilisée pour le chiffrement d'image dans ce présent mémoire, (le RSA) en expliquant son principe et en présentant quelques exemples.

Le quatrième et dernier chapitre consiste à présenter, analyser, et discuter les résultats obtenus.

Puis on va terminer par une conclusion générale et quelques perspectives pouvant aider dans l'amélioration du système dans le futur.

Chapitre I :
Cryptographie

I.1. Introduction

Le mot « cryptographie » est un mot d'origine grecque composé de deux parties : « cryptos », qui signifie caché et « logos » qui signifie mot. La cryptographie et la cryptanalyse forment les deux disciplines de la cryptologie ; la cryptographie visant à crypter les messages et la cryptanalyse à les décrypter. La cryptologie est un art ancien et une science nouvelle : un art ancien car Jule César l'utilisait déjà, une science nouvelle parce que ce n'est un thème de recherche scientifique académique, c'est-à-dire universitaire, que depuis les années 1970. [1]

I.2. Définition de la cryptographie

La cryptographie est l'art de chiffrer, coder les messages est devenue aujourd'hui une science à part entière. Au croisement des mathématiques, de l'informatique, et parfois même de la physique, elle permet ce dont les civilisations ont besoin depuis qu'elles existent : le maintien du secret. Elle est donc l'étude des méthodes permettant de transmettre des données de manière confidentielle.

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité.

I.3. Quelques techniques de cryptographie classique

Contrairement à ce que l'on peut penser, la cryptographie n'est pas seulement une technique moderne, ni un produit de l'ère informatique. En effet de tout temps, les hommes ont ressenti le besoin de cacher des informations confidentielles. Bien évidemment depuis ses débuts la cryptographie a grandement évolué. Au cours des siècles, de nombreux systèmes de chiffrage ont été inventés, tous de plus en plus perfectionnés, et il est vrai que l'informatique y a beaucoup contribué. Mais au commencement les algorithmes étaient loin d'être aussi complexes et astucieux qu'à notre époque. La majeure partie des méthodes d'antan reposait sur deux principes fondamentaux :

- **La substitution** (remplacer certaines lettres par d'autres).
- **La transposition** (permuter des lettres du message afin de le brouiller).

I.3.1. Système de César

L'un des systèmes les plus anciens et les plus simples est le codage par substitution mono alphabétique (ou alphabets désordonnés). Il consiste à remplacer chaque lettre par une lettre différente. Il existe donc grâce à cette technique 26 façons de coder un message, ce qui fait que ce système a été longtemps utilisé par les armées pendant l'antiquité. Ce procédé très fiable à l'époque est tout de même problématique car il nécessite que les interlocuteurs se souviennent tous deux de la clef. De plus, il est évident que la sûreté de ce codage est quasi nulle et qu'il pourrait être déchiffré par n'importe quelle personne qui y mettrait le temps nécessaire. [2]

Texte clair	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Texte codé	W	X	E	H	Y	Z	T	K	C	P	J	I	U	A	D	G	L	Q	M	N	R	S	F	V	B	O

Tableau 1 : Un exemple de substitution.

La méthode la plus ancienne admise par l'histoire (par substitution alphabétique) est le non moins connu code de César, consistant en un décalage simple de lettres. Par substitution si l'on remplace le A par le C, alors le B devient D, le D un F, etc.… César utilisait ce code simple pour transmettre via un message des consignes à ces généraux d'armées sans qu'ils puissent être exploité par un quelconque ennemi dans le cas où le message serait intercepté. Malheureusement il n'y a que 26 façons différentes de chiffrer à l'aide de ce code ce qui en fait un code très peu sûr. Mais ce qui est d'autant plus insolite, c'est le fait que ce code de « César » est encore utilisé de nos jours sur Internet avec le ROT13 (rotation de 13 lettres) qui consiste à cacher des messages afin qu'ils ne soient pas lus involontairement, comme par exemple s'ils dévoilent le dénouement d'un film ou encore qui donne la réponse à une devinette.

Notons que Jule César (100 – 44 av. J.-C) était un général, homme politique et écrivain romain.

I.3.2. Système de Vigenère

Un autre système de cryptographie des plus anciens est cette fois-ci, la substitution poly alphabétique, qui utilise plusieurs alphabets décalés pour crypter un message. L'algorithme de substitution poly alphabétique le plus connu est le chiffre de Vigenère, mis au point par Blaise de Vigenère en 1586, qui fut utilisé pendant plus de 3 siècles. Son chiffre consiste à utiliser le chiffre de César, mais en changeant le décalage à chaque fois. Il utilise alors un

carré composé de 26 alphabets alignés, décalés de colonne en colonne d'un caractère.

Il place également au-dessus de ce carré, un alphabet pour la clef et à sa gauche un autre alphabet pour le texte à coder. Il suffit alors, pour chiffrer un message, de choisir un mot de longueur quelconque, de l'écrire sous le message à coder (de façon répétée s'il le faut) et de regarder dans le tableau l'intersection de la lettre à coder et de la lettre de la clef. [2]

Blaise de Vigenère (1523 – 1596) était un diplomate français, son chiffrement est une amélioration décisive du chiffrement de César.

Pour mieux comprendre le fonctionnement du Carré de Vigenère nous vous proposons cet exemple :

Supposons que nous voulons coder le texte {**CARRE DE VIGENERE**} avec la clef

{**MALICE**}

On commence par écrire la clef sous le texte à coder :

C	A	R	R	E		D	E		V	I	G	E	N	E	R	E
M	A	L	I	C		E	M		A	L	I	C	E	M	A	L

Tableau 2 : Cryptage par carre de Vigenère

Pour coder la lettre C, la clef est donnée par la lettre M. On regarde dans le tableau l'intersection de la ligne donnée par le C, et de la colonne donnée par le M. On trouve O. Puis on continue, jusqu'à ce qu'on ait fini de chiffrer notre texte. En chiffrant le texte « Carre de Vigenère », on obtient donc le texte « OAUZG HG VTOGRQRP ». Cet algorithme de cryptographie ainsi que celui de César sont les premiers des algorithmes à clef privée.

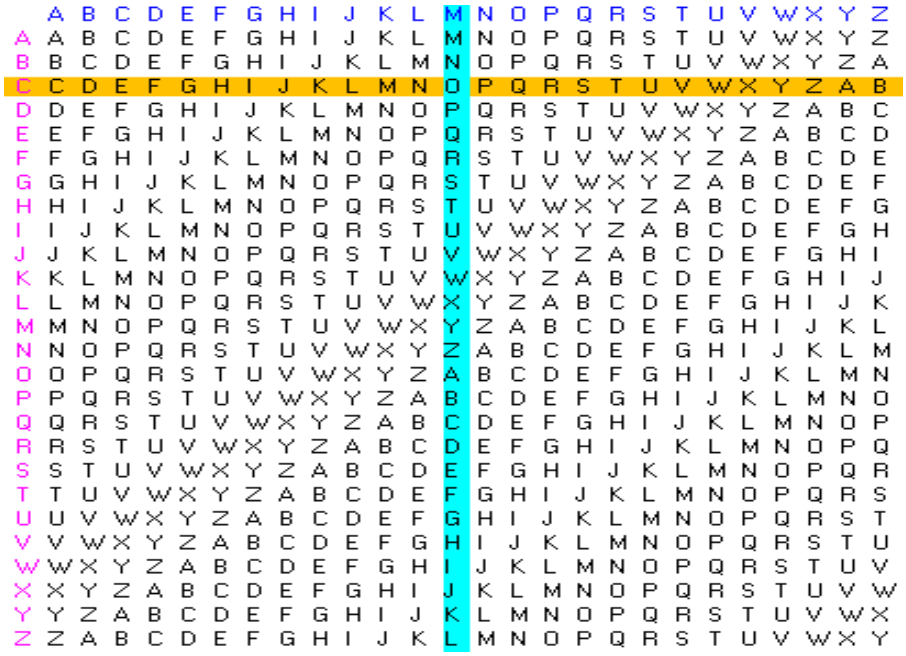


Figure 1 : Cryptage par carre de Vigenère

I.3.3. Système de Playfair

Il existe d'autres systèmes presque aussi anciens basés également sur des techniques par substitution mais moins connus que ceux vus précédemment. Il s'agit des systèmes par substitution de poly grammes. En effet au lieu de substituer des caractères, on substitue par exemple des diagrammes (des groupes de lettres le plus souvent). Le système de {**Playfair**} fut inventé par Sir Charles Wheatstone. Cet algorithme remplace chaque paire de lettre du texte en clair par une autre paire. Il utilise pour cela une table (matrice) carrée de coté 5, construite à partir d'une clef, qui contient toutes les lettres de l'alphabet hormis une (souvent le J par similitude avec le I). Chaque couple de lettre donne les coordonnées d'un rectangle dans la matrice. On remplace donc ce couple par les lettres formant les deux autres coins du rectangle. Si les deux lettres du couple sont sur la même ligne, on prend les deux lettres suivantes. Si elles sont sur la même colonne, on prend les 2 lettres du dessous. Si les 2 lettres sont identiques, on intercale entre elles une lettre convenue à l'avance (X ou Y).

Malheureusement, ce chiffre ingénieux ne fut pas utilisé souvent en raison du fait qu'il se déchiffre aisément en regardant quel couple de lettres apparaît le plus souvent dans le texte chiffré, et en supposant qu'ils représentent le couple de lettres le plus courant.

I.4 Cryptosystèmes modernes

Les méthodes utilisées de nos jours sont plus complexes, cependant la philosophie reste

la même. La différence fondamentale est que les méthodes modernes (les algorithmes, puisque l'on utilise maintenant des ordinateurs) manipulent directement des bits contrairement aux anciennes méthodes qui opéraient sur des caractères alphabétiques.

Ce n'est donc qu'un changement de taille (ou de représentation), puisque l'on utilise plus que deux éléments au lieu des 26 lettres de l'alphabet. La plupart des bons systèmes de cette catégorie combinent toujours des substitutions et des transpostions, et les règles sont connues de tous (principe de Kerckhoffs), c'est pourquoi on appelle cette classe : le chiffrement à usage général. La sécurité de ces méthodes repose maintenant sur un nouveau concept : les clés.

I.4.1 Cryptographie à clefs privés

La cryptographie à clefs privées, appelée aussi cryptographie symétrique est utilisée depuis déjà plusieurs siècles. C'est l'approche la plus authentique du chiffrement de données et mathématiquement la moins problématique. La clef servant à chiffrer les données peut être facilement déterminée si l'on connaît la clef servant à déchiffrer et vice-versa. Dans la plupart des systèmes symétriques, la clef de chiffrement et la clef de déchiffrement sont une seule et même clef.

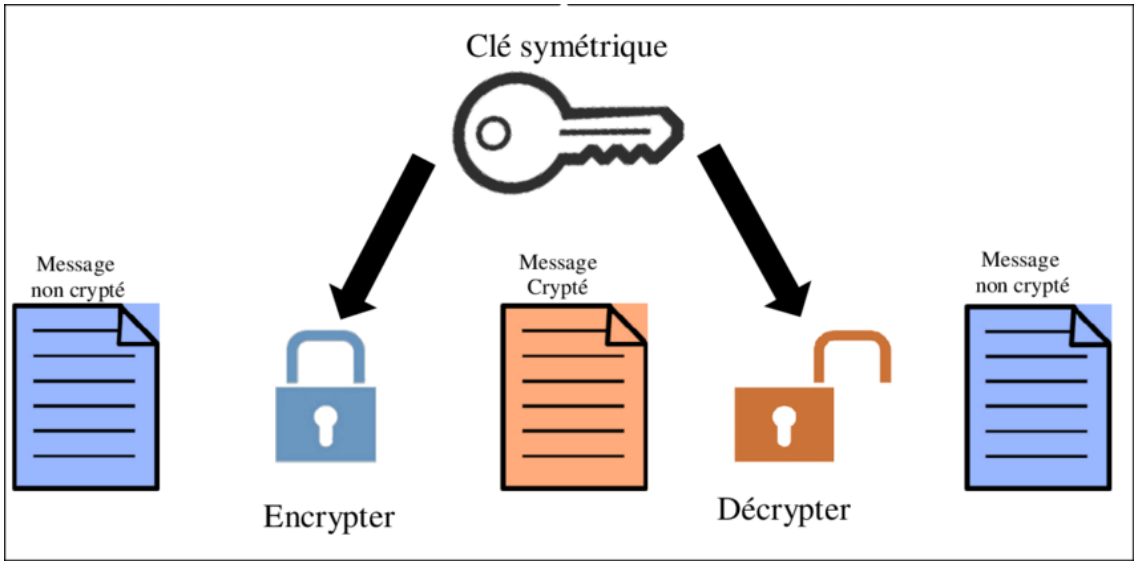


Figure 2 : Cryptographie symétrique

Les principaux types de cryptosystèmes à clefs privés utilisés aujourd'hui se répartissent en deux grandes catégories :

- Les cryptosystèmes par flots
- Les cryptosystèmes par blocs

A/ Chiffrement par blocs

Dans ce type de chiffrement, il y a une séparation du texte clair en blocs d'une longueur fixe selon un alphabet. L'idée générale du chiffrement par blocs est la suivant :

- Remplacer les caractères par un code binaire
- Découper cette chaîne en blocs de longueur donnée
- Chiffrer un bloc en l'additionnant bit par bit à une clef.
- Déplacer certains bits du bloc.
- Recommencer éventuellement un certain nombre de fois l'opération 3.
- Passer au bloc suivant et retourner au point 3 jusqu'à ce que tout le message soit chiffre.

A.1/ Réseau de Feistel

Le réseau de Feistel est une construction utilisée dans les algorithmes de chiffrements par blocs. Dans ce système de chiffrement, un bloc de texte en clair d'un nombre pair de bits est découpé en deux. La transformation de ronde est appliquée à une des deux moitiés, et le résultat est combiné avec l'autre moitié par un XOR. Les deux moitiés sont alors inversées pour la ronde suivante. [3]

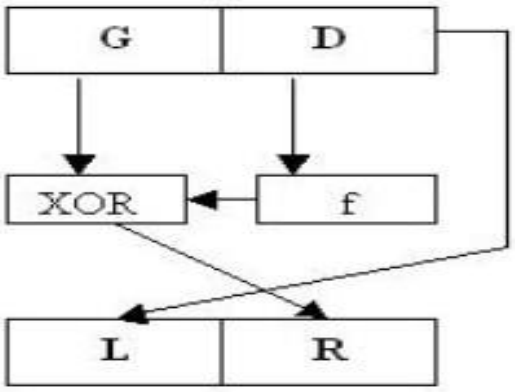


Figure 3 : Réseau de Feistel

A.2/ L'algorithme du DES

Le D.E.S. (ou Data Encryption Standard) naît en 1975 suite à une requête d'I.B.M. en 1960 pour son programme de recherche sur le chiffrement informatique. Au début, les spécialistes de la N.S.A. (National Security Agency, le service de sécurité intérieure américaine) se cassent les dents dessus donc I.B.M. est contraint de l'utiliser sous une forme plus simple que prévu. L'utilisation du D.E.S. se généralise alors peu à peu dans les administrations américaines. Depuis, le D.E.S. est remis à niveau tous les 5 ans environ pour

faire face à la puissance croissante des ordinateurs qui le mettent en péril. [4]

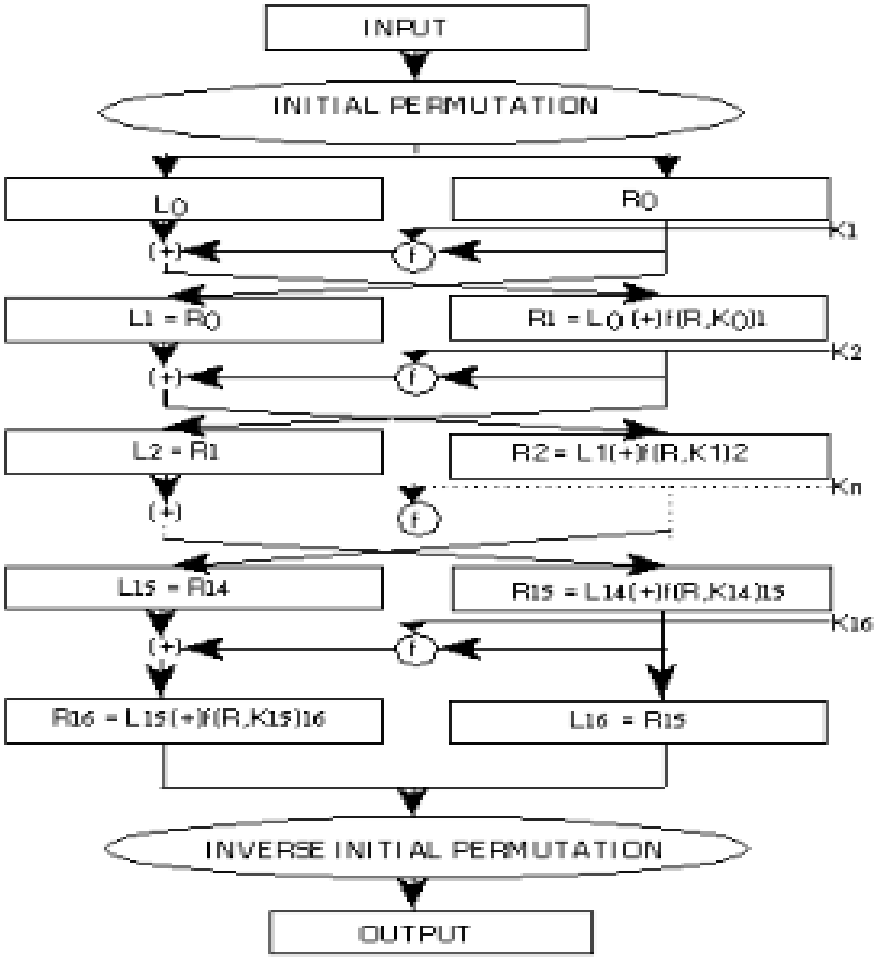


Figure 4 : L'algorithme du DES

Les grandes lignes de l'algorithme sont les suivantes :

- Fractionnement du texte en blocs de 64 bits (8 octets).
- Permutation initiale des blocs.
- Découpage des blocs en deux parties : gauche et droite, nommées G et D.
- Etapes de permutation et de substitution répétées 16 fois (appelées rondes).
- Recollement des parties gauche et droite puis permutation initiale inverse.

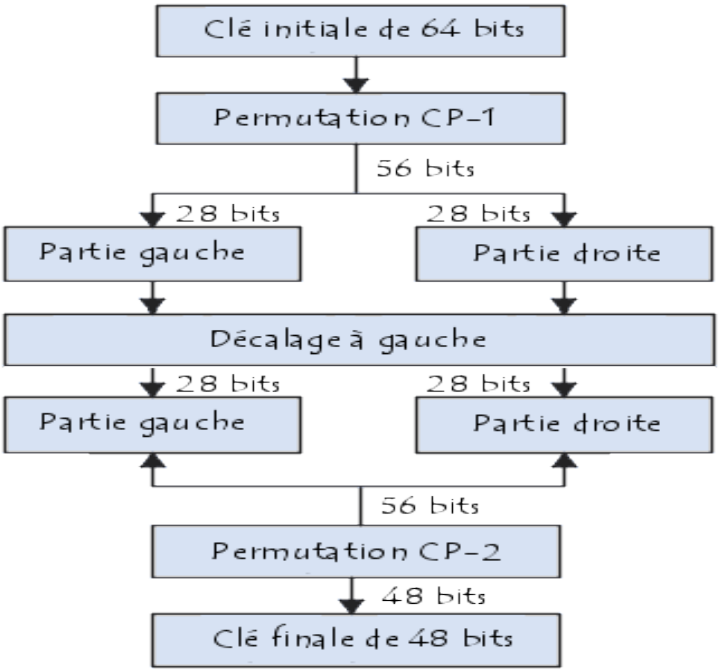


Figure 5 : Génération des clés DES

B/ Chiffrement par flot

Le chiffrement par flot est un chiffrement a clé symétrique permettant de traiter des donnés de longueur quelconque. Les bits du texte clair sont généralement combinés par opération XOR avec un flux de bits pseudo-aléatoire. [5]

Un des algorithmes de chiffrement par flot le plus repandu est le **RC4**, il a été conçu en 1987 par Ronald Rivest.

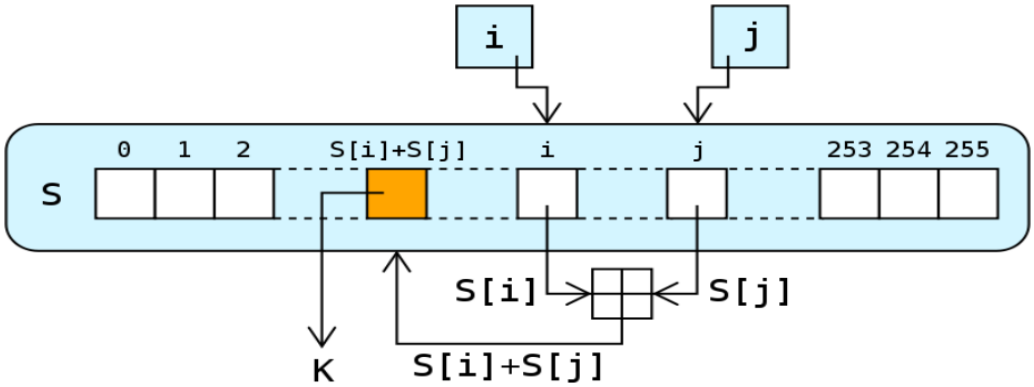


Figure 6 : Chiffrement par flot

2.3.2 Cryptographie à clefs publiques

Tous les algorithmes évoqués jusqu'à présent sont symétriques en ce sens que la même clef est utilisée pour le chiffrement et le déchiffrement. Le problème essentiel de la cryptographie symétrique est la distribution des clefs : pour que n personnes puissent

communiquer de manière confidentielle, il faut n (n-1)/2 clefs. L'idée de base des cryptosystèmes à clefs publiques a été proposée dans un article fondamental de Diffie et Hellman en 1976. [2]

Le principe fondamental est d'utiliser des clefs de chiffrement et déchiffrement différentes, non reconstructibles l'une à partir de l'autre :

- Une clef publique pour le chiffrement.
- Une clef secrète pour le déchiffrement.

A/ Cryptage RSA

L'algorithme le plus célèbre d'algorithme à clef publique a été inventé en 1977 par Ron Rivest, Adi Shamir et Len Adleman, à la suite de la publication de l'idée d'une cryptographie à clef publique par Diffie et Hellman. Il fut appelé **RSA**, des initiales de ces inventeurs.

RSA est basé sur la difficulté de factoriser un grand nombre en produit de deux grands facteurs premiers.

L'algorithme fonctionne de la manière suivante :

Imaginons que Bob souhaite recevoir d'Alice des messages en utilisant **RSA**.

- **Génération des clefs :**

a. p et q, deux grands nombres premiers sont générés au hasard grâce à un algorithme de test de primalité probabiliste, avec **n = p*q**.

b. Un nombre entier e premier avec **(p-1)*(q-1)** est choisi. Deux nombres sont premiers entre eux s'ils n'ont pas d'autre facteur commun que **1**.

c. L'entier d est l'entier de l'intervalle [2, (p-1)*(q-1)] telle que **ed** soit congrue à **1 modulo (p-1)*(q-1),** c'est-à-dire tel que **ed-1** soit un multiple de **(p-1)*(q-1).**

- **Distribution des clefs :**

Le couple **(n, e)** constitue la clef publique de Bob. Il la rend disponible à Alice en lui envoyant ou en la mettant dans un annuaire. Le couple **(n, d)** constitue quant à lui sa clef privée.

- **Chiffrement du message :**

Pour crypter le message Alice représente le message sous la forme d'un ou plusieurs entiers **M** compris entre 1 et n-1. Elle calcule **$C = M^e \bmod n$** grâce à la clef publique (n, e) de Bob et envoie **C** à Bob.

- **Déchiffrement du message :**

Bob reçoit **C** et calcule grâce à sa clef privée $\mathbf{C^d \bmod n}$. Il obtient ainsi le message initial **M.**

B/ L'algorithme Diffie-Hellman

Parallèlement à leur découverte du principe de la cryptographie à clé publique, Diffie et Hellman ont proposé en 1976 un protocole d'échange de clés totalement sécurisé.

L'objectif de Diffie-Hellman est de permettre l'établissement d'une clé privée entre deux parties, via l'échange de messages sur un canal non sécurisé. Lors de l'établissement d'une clé avec Diffie-Hellman, les messages sont en effet envoyés en clair sur le réseau, et toute personne qui intercepte les messages transmis ne doit pas pouvoir en déduire la clé générée. [6]

Supposons qu'Alice et Bob souhaitent se mettre d'accord sur une clé privée. L'algorithme Diffie-Hellman permet l'établissement de cette clé privée, via les étapes suivantes :

- Alice et Bob se mettent d'accord sur 2 nombres : **p** (un très grand nombre premier), et **g** (un autre très grand nombre, appelé générateur). **p** et **g** sont transmis en clair sur le réseau.
- Alice et Bob choisissent chacun de leur côté un très grand nombre aléatoire, qu'ils gardent secret. Soit **x** le nombre choisi par Alice, et **y** le nombre choisi par Bob.
- Alice calcule $\mathbf{P1 = g^x \bmod p}$, et transmet le résultat à Bob
- Bob calcule $\mathbf{P2 = g^y \bmod p}$, et transmet le résultat à Alice
- Alice calcule $\mathbf{K1 = P2^x}$ mod p, et Bob calcule $\mathbf{K2 = P1^y \bmod p}$

A ce stade, la valeur **K1** calculée par Alice vaut donc $\mathbf{g^{x*y} \bmod p}$. La valeur K2 calculée par Bob vaut elle $\mathbf{g^{x*y} \bmod p}$.

Les lois de l'arithmétique prouvent que les deux valeur **K1** et **K2** sont égales. Alice et Bob sont donc parvenus à se mettre d'accord sur une clé privée commune.

I.5. Conclusion

Dans ce chapitre nous avons abordé l'historique de la cryptographie et passé en revue les méthodes classiques et modernes de chiffrement de message.

Au fil du temps, les cryptogrammes sont devenus de plus en plus difficile à crypter et à décrypter. En effet, pendant l'antiquité, les informations sensibles étaient limitées en nombre.

Mais de nos jours, la protection sécurisée des données confidentielles, des communications, et des paiements a engendré une croissance exponentielle de la cryptographie.

Dans le chapitre suivant, nous abordons les notions relatives aux images, leurs formats et caractéristiques, techniques de traitement ainsi que les critères d'valuation de la qualité d'images.

Chapitre II :
Image

II.1. Introduction

L'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre. C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes. [7]

II.2. Définition

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain.

Une image numérique est une matrice de pixels repérés par leur coordonnées (x, y). S'il s'agit d'une image couleur, un pixel est codé par 3 composantes (r, g, b) (chacune comprise au sens large entre 0 et 255), représentant respectivement les "doses" de rouge, vert et bleu qui caractérisent la couleur du pixel. S'il s'agit d'une image en niveau de gris, il est codé par 1 composante comprise au sens large entre 0 et 255, représentant la luminosité du pixel. La taille totale de l'image est le nombre de pixels de largeur par le nombre de pixels de hauteur. [7] [8]

II.3. La numérisation

Numériser une image c'est lui donner une représentation électronique à partir de l'objet réel qui lui sert de support (papier, film, diapo, négatif, mais aussi objet 3D).

Cette représentation sera la plupart du temps matricielle, c'est-à-dire une matrice (un tableau) où chaque point sera représenté par une couleur.

Cette représentation électronique de l'image sera caractérisée par deux paramètres :

- La résolution : exprimée en dpi (**D**ot **P**er **I**nch (**DPI**)= **P**oint **P**ar **P**ouce (**PPP**)) c'est le nombre de points de la représentation par unité de longueur de l'objet physique à

numériser.

- La dynamique : le nombre de couleurs disponibles pour coder l'image. [9]

II.3.1. La résolution

La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur. Elle s'exprime le plus souvent en **PPP** (**P**oint **P**ar **P**ouce) ou en **DPI** (**D**ot **P**er **I**nch) parfois en point par cm. [8]

La résolution définit la netteté et la qualité d'une image. Plus la résolution est grande (c'est-à dire plus il y a de pixels dans une longueur de 1 pouce), plus votre image est précise dans les détails.

II.3.2. La dynamique

La dynamique d'une image est l'étendue de la plage de couleurs utilisable. Elle est liée à la longueur du codage de chaque couleur :

Figure 7 : Image binaire

- Si une couleur est représentée par un seul bit, on aura deux valeurs possibles, 0 ou 1, c'est-à-dire blanc ou noir. L'image sera dite binaire.
- Si une couleur est représentée sur un octet (8 bits), on aura 28 = 256 couleurs possibles. C'est le cas des images dites en "fausses couleurs" ou "à palette" (format GIF par exemple) et des images en "niveaux de gris".

Figure 8 : Image en niveau de gris

- Enfin, on parle de ‘vraies couleurs ‘ lorsqu'on utilise un octet pour stocker chacune des composantes dans l'espace de représentation des couleurs **RVB** (**R**ouge - **V**ert - **B**leu) on aura 28*28*28 = 16 millions de couleurs possibles mais chaque point sera codé sur 3 Octets.

Figure 9 : Image en couleur

II.4. Les formats d'images

Pour représenter une image, on peut la décrire à l'aide de fonctions mathématiques (représentation vectorielle) ou par l'ensemble des points qui la composent (représentation matricielle)

II.4.1. Image vectorielle

Une image vectorielle peut être agrandie ou rétrécie sans dégradation car l'image sera

recalculée précisément en fonction de la taille souhaitée. En général, le fichier correspondant est peu volumineux

Quelques formats d'images vectorielles

II.4.1.1. Le format Scalable Vector Graphics (SVG) est un format ouvert d'image vectorielle ; il est surtout utilisé en cartographie et sur les téléphones portables.

II.4.1.2. Le format Dessin de l'Open Document Format (ODF) est un format ouvert de dessin vectoriel ; il est utilisé par l'application Draw d'Open Office.

II.4.2. Image matricielle

Une image matricielle se dégrade si on l'agrandit : la pixellisation devient visible. En fonction de la taille de l'image et du nombre de couleurs utilisées, le fichier correspondant peut devenir volumineux. Pour transiter sur Internet, on utilisera des formats matriciels compressés.

Quelques formats d'images matricielles

II.4.2.1 Le format BitMAP

Le format **BitMAP** (**.bmp**) est le format d'une image numérisée représentée par un tableau de pixels de couleur. La couleur de chaque pixel est codée sur un certain nombre de bits : 1, 4, 8, 24 ou 32. Cette image peut se visualiser sur un écran d'ordinateur, s'imprimer sur une feuille de papier ou être stockée sur un support quelconque.

C'est un format intéressant d'un point de vue graphique car une telle image peut afficher beaucoup de détails. Elle présente toutefois le désavantage d'une grande taille et son éventuelle modification est délicate. En effet, toute modification d'une image en format .bmp engendre des changements point par point.

En général, le format BitMAP n'est pas compressé. Dans certains cas on peut lui appliquer une compression **RLE** (**R**un **L**ength **E**ncoding).

II.4.2.2. Le format TIFF (Tagged Image File Format)

Le format **TIF ou TIFF (.tif)** est un ancien format graphique qui permet de représenter des images BitMAP compressées sans perte de qualité.

Le format TIF utilise des balises pour décrire les caractéristiques de l'image :

- Les dimensions.
- Le nombre de couleurs utilisées.
- Le type de compression (**RLE**, **JPEG**, **LZW**).

- Les corrections appliquées.

L'usage des balises pour la description de l'image favorise les traitements à appliquer par programmation. Par contre, le grand choix d'options fait que la compatibilité des lecteurs est minimale et il arrive souvent qu'une image en format **TIFF** ne soit pas lisible car certaines options n'ont pas été intégrées.

II.4.2.3. Le format GIF (Graphics Interchange Format)

Les inventeurs du format **GIF** ont eu l'idée à l'époque où les ordinateurs étaient limités dans l'affichage des couleurs, de créer un format d'image qui permettrait de limiter la palette des couleurs d'une image à 256 couleurs.

Ainsi l'image si elle contient plus de couleurs que 256, le programme n'en retient que les 256 principales.

Notre image perd ainsi toutes les subtilités, surtout dans les dégradés, où bien souvent il faut plus de 256 couleurs.

On dit alors dans le langage courant que le format **GIF** est un format destructeur de l'image car toutes les informations qui composent l'image à l'origine ne sont pas préservées.

Points forts :

- La possibilité de transparence dans une image
- La possibilité de choisir le nombre de couleur de 2 à 256

Point faible :

- Les dégradés en limitant la palette des couleurs d'une image au maximum à 256 couleurs on allège considérablement certaines images, surtout que l'on peut descendre le choix des couleurs jusqu'à 2 couleurs

Ce format est très utilisé dans les pages Web, notamment pour les logos. En plus de la compression, il permet d'obtenir facilement de petites animations à partir de la version GIF89. [10]

II.4.2.4. Le format PNG ou Ping (Portable Network Graphics)

Le format **PNG** (.**png**) est un format de fichier graphique bitmap (raster). Il a été mis au point en 1995 afin de fournir une alternative libre au format **GIF**, format propriétaire dont les droits sont détenus par la société Unisys (propriétaire de l'algorithme de compression **LZW**), ce qui oblige chaque éditeur de logiciel manipulant ce type de format à leur verser des

royalties. Ainsi **PNG** est également un acronyme récursif pour **PNG's *Not Gif*.**

Le format **PNG** permet de stocker des images en noir et blanc (jusqu'à 16 bits par pixels de profondeur de codage), en couleurs réelles (True color, jusqu'à 48 bits par pixels de profondeur de codage) ainsi que des images indexées, faisant usage d'une palette de 256 couleurs.

De plus, il supporte la transparence par couche alpha, c'est-à-dire la possibilité de définir 256 niveaux de transparence, tandis que le format **GIF** ne permet de définir qu'une seule couleur de la palette comme transparente. Il possède également une fonction d'entrelacement permettant d'afficher l'image progressivement.

A la différence du format **GIF**, le format **PNG** ne peut pas afficher des images animées.

II.4.2.5 Format JPEG ou JPG (Joint Photographic Expert Group)

Les images **JPEG** ont l'extension "**.jpg**", "**.jpeg**", "**.jpe**" ou "**.jfif**".

Le format **JPEG**, très couramment utilisé pour le codage des images bitmap et des photos, est un format de compression très efficace. La perte de qualité d'image occasionnée par l'algorithme de compression peut être maîtrisée car le taux de compression des fichiers **.jpeg** est réglable.

Le format **JPEG** est complémentaire des formats **GIF** et **PNG** pour la publication d'images sur le Web : il sauvegarde plus d'informations couleur que le format **GIF** et permet de comprimer des photographies ou des images lourdes.

Le principal avantage de ce format est le taux de compression réglable qui permet à l'utilisateur de trouver un compromis entre le taux de compression et la qualité de l'image.

II.5. Qualité d'Image

La mesure de la qualité objective (par opposition à l'évaluation subjective de qualité par les observateurs humains) cherche à déterminer la qualité des images algorithmiquement. Le but de la recherche de l'évaluation de la qualité objective est de concevoir des algorithmes dont la prévision de la qualité est en accord avec l'évaluation subjective des observateurs humains.

Les mesures quantitatives les plus utilisées sont : l'erreur quadratique moyenne (MSE), le rapport crête signal sur bruit (Peak Signal to Noise Ratio, PSNR), le rapport signal sur bruit (Signal to Noise Ratio : SNR) etc.

II.5.1. Erreur quadratique moyenne (MSE)

L'image dégradée $\hat{I}$ est toujours comparée à l'originale I pour déterminer son rapport de ressemblance. Ce critère est le plus utilisé. Il est basé sur la mesure de l'erreur quadratique moyenne (MSE) calculée entre les pixels originaux et dégradés [11] :

$$EQM = \frac{1}{M * N} \sum_{i}^{N} \sum_{j}^{M} \left| I_{i,j} - \hat{I}_{i,j} \right|^2$$

Où (M × N) est la taille de l'image, et Ii,j et $\hat{I}i,j$ sont respectivement les amplitudes des pixels sur les images originale et dégradée. Il est vraisemblable que l'œil tienne beaucoup plus compte des erreurs à grandes amplitudes, ce qui favorise la mesure quadratique.

II.5.2. Rapport crête signal sur bruit (PSNR)

Au lieu de mesurer la distorsion, cette valeur (Peak Signal to Noise Ratio, PSNR) [11] mesure la fidélité, puisqu'elle est proportionnelle à la qualité. Tout de même, elle est une fonction de MSE ; sa définition et son utilisation proviennent du domaine du traitement de signal :

$$PSNR_{dB} = 10log_{10}\left(\frac{L^2}{EQM}\right)$$

Pour une image à niveau de gris, L = *Imax* désigne la luminance maximale possible. Une valeur de PSNR infini correspond à une image non dégradée. Et cette valeur décroît en fonction de la dégradation. Le PSNR relie donc le MSE à l'énergie maximale de l'image.

II.5.3 Rapport signal sur bruit (SNR)

On utilise parfois une autre variante du rapport signal sur bruit (Signal to

Noise Ratio : SNR), [11] qui relient le MSE à l'énergie moyenne de l'image :

$$SNR = 10\log_{10}\left(\frac{\frac{1}{N}\sum I^2}{MSE}\right)$$

II.6. Conclusion

Dans ce chapitre nous avons présenté les différents formats d'image et leurs

caractéristiques ainsi que leurs points fort et points faibles, et aussi nous mis en exergue la place importante qu'occupe les images dans les communications sur les réseaux (internet notamment).

Le chiffrement des images a beaucoup d'applications dans divers domaines, tel que la communication mobile, Internet, l'imagerie médicale, la télémédecine et les communications militaires. Il existe un nombre très important de techniques de chiffrement d'images. Dans le chapitre suivant, nous étudierons en détail une de ces possibles techniques, le protocole de chiffrement RSA que nous allons par la suite utiliser pour chiffrer nos images.

Chapitre III : Cryptosystème RSA

III.1 Préambule

Dans ce chapitre, nous détaillons l'algorithme RSA qui sera utilisé dans notre cryptosystème. **RSA**, acronyme de Rivest-Shamir-Adleman (initiales de ses trois inventeurs) est très utilisé dans le domaine de l'internet notamment pour des échanges d'informations confidentielles. Le plus connu des algorithmes de cryptographie asymétrique (appelé habituellement le chiffrement RSA) semblent avoir été développés indépendamment par le *Government Communications Headquarters* (GCHQ), le service de renseignements électroniques du gouvernement du Royaume-Uni dans les années 1976. [12]

Pour commencer, le chiffrement asymétrique est appelé chiffrement à clé publique c'est à dire que l'algorithme utilise 2 clefs :

- Une clef pour chiffrer : Clé publique
- Une clef pour déchiffrer : Clé secrète.

III.2 L'algorithme RSA

Le premier système à clé publique solide utilisé jusqu'à nos jours et le plus utilisé actuellement est le système RSA. Le RSA est fondé sur deux principes mathématiques fondamentaux : la difficulté de factoriser des grands nombres et l'arithmétique des congruences.

III. 3 Principe de fonctionnement de l'algorithme RSA [13]

Pour qu'Alice puisse échanger sa clé avec Bob, elle doit d'abord la calculer en mettant œuvre des notions mathématiques remarquables par leur simplicité. L'algorithme RSA est ainsi défini par 03 phases :

- Génération des clés (effectué par la destinataire Alice)
- Chiffrement (effectué par l'expéditeur Bob)
- Déchiffrement (effectué par la destinataire Alice)

III.3.1 Génération des clés :

Cette phase peut se résumer en 03 étapes :

1ère étape : Alice choisit au hasard deux grands nombres entiers, naturels et premiers. Les nombres p et q ont environ 100 chiffres chacun ou plus pour rendre une factorisation difficile. Dans notre exemple simplifié elle choisit :

p =31 et q = 53 Et calcule leur produit : n = p*q =1643

2er étape : Alice détermine la fonction d'Euler associée à 'n' déjà calculé en utilisant la formule (3.1):

$$\phi(n) = (p-1) \times (q-1)$$
$$\phi(n) = 30 \times 52 = 1560 \qquad (3.1)$$

Une fois la fonction d'Euler déterminée, Alice choisie une clé publique « e », cette clé doit être un nombre premier compris entre 1 et ϕ(n) et premier avec ϕ (n) (3.2).

$$1 < e < \phi(n)$$
$$\text{PGCD}(e, \phi(n)) = 1 \qquad (3.2)$$

Alice prend : e = 11, D'où le couple (e, n) constitue la clé publique. Avec n étant le modulo et e l'exposant du chiffrement. La clé publique est donc (11, 1643).

3er étape : Cette dernière étape consiste à trouver la clé privée « d » qui correspond à la clé publique choisie précédemment avec d compris entre 1 et phi(n). Pour se faire, il faut résoudre l'équation suivante (3.3).

$$e * d \text{ modulo } \phi(n) = 1 \quad \text{c.à.d. } e*d \text{ congru à 1 modulo } \phi(n)$$
$$\text{Donc : } e*d = k*\phi(n) + 1 \qquad (3.3)$$

Pour notre exemple la résolution de cette équation à l'aide de l'algorithme de Bezout-Euclide on aura la solution k = 6 et d = 851.

Le couple (d, n) constitue la clé privée. La clé privée est donc (851, 1643). Avec n qui est le module et d est l'exposant du déchiffrement.

Nous notons qu'après la publication de la clé publique (e, n), toute personne peut chiffrer un message, seul Alice qui dispose de la clé secrète peut déchiffrer.

III.3.2 Chiffrement

Bob veut donc transmettre le message M « ANEMONE » à Alice. Il cherche dans l'annuaire la clé de chiffrement qu'Alice a déjà publiée. Il sait maintenant qu'il doit utiliser le système RSA avec les deux entiers n et e (Pour notre exemple on prend n = 1643 et e = 11).

Chiffrement :

- Il transforme en nombres son message en remplaçant chaque lettre par son rang dans l'alphabet latine : a = 01, b = 02 …… z = 26.

Le message M = A N E M O NE est transformé en : M = 01 14 05 13 15 14 05

- Il découpe son message numérisé en blocs de longueurs égales et dont la taille est égale ou inférieure à celle de n ce qui empêche la simple substitution.

Dans notre exemple la taille de n est 3, ce qui donne des tranches m_i de 03 chiffres

chacune, le message devient :

M = 001 140 513 151 405

$M = m_1\ m_2\ m_3\ m_4\ m_5$

Chaque bloc m_i est chiffré par l'équation : $\mathbf{C_i = M_i^e}$ **modulo n**

Ce qui donne :

$C1 = m_1^{11} \bmod 1643 = 001$

$C2 = m_2^{11} \bmod 1643 = 109$

.. . . .

.. . . .

$C_5 = m_5^{11} \bmod 1643 = 374$

Alors le message chiffré C sera :

C = C1 C2 C3 C4 C5 = 0001 0109 0890 1453 0374

Enfin, Bob a son message chiffré, il peut donc l'envoyer à Alice.

Déchiffrement :

Alice reçoit le message de Bob, à partir de p et q, qu'elle a gardé secret, et la clé d de déchiffrement (clef privée).

Chacun des blocs Ci du message chiffré sera déchiffré par l'équation : $\mathbf{Mi = C_i^d}$ **modulo n**

Ce qui donne :

$m1 = 1^{851}$ modulo 1643 = 1

$m2 = 109^{851}$ modulo 1643 = 140

$m3 = 890^{851}$ modulo 1643 = 513

$m4 = 1453^{851}$ modulo 1643 = 151

$m5 = 374^{851}$ modulo 1643 = 405

Alors le message clair est : M = 001 140 513 151 405

Lors du déchiffrement, sachant qu'il faut obtenir des blocs de 2 éléments (grâce au codage particulier de l'exemple). Finalement, Alice prend sa table de correspondance alphabétique pour restituer le message M, elle aura :

01 14 05 13 15 14 05 = ANEMONE

III.4. Résumé :

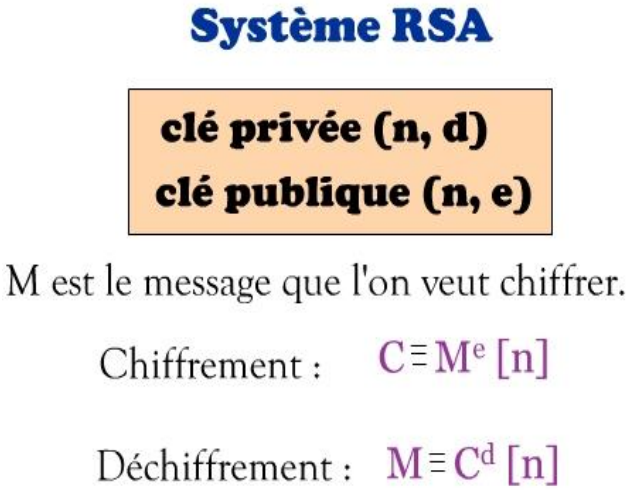


Figure 10 : Chiffrement et Déchiffrement RSA

Algorithme de la RSA (Chiffrement asymétrique)

- Génération de 2 nombres premiers p et q
- Calcul de n = p ×q
- Déterminer e tel que 1< e <ϕ(n) et pgcd (e, ϕ(n)) =1
- Calculer d tel que e × d ≡ 1 mod ϕ(n)
- Clé publique : (e, n)
- Clé privée : (d, n)
- p et q doivent rester secrets, voire supprimés.
- $C = M^e \bmod n$ et $M = C^d \bmod n$

Remarque : Pour un chiffrement très robuste, en réalité les valeurs de P et Q et doivent être très grandes, afin de prévenir la cryptanalyse par force brute.

Exemple de deux nombres premiers P et Q :

P = 16347336458092538484431338838650908598417836700330
92312181110852389333100104508151212118167511579

Q = 19008712816648221131268515739354139754718967899685154936666385390880271038021044989571912614655 71

III.5. Conclusion

Nous avons présenté dans ce chapitre, la structure détaillée de l'algorithme RSA. Nous avons défini les étapes à suivre pour générer la paire de clefs privée-publique de l'algorithme RSA. L'utilisation de cet algorithme est avantageuse, car il occupe peu de mémoire, de moindre complexité et facile à implémenter.

Le chiffrement RSA est l'algorithme de chiffrement asymétrique le plus fiable puisqu'il est impossible de trouver la clef de déchiffrement à partir de la clef de chiffrement avec deux nombres premiers très grands. Le chiffrement RSA est un des plus utilisés de nos jours, il est présent dans nos cartes bancaires, nos transactions, nos messageries et nos logiciels (tel le chiffrement de winrar).

Dans le prochain chapitre nous allons utiliser la méthode RSA pour chiffrer nos images numériques. Nous utilisons des images noir et blanc, chaque pixel d'une image étant codé sur 8 bits (octets) donc ayant une valeur comprise entre 0 et 255.

Chapitre IV :
Résultats et Discussions

IV.1. Préambule :

Dans ce mémoire, notre travail consiste en la simulation, par le cryptosystème RSA, le chiffrement/déchiffrement d'images standards présents sur Matlab et utilisés par les chercheurs qui travaillent sur le traitement d'images.

Dans la discussion des résultats, nous comparerons les résultats des différentes méthodes abordées pour déterminer la meilleure approche possible en se basant sur les deux critères d'évaluation de qualité d'image (l'EQM et PSNR) étudiée dans le chapitre II.

Par ailleurs nous procéderons à une étude analytique entre les résultats issus de notre simulation et les résultats d'autres d'études publiés par les chercheurs pour évaluer la qualité de la méthode de chiffrement proposée.

a) Les paramètres RSA sont :

- **Pour N1 = 253**

Les nombres premiers p et q tels que leur produit est égal à N1 est : p1 = 11 ; q1 = 23

L'indicateur d'Euler Phi(N1) = (p1-1)*(q1-1) = 220.

La clé publique de chiffrement est (e1, N1) = (7, 253)

La clé privée de déchiffrement est (d1, N1) = (63, 253)

- **Pour N2 = 259**

Les nombres premiers p et q tels que leur produit est égal à N2 est : p2 = 07 ; q2 = 37

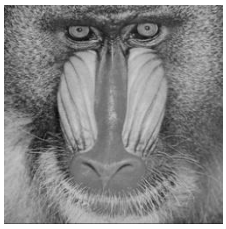
L'indicateur d'Euler Phi(N2) = (p2-1)*(q2-1) = 216.

La clé publique de chiffrement est (e2, N2) = (7, 259)

La clé privée de déchiffrement est (d2, N2) = (31, 259)

- **Pour N3 = min (N1 ; N2)**
- **Pour N4 = (N1+N2)/2 = 256**

b) Les images utilisées pour la simulation :


Baboon.png (256, 256)


Pepper.bmp (256, 256)


cameraman.tif (256, 256)


voiture.bmp (256, 256)


lena.jpg (256, 256)

c) Caractéristiques du micro utilisé et Version Matlab :

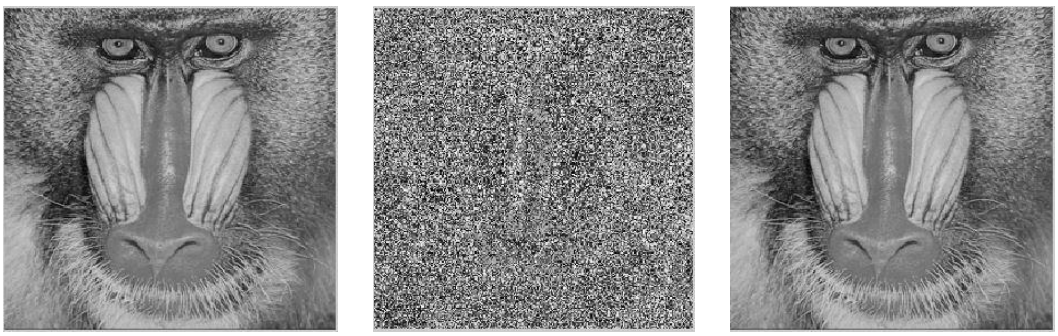
La programmation a été faite en langage Matlab (2014a). Les spécifications de PC : Intel (R) Pentium (R) CPU B960 @ 2.20 GHz, 4 GB RAM ; Système d'exploitation 64 bits, processeur x64.

IV.2. Les résultats de la simulation :

IV.2.1 Les images

1. Baboon.png (256, 256)

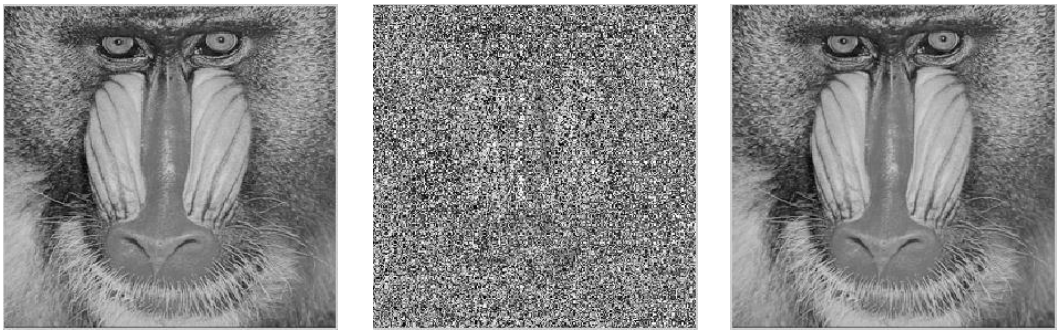
- **N1 = 253**



(a) (b) (c)

Figure 11.1 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

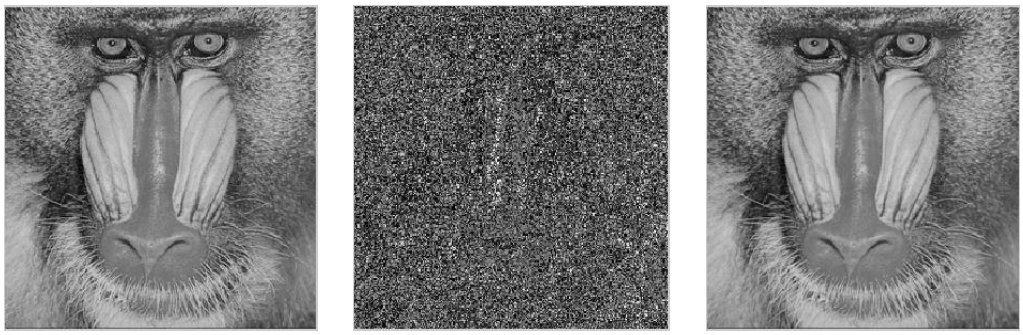
- **N2 = 259**



(a) (b) (c)

Figure 11.2 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

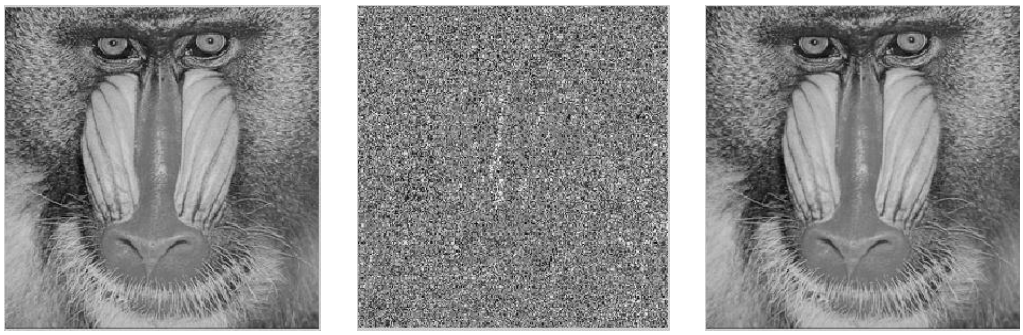
- **N3 = min (N1, N2)**



(a) (b) (c)

Figure 11.3 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N4 = (N1+N2)/2 = 256**



(a) (b) (c)

Figure 11.4 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

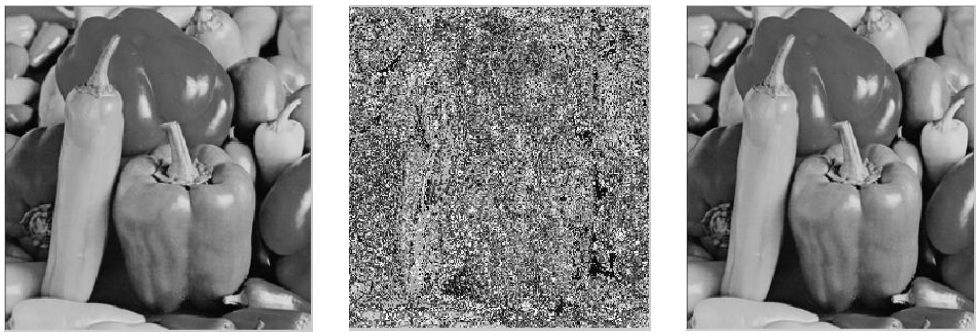
2. Pepeer.bmp (256, 256)

- **N1 = 253**

(a) (b) (c)

Figure 12.1 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

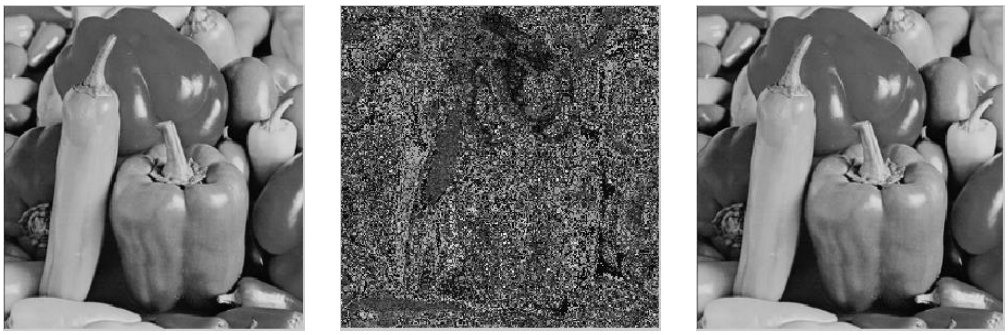
- **N2 = 259**



(a) (b) (c)

Figure 12.2 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N3 = min (N1, N2)**



(a) (b) (c)

Figure 12.3 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

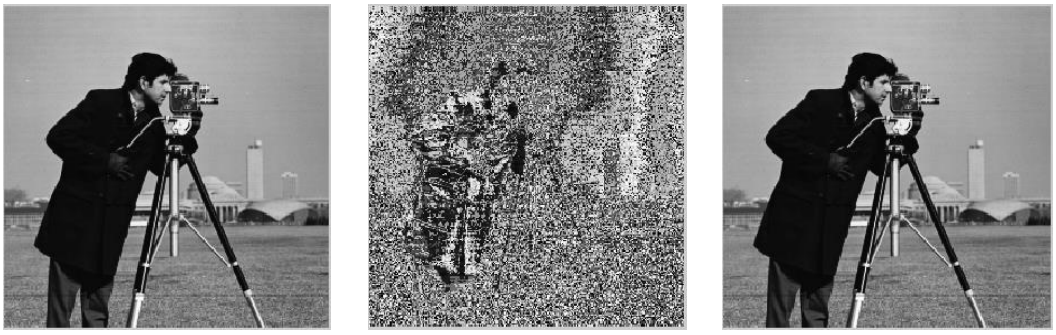
- **N4 = (N1+N2)/2 = 256**

(a) (b) (c)

Figure 12.4 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

3. cameraman.tif (256, 256)

- **N1 = 253**



(a) (b) (c)

Figure 13.1 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N2 = 259**

(a) (b) (c)

Figure 13.2 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N3 = min (N1, N2)**

(a) (b) (c)

Figure 13.3 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

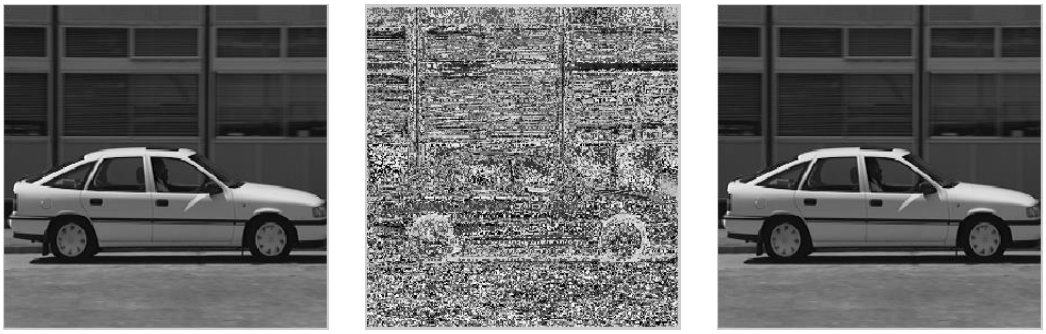
- **N4 = (N1+N2)/2 = 256**

(a) (b) (c)

Figure 13.4 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

4. voiture.bmp (256, 256)

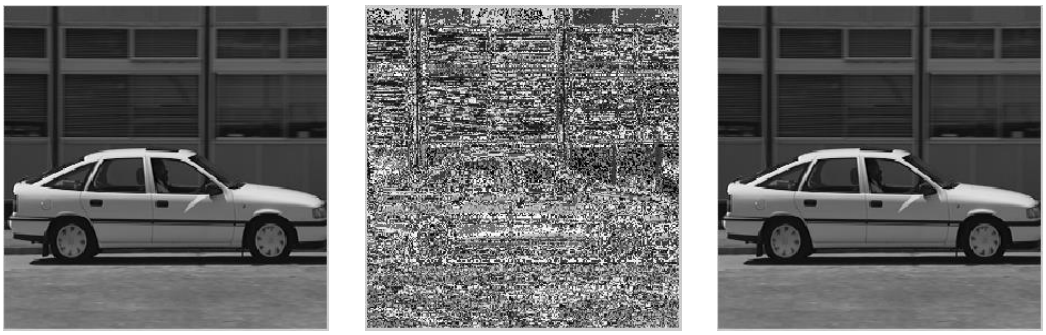
- **N1 = 253**



(a) (b) (c)

Figure 14.1 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

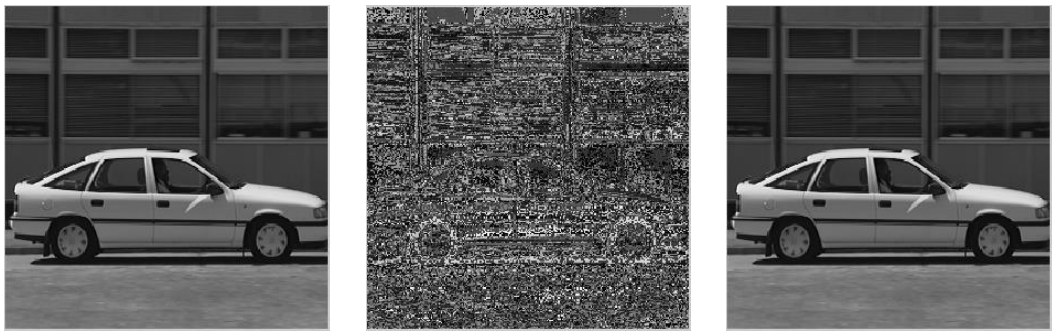
- **N2 = 259**



(a) (b) (c)

Figure 14.2 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

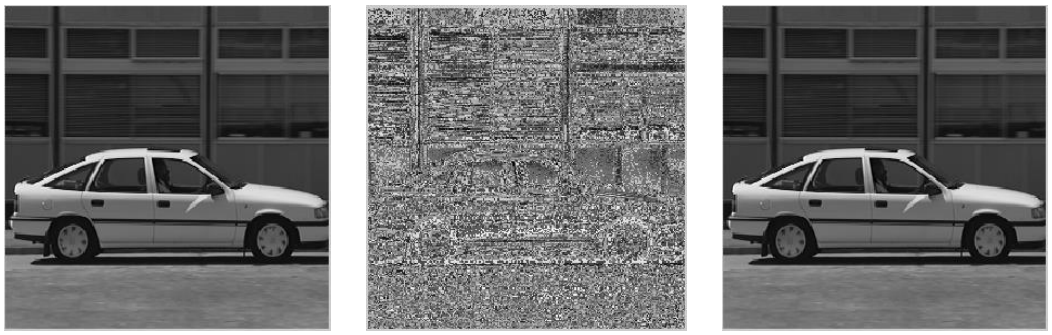
- **N3 = min (N1, N2)**



(a) (b) (c)

Figure 14.3 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N4 = (N1+N2)/2 = 256**

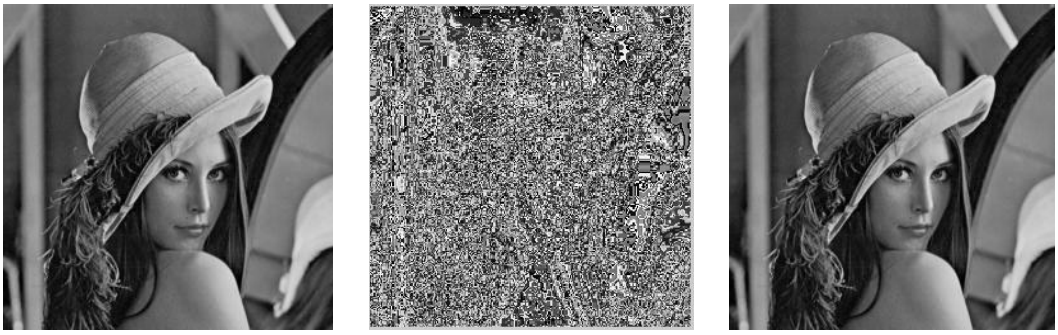


(a) (b) (c)

Figure 14.4 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

5. lena.jpg (256, 256)

- **N1 = 253**



(a) (b) (c)

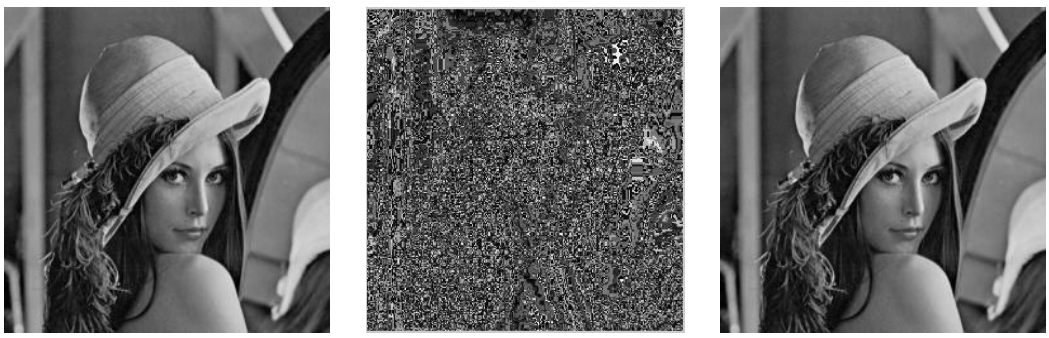
Figure 15.1 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N2 = 259**

(a) (b) (c)

Figure 15.2 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

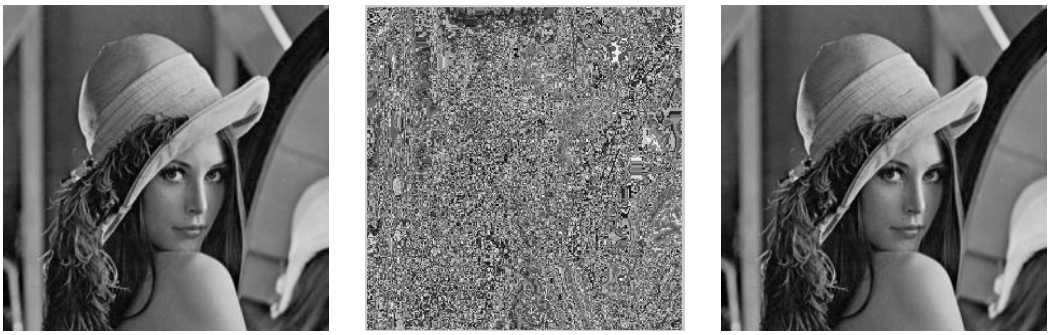
- **N3 = min (N1, N2)**



(a) (b) (c)

Figure 15.3 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

- **N4 = (N1+N2)/2 = 256**



(a) (b) (c)

Figure 15.5 : a) Image claire ; b) Image chiffrée ; c) Image déchiffrée

IV.2.2. Discussion des résultats :

A. EQM :

- Les résultats de l'EQM entre l'image claire et l'image chiffrée :

	N			
	N1	N2	N3 (min)	N4 (moy)
Baboom.png (256, 256)	120.3584	96.3433	**165.7888**	118.1559
Pepper.bmp (256, 256)	110.0321	90.7459	**154.5069**	106.5141
cameraman.tif (256, 256)	118.7020	90.4547	**154.6678**	120.8627
voiture.bmp (256, 256)	64.2795	55.7036	**102.7376**	47.0328
lena.jpg (256, 256)	87.9570	71.6918	**126.8859**	76.3614

Tableau 3 : l'EQM entre l'image claire et l'image chiffrée

- Les résultats de l'EQM entre l'image claire et l'image déchiffrée :

	N			
	N1	N2	N3 (min)	N4 (moy)
Baboom.png (256, 256)	0	0	0	0
Pepper.bmp (256, 256)	0	0	0	0
cameraman.tif (256, 256)	0.0039	0	0.0039	0.0039
voiture.bmp (256, 256)	0.0078	0	0.0078	0.0078
lena.jpg (256, 256)	0	0	0	0

Tableau 4 : l'EQM entre l'image claire et l'image déchiffrée

La méthode N3 (min) qui consiste à prendre la valeur du niveau de gris entre N1 et N2 qui se rapproche le plus de la valeur 256. Cette image formée présente un EQM très élevé par rapport au clair ; ce qui correspond à un bon chiffrement. Quoique aussi les méthodes N1, N2 et N4(moy) donnent aussi des résultats très acceptables.

B. PSNR :

- Les résultats du PSNR entre l'image claire et l'image chiffrée :

	N			
	N1	N2	N3 (min)	N4 (moy)
Baboom.png (256, 256)	27.3260	28.2926	**25.9353**	27.4063
Pepper.bmp (256, 256)	27.7156	28.5525	**26.2413**	27.8567
cameraman.tif (256, 256)	27.3862	28.5665	**26.2368**	27.3079
voiture.bmp (256, 256)	30.0501	30.6720	**28.0135**	**31.4068**
lena.jpg (256, 256)	28.6881	29.5761	**27.0967**	29.3021

Tableau 5 : PSNR entre l'image claire et l'image chiffrée

Un PSNR inférieur à 30 correspond à un bon chiffrement. On note qu'à l'exception de N4 (moy), pour le cas de l'image voiture et dont le chiffrement est peu bon, la méthode N3 (min) donne de très bons résultats.

- Les résultats du PSNR entre l'image claire et l'image déchiffrée :

	N			
	N1	N2	N3 (min)	N4 (moy)
Baboom.png (256, 256)	>>	>>	>>	>>
Pepper.bmp (256, 256)	>>	>>	>>	>>
cameraman.tif (256, 256)	72.2302	>>	72.2302	72.2302
voiture.bmp (256, 256)	69.2199	>>	69.2199	69.2199
lena.jpg (256, 256)	>>	>>	>>	>>

Tableau 6 : PSNR entre l'image claire et l'image déchiffrée

Un PSNR lors du déchiffrement qui dépasse la cinquantaine est acceptable. On constate que les 4 méthodes étudiées donnent le même résultat pratiquement. Nous notons que ces résultats correspondent à un canal de transmission idéal (Pas de bruit). Cependant, il y'a un compromis entre le chiffrement et déchiffrement et à cet effet, la méthode N3 reste la meilleure.

C. Temps de simulation (seconde)

- Le temps de chiffrement :

	N			
	N1	N2	N3 (min)	N4 (moy)
Baboom.png (256, 256)	0.4532	0.4571	0.4632	**0.4836**
Pepper.bmp (256, 256)	0.3957	0.3987	0.4010	**0.4019**
cameraman.tif (256, 256)	0.4095	0.4099	0.4107	**0.4190**
voiture.bmp (256, 256)	0.4267	0.4267	0.4277	**0.4283**
lena.jpg (256, 256)	0.4312	0.4319	0.4402	**0.4485**

Tableau 7 : Temps de chiffrement

Le temps de chiffrement dépend en premier lieu de la dimension de l'image. Pour nos images utilisées, les dimensions sont identiques, nous remarquons que le temps de chiffrement est fonction du nombre d'opérations effectuées. Pour les méthodes N1 et N2, les temps sont identiques pour les 4 images car le chiffrement correspond au calcul d'une puissance modulo (x^e modulo N). Pour la méthode N3, en plus du calcul modulo, il faut rajouter un temps de comparaison pour prendre la plus petite valeur. Pour la méthode N4, la méthode avec un temps de simulation le plus élevé, en plus du calcul modulo, il y'a une somme à effectuer puis une division par deux.

- Le temps de déchiffrement :

	N			
	N1	N2	N3 (min)	N4 (moy)
Baboom.png (256, 256)	0.2056	0.2066	0.2101	**0.2150**
Pepper.bmp (256, 256)	0.1716	0.1785	0.2738	**0.2840**
cameraman.tif (256, 256)	0.2591	0.2593	0.2617	**0.2677**
voiture.bmp (256, 256)	0.1711	0.1711	0.1746	**0.1750**
lena.jpg (256, 256)	0.2624	0.2827	0.2838	**0.2846**

Tableau 8 : Temps de déchiffrement

Quant au temps de déchiffrement, c'est Le même raisonnement fait précédemment en ce qui concerne le chiffrement mais avec maintenant l'opération modulo qui est (c^d modulo N). La méthode N4 donne les temps les plus élevés par rapport aux trois autres méthodes.

D. Comparaison avec d'autres papiers :

	WT [14]	WIFI AVC [15]	Méthode RSA proposée N3 = min (N1, N2)
Images	**Cryptage**	**Cryptage**	**Cryptage**
Lena	29.02		**27.0967**
Barbara			
Cameraman	28.37	28.9	**26.2368**
Baboom			**25.9353**
Pepper			**26.2413**

Tableau 9 : Comparaison avec d'autres papiers

L'étude comparative avec ces résultats donnés par différents chercheurs montre que notre approche s'avère meilleure. En effet, les valeurs de PSNR issues de notre simulation sont légèrement inférieures à celles de [1] et [4], ce qui s'explique par le fait que le cryptosystème RSA restitue exactement les valeurs chiffées pour un canal sans bruit. En fait le chiffré n'étant que le modulo d'un nombre qui, élevé à la puissance de l'inverse de la clé de chiffrement pour retrouver le même nombre.

Conclusion Générale

Conclusion générale

Le transfert sécurisé d'information est nécessaire et énormément utilisé dans le monde numérique. Les réseaux numériques ont fortement évolué ces dernières années et sont devenus inévitables pour la communication moderne. Cette évolution est importante car la cryptographie joue un grand rôle dans la sécurité de l'information.

Le cryptage permet de rendre l'information secrète. Les gouvernements, armées et industries utilisent ces technologies afin de protéger certaines informations. Les individus utilisent à leur tour ces technologies, dans le cadre de leur vie privée. Le développement des outils informatiques permet aujourd'hui une sécurité accrue dans les échanges d'informations mais permet aussi un décryptage toujours plus facile par des personnes indélicates.

RSA est l'algorithme de chiffrement asymétrique le plus utilisé de nos jours. Le RSA est utilisé de nos jours comme un deuxième chiffrement sécurisant le WIFI qui est déjà protégé par le RC4.

Actuellement, aucun algorithme de chiffrement connue d'images par RSA est mis à la disposition du public. Jusqu'à nos jours cet algorithme est gardé secret.

Pour solutionner ce problème, et dans le contexte de notre projet de fin d'études, nous avons proposé quelques méthodes de chiffrement /déchiffrement d'images par la méthode RSA.

Pour résoudre le problème de factorisation du nombre N = 255 en deux nombres premiers, nous avons proposé quatre valeurs se rapprochant de N et qui sont décomposables en produit de deux nombres premiers. Une première valeur N1, inferieur à N et pouvant se décomposer en produit de deux nombres premiers, une deuxième valeur N2 supérieur à N et répondant au principe de RSA. Les valeurs N3 et N4 étant la combinaison des deux valeurs de N considérées (N1 et N2) ; avec N3 les plus petites valeurs issues du chiffrement des octets avec N1 et N2, et N4 la moyenne des dites valeurs.

Par la suite nous avons comparé les résultats issus des quatre contributions pour déterminer la meilleure, en utilisant comme critères l'Ecart Quadratique Moyen (EQM) et le Rapport Signal/Bruit (PSNR). Aussi, nous avons analyser nos résultats au vu d'autres résultats de chiffrement d'image existant dans la littérature scientifique. Nos résultats sont très performants par rapport aux résultats publiés récemment.

De par les résultats de notre simulation, nous concluons que la méthode N3 (min) qui consiste à prendre la valeur du niveau de gris entre N1 et N2 qui se rapproche le plus de la valeur 256 est la meilleure approche d'image par RSA. Quoique aussi les méthodes N1, N2 et N4(moy) donnent aussi des résultats très acceptables.

En perspectives à ce mémoire, nous proposons d'étendre le chiffrement afin de l'appliquer aux images chromatiques. Et aussi, envisager d'autres approches tel que le chiffrement RSA par colonne puis en faisant un chiffrement hybride combinant les chiffrés des colonnes et des lignes.

Références :

[1] http://dspace.univ-tlemcen.dz/bitstream/112/6836/1/Etude-comparative-entre-la- cryptographie.pdf

[2] http://deptinfo.unice.fr/twiki/pub/Linfo/PlanningDesSoutenances20032004/blanc- degeorges.pdf

[3] http://www.cryptage.org/feistel.html

[4] https://belhob.wordpress.com/2007/12/14/the-des-algorithm/

[5] http://www.supinfo.com/articles/single/1290-cryptanalyse-chiffrement-flot.

[6] 17485/ijst/2015/v8i12/62433, June 2015.

[7] http://www.univorleans.fr/mapmo/membres/louchet/teaching/timo/ben_hamadi/rapport_ benhamadi.pdf

[8] https://mtlnumerique.uqam.ca/upload/files/presentation1_LeonRobinchaud_theorie.pdf

[9] http://www.unige.ch/cyberdocuments/didacticiel/unite2/module4.html

[10] www.awt.be/web/img/index.aspx?page=img,fr,tel,040,010

[11] B.Girod, “What’s wrong with mean-squared error,”in Digital Images and Human Vision (A.B.Watson, ed.), pp. 207-220, the MIT press, 1993.

[12] https://fr.wikipedia.org/wiki/Histoire_de_la_cryptologie

[13] http://www.nymphomath.ch/crypto/rsa/index.html

[14] Wavelet Transforms, Indian Journal of Science and Technology, Vol. 8(12), DOI: 10.17485/ijst/2015/v8i12/62433, June 2015.

[15] Visual Cryptography, called Wi-Fi AVC, VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608