وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار- عنابة

**Année : 2018**

**Faculté: Sciences de l'ingéniorat**
**Département: Electronique**

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

**Intitulé**

# Application du protocole d'EL GAMAL: Chiffrement /déchiffrement d'images

**Domaine : Sciences et Technologie.**

**Filière : Electronique.**

**Spécialité : Réseaux et Télécommunication.**

**Par: M$^{elle}$ BOUMACHTA Imen.**

**DEVANT Le JURY**

**Président : FEZARI M.**

**Directeur de mémoire : KADDECHE M.**

**Examinateur : SMIRA H.**

**Examinateur : TAIBI M.**

# *DEDICACES*

*Je dédie ce modeste travail à :*

*Mon cher père.*

*Ma chère mère.*

*Mes frères : Saber et Abdallah*

*Mes sœurs : Meriem et Khouloud*

*Toutes mes fidèles amies.*

*A tous ceux qui me connaissent de près ou de loin.*

*IMEN*

# Remerciements

Je remercie dieu qui nous a donné naissance, santé et plénitude de nos sens.

Je tiens à remercier mes chers parents, de m'avoir

Soutenu et permis d'étudier pendant toutes ces années dans des conditions optimales.

Je remercie très vivement mon encadreur :

*MR. KADDECHE Mohamed.*

Pour son encadrement, ces précieux conseils et orientations.

Mes remerciements s'adressent également à tous les membres du Jury, a tous mes Enseignants qui m'ont accompagné durant toute ma formation.

Je remercie mes amis et toute personne ayant collaboré de près ou de loin à l'élaboration de ce modeste travail.

*Merci*

## Résumé

Le but de la cryptographie moderne est de traiter plus généralement des problèmes de la sécurité des communications et de fournir un certain nombre de services de sécurité tels que la confidentialité, l'intégrité et l'authenticité des données transmises. La cryptographie moderne se compose de deux grandes parties : La cryptographie symétrique et la cryptographie asymétrique.

Dans ce papier, nous présentons une des techniques de la cryptographié moderne nommé chiffrement d'ELGAMAL. La simulation de ce protocole D'ELGAMEL est appliquée sur de différentes images issues de bases de données. Nous procédons à des choix aléatoires des paramètres utilisés dans cette méthode à savoir : le nombre premier '**p**', '**a**' qui est une racine primitive de '**p**', différentes clé secrètes '**sk**' et un ensemble de nombres entiers aléatoires '**k**'. Nous essayons par les variations de ces différents paramètres d'aboutir à une qualité de chiffrement/déchiffrement appréciable.

**Les mots clés** : Image, Cryptage, ELGAMAL, EQM, PSNR.

# Résumé


## 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 ELGAMAL encryption. The simulation of this ELGAMEL protocol is applied to different images from databases. We make random choices of the parameters used in this method namely: the prime number '**p**', '**a**' which is a primitive root of **'p**', different secret keys' **sk** 'and a set of random integers '**k**'. We try by variations of these different parameters to achieve a quality of encryption / decryption appreciable.



**Keywords:** Picture, Encryption, ELGAMAL, EQM, PNSR

# Résumé


**ملخص**

الغرض من التشفير الحديث هو معالجة مشاكل أمن الاتصالات بشكل عام وتوفير عدد من الخدمات الأمنية مثل سرية وسلامة ومصداقية البيانات المرسلة .يتكون التشفير الحديث من جزأين رئيسيين: تشفير متماثل والتشفير غير المتماثل. في هذه الورقة ، نقدم أحد تقنيات التشفير الحديثة المعروفة بتشفير ELGAMAL .يتم تطبيق محاكاة بروتوكول ELGAMEL هذا على صور مختلفة من قواعد البيانات .نقوم باختيارات عشوائية للمعلمات المستخدمة في هذه الطريقة وهي: الرقم الأول**'p'** ، **'a'** وهو الجذر الأساسي لـ **'p'** ، ومفاتيح سرية مختلفة **'sk'** ومجموعة من الأعداد الصحيحة العشوائية **'k'** . نحن نحاول من خلال هذه المعايير المختلفة لتحقيق جودة التشفير / فك التشفير مقبول.



**كلمات البحث:** الصورة والتشفير،ELGAMAL، EQM , PSNR


.

# Liste des abréviations

## Liste des abréviations

- **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.
- **RSA:** Du nom de ses inventeurs ron Rivest, adi Shamir et len Aldeman.
- **PGCD:** Plus Grand Commun Diviseur.

## Liste des figures

### Chapitre 1



### Chapitre 2



### Chapitre 3



### Chapitre 4

# Liste des figures

## Liste des tableaux

### Chapitre 2



### Chapitre 3



### Chapitre 4

## Table des matières

# Table des matières



## Chapitre 3 : Chiffrement El Gamal



## Chapitre 4 : Résultats et discussion



## Conclusion générale

# Introduction générale

## Introduction générale

De tous temps, les services secrets ont utilisé toutes sortes de codages et de moyens cryptographiques pour communiquer entre agents et gouvernements, de telle sorte que les "ennemis" ne puissent pas comprendre les informations échangées. La cryptologie a alors évolué dans ces milieux fermés qu'étaient les gouvernements, les services secrets et les armées. Ainsi, très peu de gens, voire personne n'utilisait la cryptographie à des fins personnelles. C'est pourquoi, pendant tant d'années, la cryptologie est restée une science discrète.

La cryptologie est de plus en plus utilisée sur le réseau mondial Internet. Avec l'apparition du commerce en ligne, c'est-à-dire la possibilité de commander des produits directement sur Internet, la cryptographie est devenue nécessaire. En effet, si les différents ordinateurs branchés sur Internet sont sécurisés par des mots de passe, c'est-à-dire à priori inaccessibles par un ennemi, les transactions de données entre deux ordinateurs distants via Internet sont, quant à elles, facilement interceptées. C'est pourquoi lorsque l'on commande un produit sur Internet en payant avec notre carte bancaire, il est beaucoup plus sûr d'envoyer notre numéro de carte bancaire une fois crypté, celui-ci ne pourra à priori, être décrypté que par la société à laquelle on a commandé ce produit.

C'est pour ces mêmes raisons d'insécurité sur Internet, et par un besoin humain d'intimité que la cryptographie à des fins purement personnelles s'est développée sur le réseau : pour la messagerie électronique. En effet lorsque l'on envoie un message électronique par Internet, on préfère qu'il reste discret vis à vis de la communauté Internet, voire qu'il ne soit compréhensible que par le destinataire du message. En d'autres termes, la cryptographie assure la confidentialité de transfert de documents.

Le travail présenté dans ce mémoire a pour objectif d’étudier le chiffrement /déchiffrement d’images par le protocole d’EL GAMAL.

Une étude comparative entre l’image clair et limage chiffrée avec le calcul de deux paramètres : l’EQM et le PSNR.

Notre mémoire sera fractionné en quatre chapitres :

Dans Le premier chapitre on a donné une notion brève sur quelques paramètres et définition d'une image et quelques méthodes de compression.

Le deuxième chapitre est dédié à l'historique de la cryptographie : cryptographie classique et la cryptographie moderne avec une brève explication de la méthode.

Dans Le troisième chapitre, nous allons détailler la méthode utilisée qui est le chiffrement EL GAMAL.

Enfin dans Le quatrième chapitre, nous allons présenter, discuter et analyser les résultats de simulation de données issues d'mages et terminer ce manuscrit par une conclusion générale.

# Chapitre 1
# L’image

## 1.1 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. [1]

## 1.2 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. [1] [2]

## 1.3 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. [3]

### 1.3.1 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. [2]

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

### 1.3.2 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 :

- 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.

**Figure 1.1 :** Image 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 1.2 :** 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 1.3 :** Image en couleur

## 1.4 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)

### 1.4.1 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

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

**1.4.1.2 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

### 1.4.2 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

#### 1.4.2.1 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).

### 1.4.2.2 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 appliqués.

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.

### 1.4.2.3 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. [4]

#### 1.4.2.4 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.

### 1.4.2.5 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.

# Chapitre 2
# Cryptographie

## 2.1 Introduction

L'origine de la cryptologie mot réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments : « cryptos », qui signifie caché et « logos » qui signifie mot. La cryptologie est aussi vieille que l'écriture elle-même, et a été utilisé depuis des milliers d'années pour assurer les communications militaires et diplomatiques. Par exemple, le célèbre empereur romain Jule César utilisait un algorithme de chiffrement pour protéger les messages à ses troupes. Dans le domaine de l'un de cryptologie peut voir deux visions : la cryptographie et la cryptanalyse. Le cryptographe cherche des méthodes pour assurer la sûreté et la sécurité des conversations alors que le Crypto analyse tente de défaire le travail ancien en brisant ses systèmes. [5]

## 2.2 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.

## 2.3 Quelque 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).

### 2.3.1 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. [6]

| 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 2.1 :** 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'il 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.

### 2.3.2 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. [6]

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 2.2** : 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 Vigenere », 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.

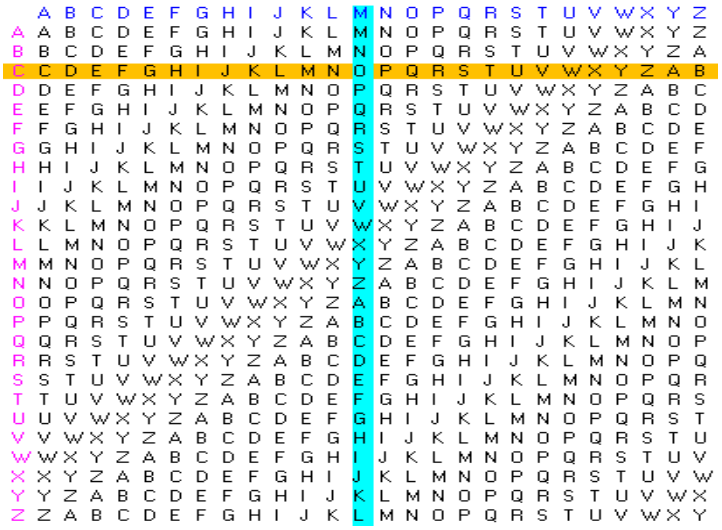


**Figure 2.1** : Cryptage par carre de Vigenère

### 2.3.3 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**} 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**.**

## 2.4 Cryptosystèmes actuels {modernes}

### 2.4.1 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 cryptage et la clef de décryptage sont une seule et même clef.


**Figure 2.2** : Cryptographie symétrique

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

- Les cryptosystèmes par flots
- Les crytosystèmes 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. [7]

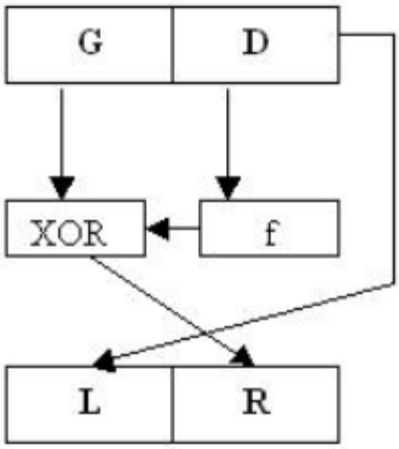



**Figure 2.3** : Réseau de Feistel

## 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 américain) 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. [8]

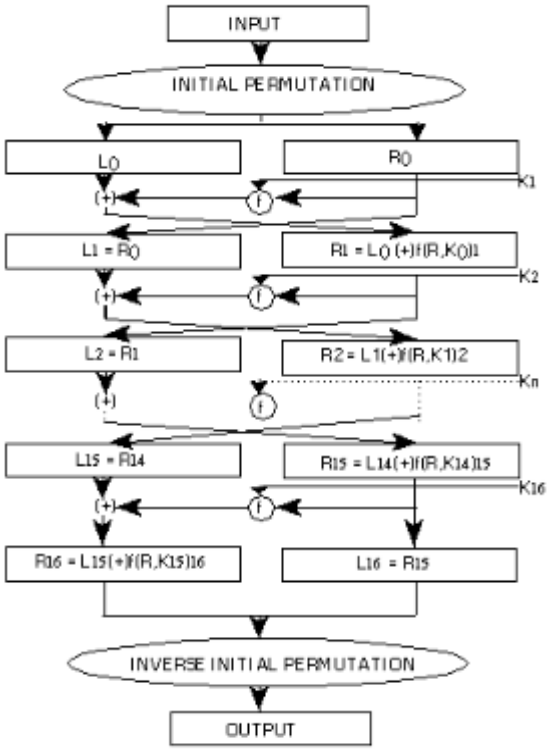



**Figure 2.4** : L'algorithme du DES

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

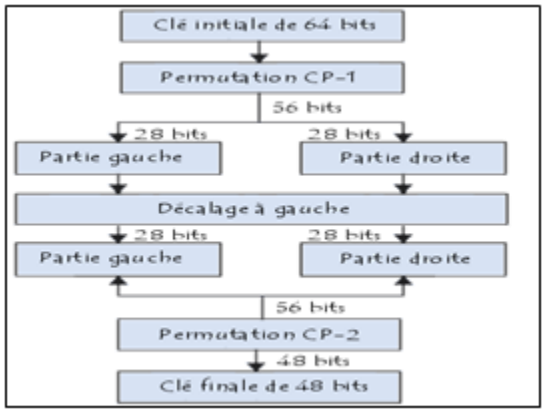



**Figure 2.5** : Génération des clés DES

### B/ Chiffrement par flot

Le chiffrement par flot est un chiffrement a clé symétrique permet 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.

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

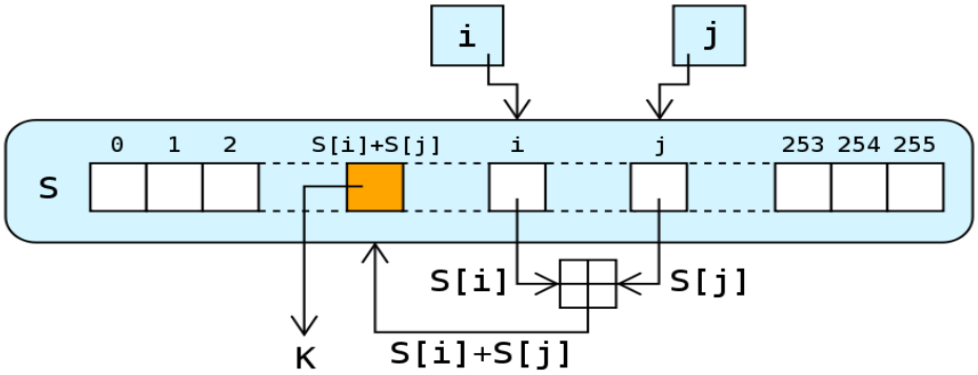



**Figure 2.6** : Chiffrement par flot

## 2.4.2 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. [6]

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 quand 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 $\mathbf{C= M^e \mod n}$ 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 $\mathbf{C^d \mod n}$. 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. [10]

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 $\mathbf{P1 = g^x \mod p}$, et transmet le résultat à Bob
- Bob calcule $\mathbf{P2 = g^y \mod p}$, et transmet le résultat à Alice

- Alice calcule **K1 =P2$^{x}$** mod p, et Bob calcule **K2 = P1$^{y}$ mod p**

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

## 2.5 Conclusion

Dans cette partie nous avons donne de brèves explication des termes suivants : La cryptographie, Quelque techniques de cryptographie classique et moderne (symétrique et asymétrique).

# Chapitre 3
# Chiffrement El Gamal

## 3.1 Introduction

Le besoin d'échanger secrètement des messages et des images sur des réseaux non sécurisés a favorisé la création de systèmes cryptographiques.

L'arithmétique est au cœur du cryptage des communications. Pour crypter un message, on commence par le transformer en un ou plusieurs nombres. Le processus de codage et de décodage fait appel à plusieurs notions mathématiques à savoir :

- Pgcd de deux nombres
- Nombres premiers
- L'algorithme d'Euclide
- Les coefficients de Bézout
- La congruence modulo
- Le petit théorème de Fermat
- Etc.

Dans cette section, nous allons examiner un système de cryptographie à clé publique particulier appelé Chiffrement EL Gamal.

## 3.2 Chiffrement EL Gamal

Le chiffre d'El Gamal est une méthode de cryptographie à clé publique inventée par Taher ElGamal en 1985. Sa sécurité repose, comme le protocole de Diffie et Hellman, sur la difficulté de calculer le logarithme discret.

## 3.3 Principe du chiffrement

Soit un entier premier **p** très grand et p−1 doit avoir un grand facteur premier. On choisit :

- Une clé secrète **s**, telle que **s** ∈ (1...p-2)
- Une clé publique reposant sur :
- L'entier **p**
- Un entier **a** premier avec p, telle que **a** ∈ (1...p-1)
- l'entier **P** telle que $\mathbf{P = a^s \bmod p.}$

Pour chiffrer un message **M**, on choisit un nombre aléatoire **k** telle que **k** ∈ (1...p-2) qui n'est connu que par celui qui chiffre. On notera que la valeur de k n'est utilisée qu'une seule fois, c'est-à-dire le chiffrement d'un seul message. La haute sécurisation de ce chiffrement d'ELGAMEL vient du fait qu'un clair pourra avoir plusieurs chiffrés. Pour chaque k choisie, on calcule alors :

$$\mathbf{C1 = a^k \bmod p.}$$

$$\mathbf{C2 = M.\ P^k \bmod p.}$$

On obtient alors le message chiffré qui est représenté par le couple **C = (C1, C2).** Le message chiffré est alors deux fois plus long que le message original ; donc il faut deux fois plus d'espace mémoire.

### 3.4 Principe du déchiffrement

A la réception du couple (C1, C2) et disposant de la clé secrète 's', on calcule

$$\mathbf{R1 = C1^s \bmod p.}$$

$$\mathbf{= a^{sk} \bmod p.}$$

$$\mathbf{= P^k \bmod p.}$$

La restitution du message clair M (Déchiffrement) est donnée par l'expression suivante:

$$\mathbf{M = C2.\ R1^{-1} \bmod p.}$$

## 3.5 Algorithme d'EL Gamel

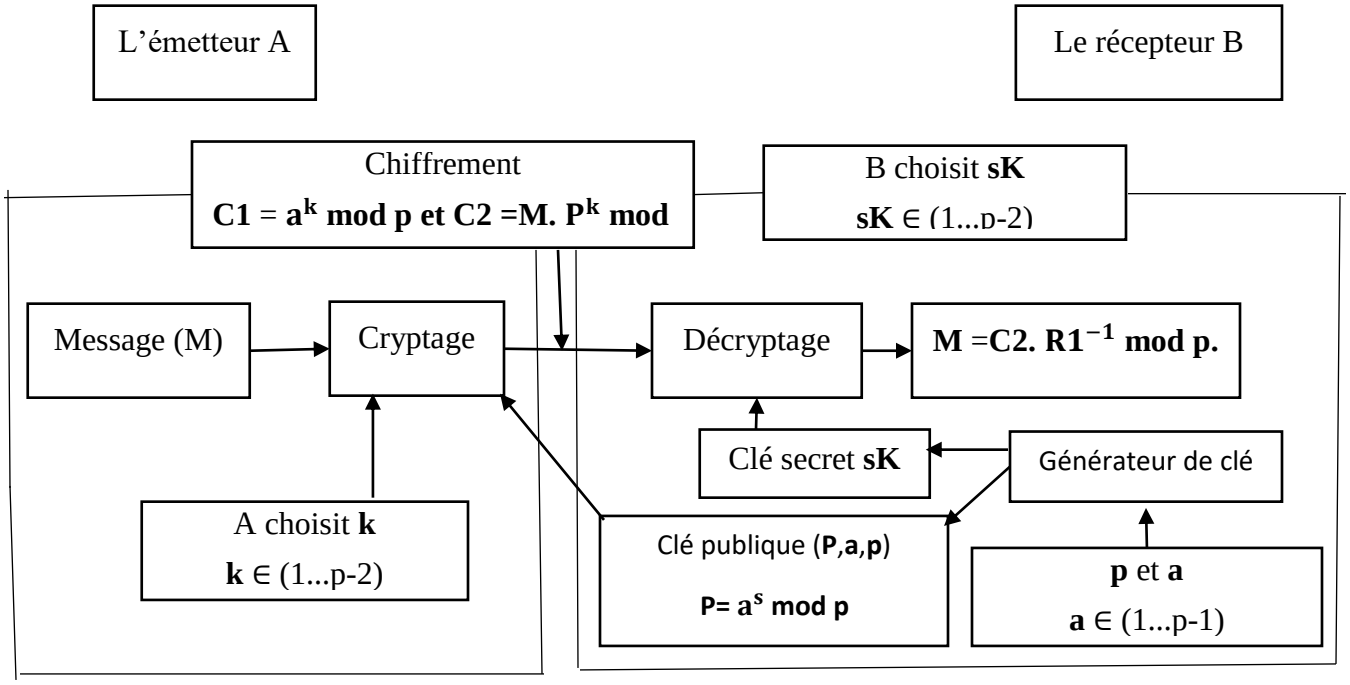



**Figure 3.1** : Algorithme ELGAMAL

## 3.6 Exemple :

Nous voulons chiffrer le mot " supinfo" par le protocole D'ELGAMEL

A cet effet, on choisit : p = 661, a = 23 et une clé secrète s =7

**A) Chiffrement :**

Il faut commencer par convertir ce message en chiffre. Nous assignons un nombre à deux chiffres à chaque caractère en se référant au tableau ci-dessous.

| espace | a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 |
| | n | o | p | q | r | s | t | u | v | w | x | y | x |
| | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

**Tableau 3.1** : Correspondance caractères - nombres

Nous obtiendrons le message en chiffre suivant:

**M = 19 21 16 09 14 06 15**

Ce message est coupé en blocs de même longueur de telle façon que la valeur numérique de chacun de ces blocs devant être inferieur à p = 661. On peut donc pour ce cas former des blocs de taille 3 chiffres.

**M = 192 116 091 406 150**

A noter qu'il faut compléter par des zéros le dernier bloc afin d'aboutir à la taille exigée.

Calculer P :

**$P = a^s \bmod p = 23^7 \bmod 661 = 566$**

La clé publique est donc : (661, 23, 566)

On choisit aléatoirement l'entier k=13

- Le chiffrement du premier le bloc **M1 = 192**

  $C1 = a^k \bmod p = 23^{13} \bmod 661 = 105.$

  $C2 = M1.\ P^k \bmod p = 192.\ 566^{13} \bmod 661 = 237.$

  **M1 chiffré = (105,237)**

- Le chiffrement du deuxième bloc **M2 = 116**

  $C1 = a^k \bmod p = 23^{13} \bmod 661 = 105.$

  $C2 = M2.\ P^k \bmod p = 116.\ 566^{13} \bmod 661 = 515.$

  **M2 chiffré = (105,515)**

- Le chiffrement du troisième bloc **M3 = 091**

  $C1 = a^k \bmod p = 23^{13} \bmod 661 = 105.$

  $C2 = M3.\ P^k \bmod p = 91.\ 566^{13} \bmod 661 = 102.$

  **M3 chiffré = (105,102)**

- Le chiffrement du quatrième bloc **M4 = 406**

  $C1 = a^k \bmod p = 23^{13} \bmod 661 = 105.$

  $C2 = M4.\ P^k \bmod p = 406.\ 566^{13} \bmod 661 = 150.$

  **M4 chiffré = (105,150)**

- Et enfin le chiffrement du cinquième bloc **M5 = 150**

$C1 = a^k \bmod p = 23^{13} \bmod 661 = 105.$

$C2 = M5.\ P^k \bmod p = 150.\ 566^{13} \bmod 661 = 495.$

**M5 chiffré = (105,495)**

**B) Déchiffrement :**

Pour tous les couples C1=105 on a la même valeur de R1

$R1 = C1^s \bmod p = 105^7 \bmod 661 = 466.$

$R1^{-1} = 466^{-1}.$

En utilisant le théorème d'Euclide (PGCD) puis celui de Bézout (calcul de u et v), on aura :

y=-200

k=141.

$R1^{-1} = y \bmod p = -200 \bmod 661 = -200 + 661 = 461.$

**M1** = $C2.\ R1^{-1} \bmod p$ = 237*461 mod 661 = **192.**

**M2** = $C2.\ R2^{-1} \bmod p$ = 515*461 mod 661 = **116.**

**M3** = $C2.\ R3^{-1} \bmod p$ = 102*461 mod 661 = **91.**

**M4** = $C2.\ R4^{-1} \bmod p$ = 150*461 mod 661 = **406.**

**M5** = $C2.\ R5^{-1} \bmod p$ = 495*461 mod 661 = **150.**

Nous obtiendrons donc le message déchiffré qui est un nombre :

M = 192 116 091 406 150

On arrange ce message par **nombre de deux bits** et avec les correspondances caractères - nombres inverses données par le tableau 3.1 nous obtiendrons :

M = 19 21 16 09 14 06 15 00

Ce message correspond bien au message clair qui est : " supinfo "

# Chapitre 4
# Résultats et Discussion

## Résultats et discussions

Le travail de ce mémoire consiste à la simulation, utilisant la méthode de Chiffrement EL Gamal, d'images réelles, synthétiques et médicales.

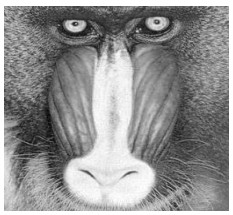
Notre but dans ce mémoire consiste à l'étude du chiffrement d'EL GAMEL et de comparer les résultats de simulation donnés avec d'autres méthodes. La simulation est réalisée sur de différentes images issues de bases de données. Nous procédons à des choix aléatoires des paramètres utilisés par cette méthode à savoir : le nombre premier p, a qui est une racine primitive de p, différentes clé secrètes sk et un ensemble de nombres entiers aléatoires k. Nous essayons par les variations de ces différents paramètres d'aboutir à une qualité de chiffrement/déchiffrement appréciable.

Nous donnons les résultats du chiffrement et du déchiffrement pour les mêmes images utilisées dans d'autres travaux afin de procéder à une étude comparative.

Nous essayons d'argumenter et de commenter ces résultats trouvés en se basant sur les deux paramètres calculés : le PSNR et l'EQM.

La programmation a été faite en langage Matlab (2017b). Les spécifications de PC : Intel (R) Core (TM) i7-5500U CPU @ 3 GHz, 4 GB RAM.

- **Les images utilisées**



**baboon.png**

**(256, 256)**

**barbara.bmp**

**(512, 512)**

**Lena.jpeg**

**(273, 234)**

**cameraman.tif**

**(256, 256)**

**Pepper.bmp**

**(256, 256)**

- **Les paramètres D'EL GAMEL**

  - **Le nombre premier p et a la racine primitive p utilisés**

  Le nombre premier **p** et **a** la racine primitive de **p** utilisés en simulation sont suivantes :

  **p** = 255 et **a** = 59.

  - **Les clés secrètes utilisées (sK)**

  Les différentes clés utilisées en simulation sont suivantes :

  **sk1 = 7, sk2 = 9 et sk3 = 13.**

  - **Les nombres aléatoires utilisés (k)**

  Les différents nombres aléatoires utilisés en simulation sont suivantes :

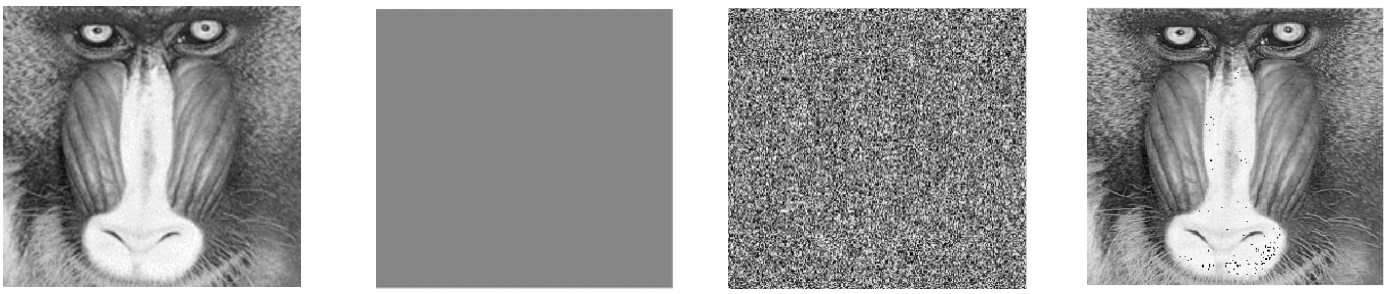
  **k1 = 7, k2 = 9, k3 = 13, k4 = 21 et k5 = 23.**

## 4.1 Les résultats de simulation

### 4.1.1 Les images

**p = 255, a = 59, sK1 = 7.**

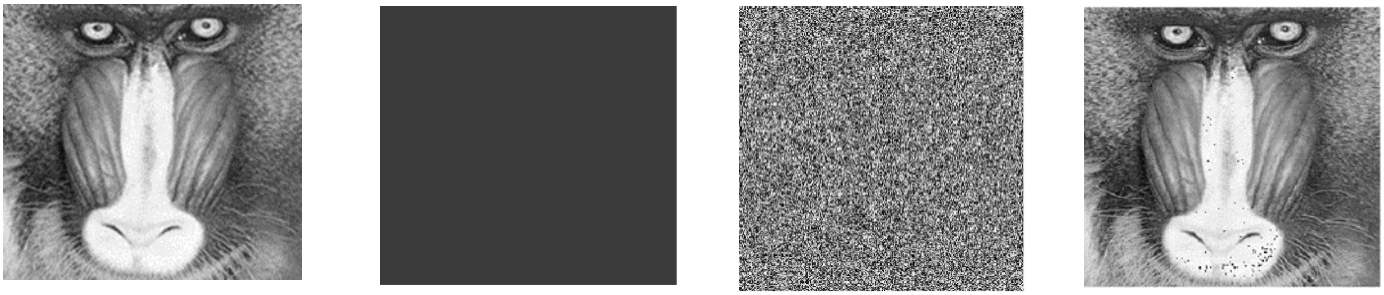
**1. baboon.png (256, 256)**
**k1 = 7**



**(a) (b) (c) (d)**

**Figure 4.1.1:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

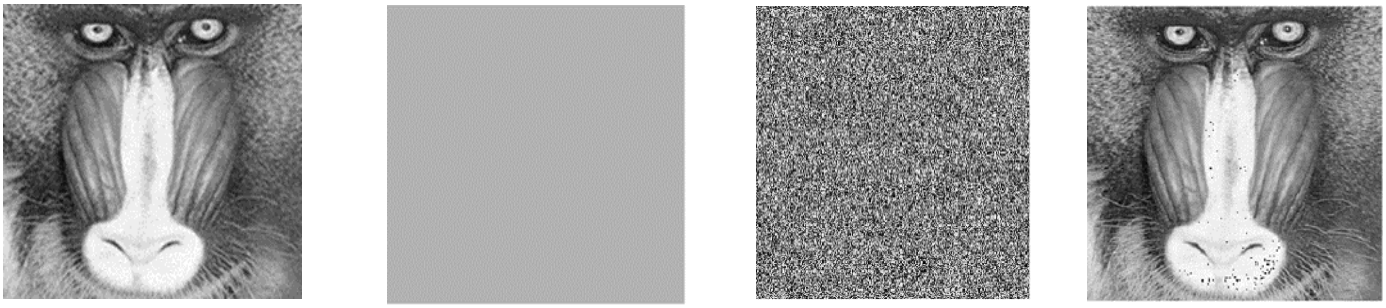
**k2 = 9**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.1.2:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

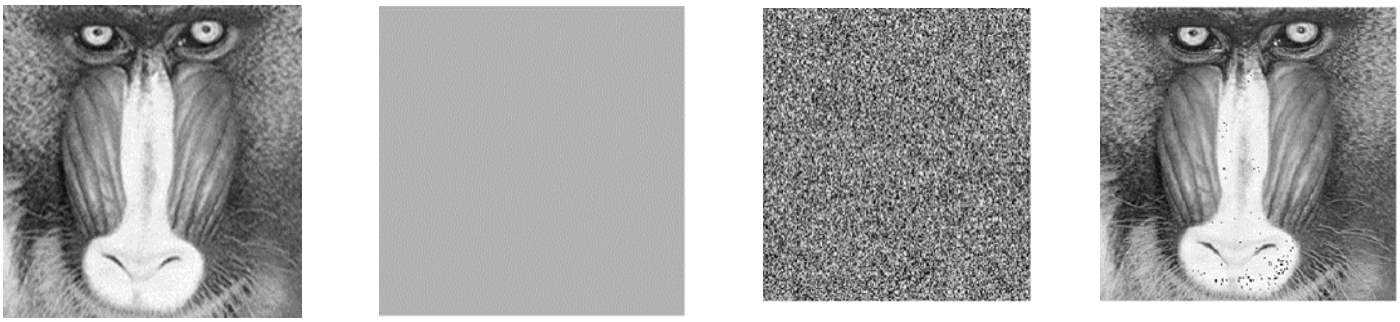
**k3 = 13**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.1.3:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

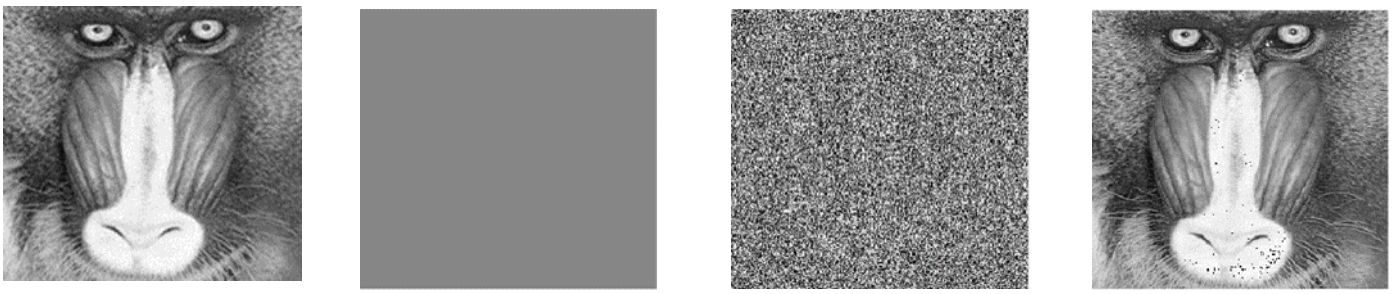
**k4 = 21**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.1.4 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**k5 = 23**

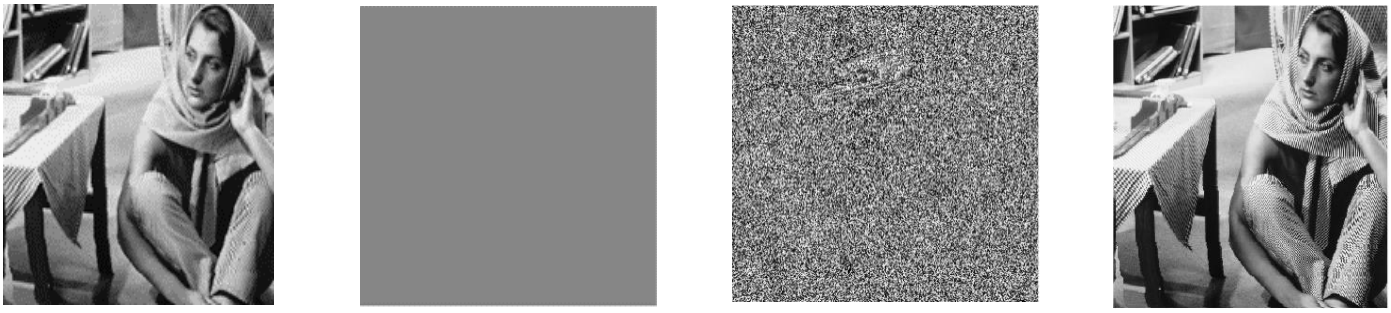


**(a)** **(b)** **(c)** **(d)**

**Figure 4.1.5 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

## 2. barbara.bmp (512, 512)

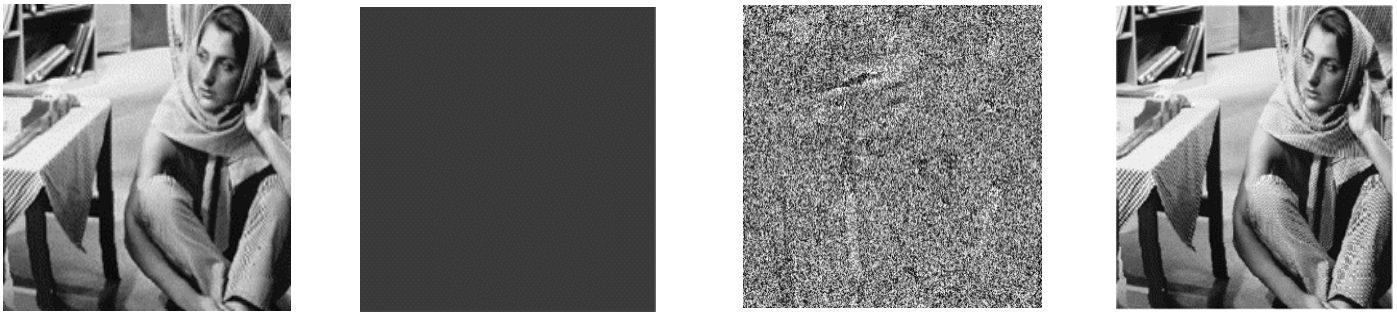
**k1 = 7**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.2.1 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

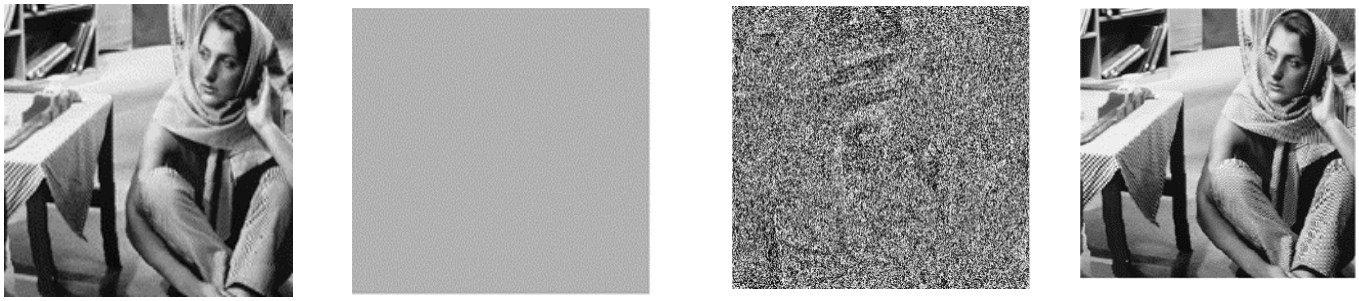
**k2 = 9**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.2.2 :** (a) claire, (b) chiffrée C1, (c) chiffré C2, (d) déchiffrée.

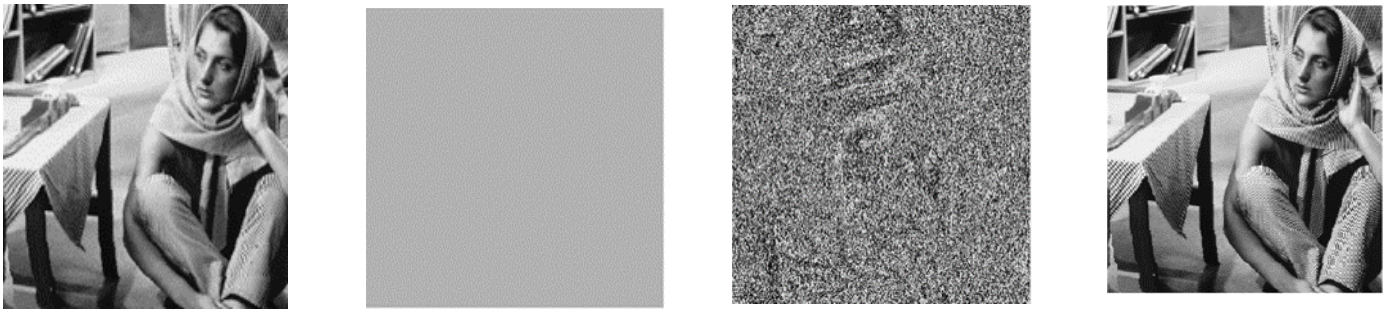
**k3 = 13**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.2.3 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

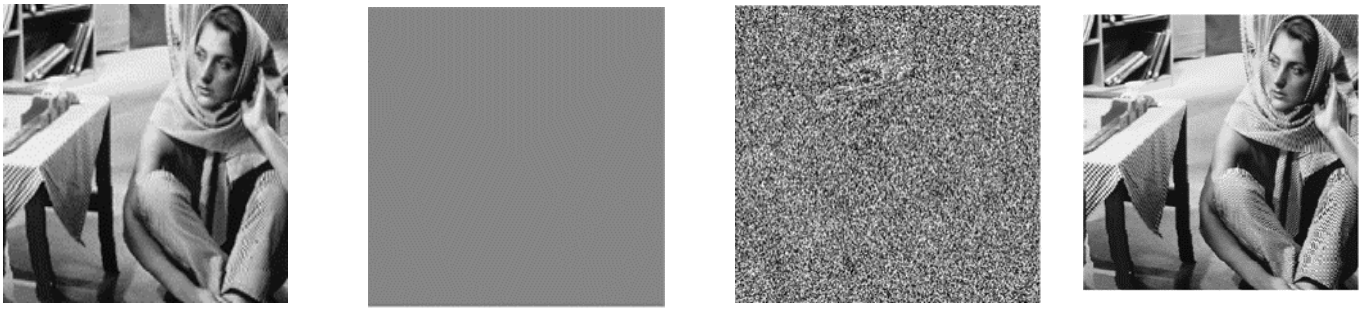
**k4 = 21**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.2.4 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**k5 = 23**

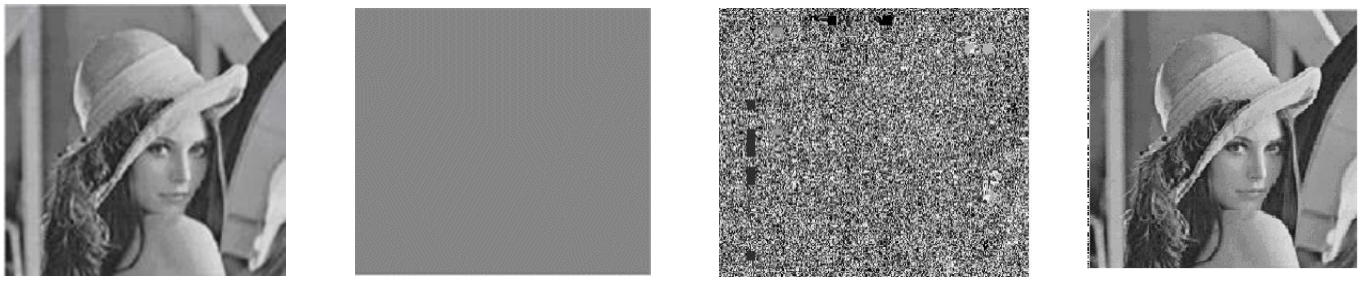


**(a)** **(b)** **(c)** **(d)**

**Figure 4.2.5 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

### 3. Lena.jpeg (273, 234)

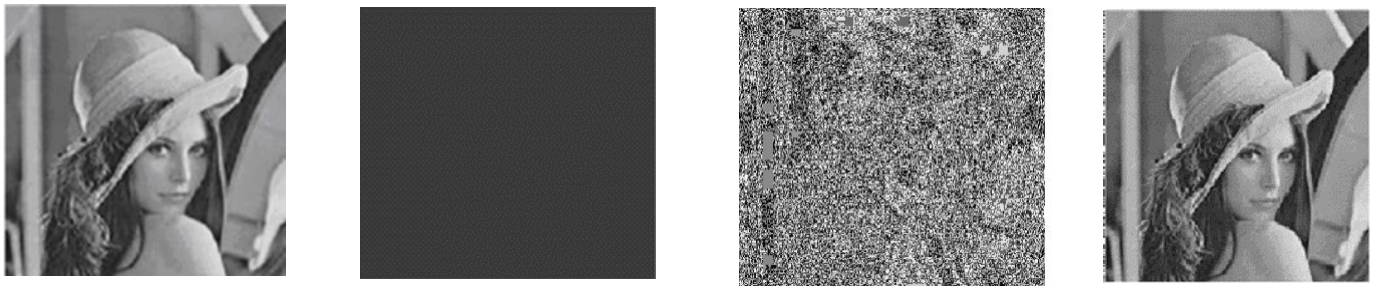
**k1 = 7**



**(a) (b) (c) (d)**

**Figure 4.3.1 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

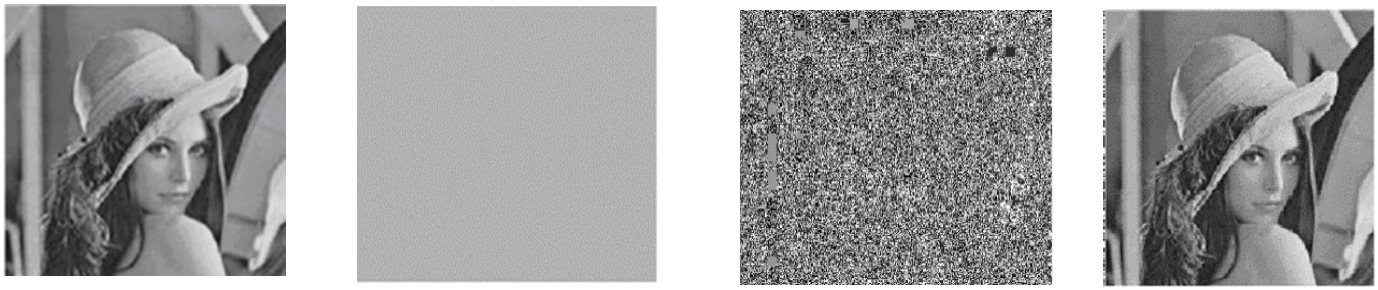
**k2 = 9**



**(a) (b) (c) (d)**

**Figure 4.3.2:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

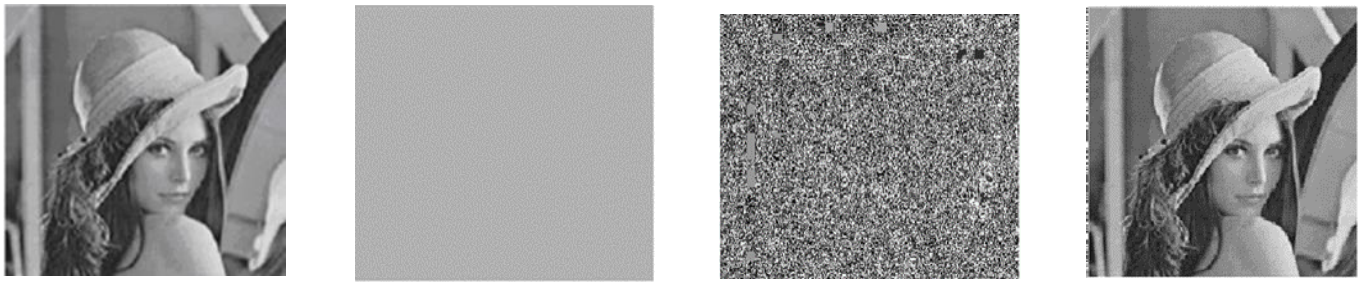
**k3 = 13**



**(a) (b) (c) (d)**

**Figure 4.3.3 :** (a) claire, (b) chiffrée C1, (c) chiffré C2, (d) déchiffrée.

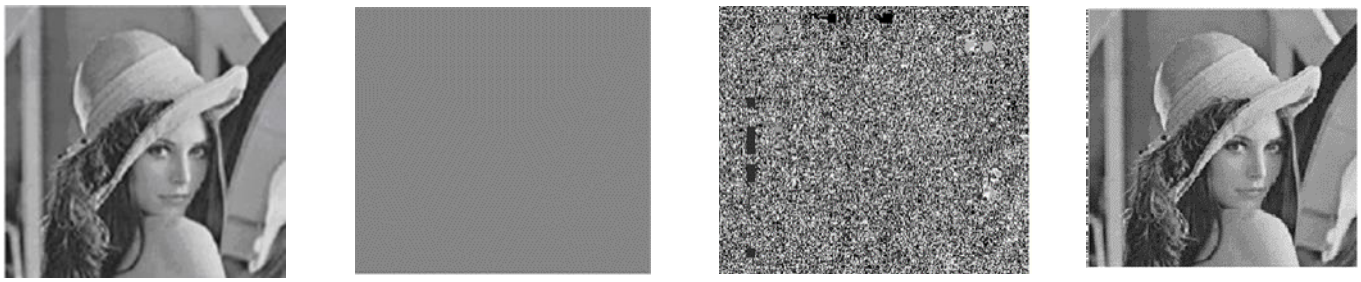
**k4 = 21**



**(a) (b) (c) (d)**

**Figure 4.3.4:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**k5 = 23**

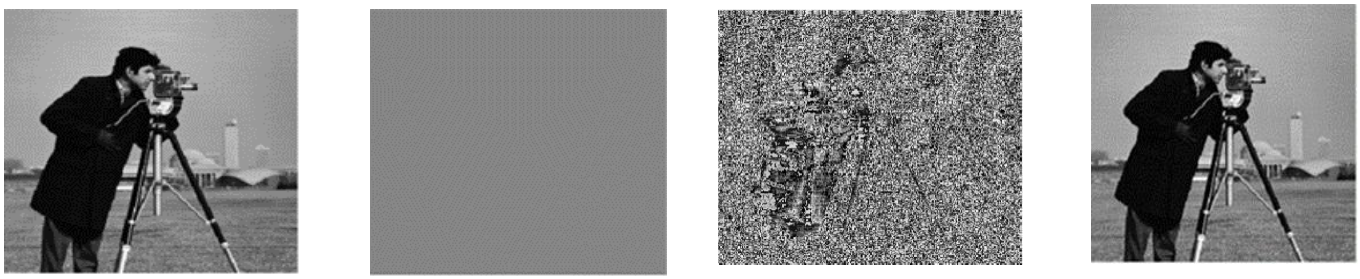


**(a) (b) (c) (d)**

**Figure 4.3.5 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**4. cameraman.tif (256, 256)**

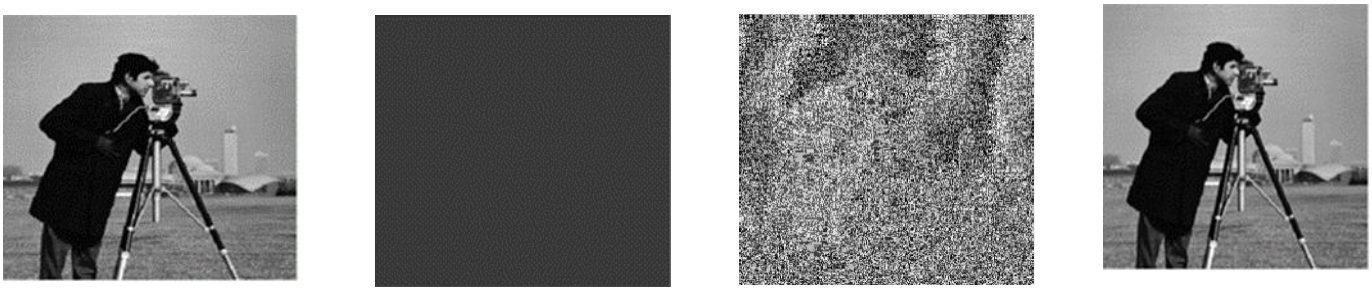
**k1 = 7**



**(a) (b) (c) (d)**

**Figure 4.4.1:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

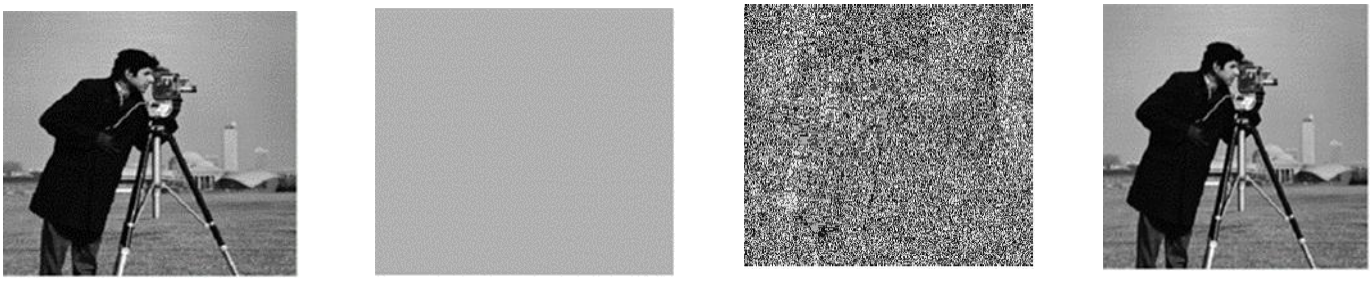
**k2 = 9**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.4.2:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

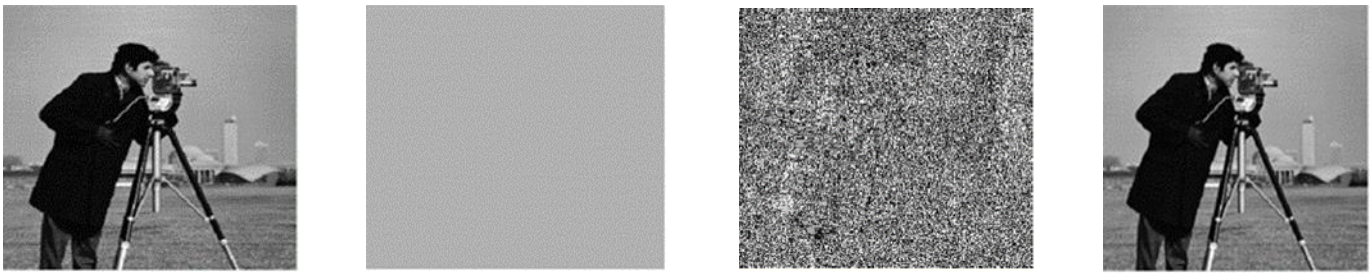
**k3 = 13**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.4.3:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

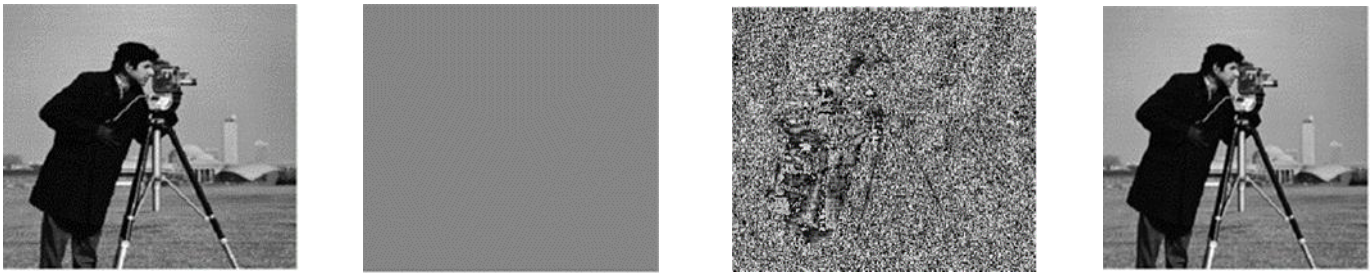
**k4 = 21**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.4.4:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**k5 = 23**

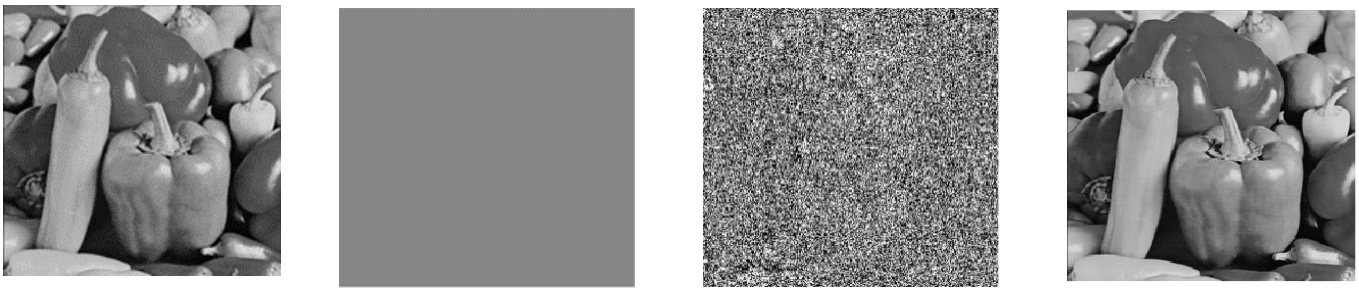


**(a)** **(b)** **(c)** **(d)**

**Figure 4.4.5 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**5. Pepper.bmp (256, 256)**

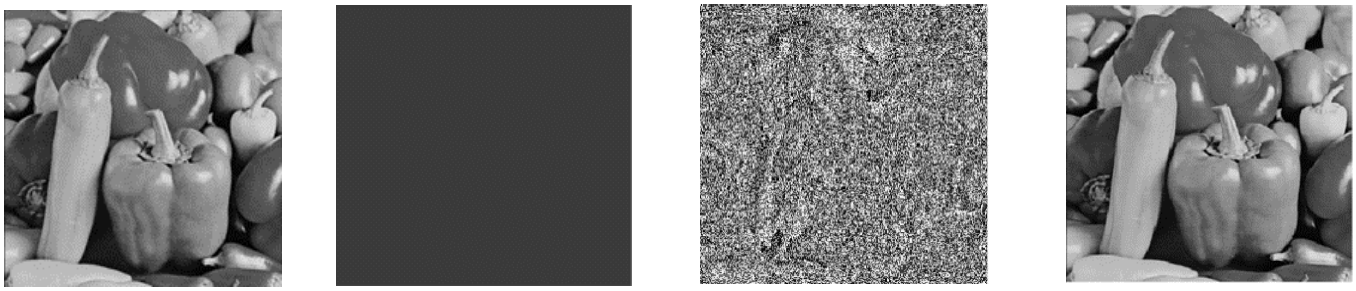
**k1 = 7**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.5.1 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

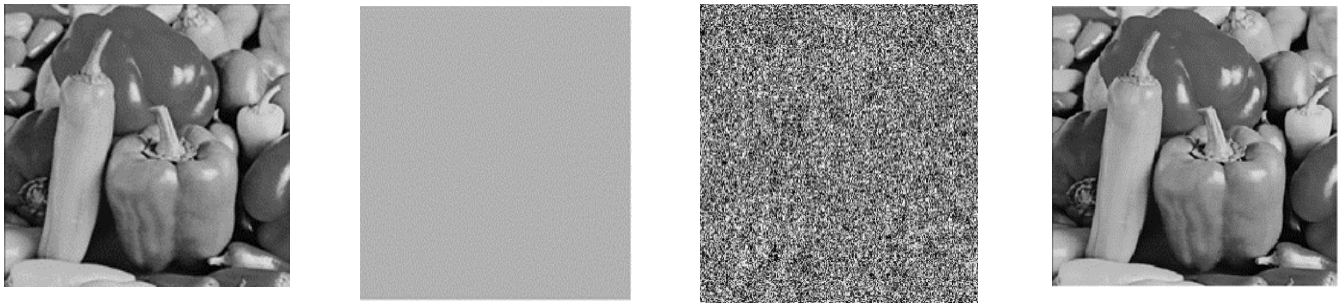
**k2 = 9**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.5.2 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

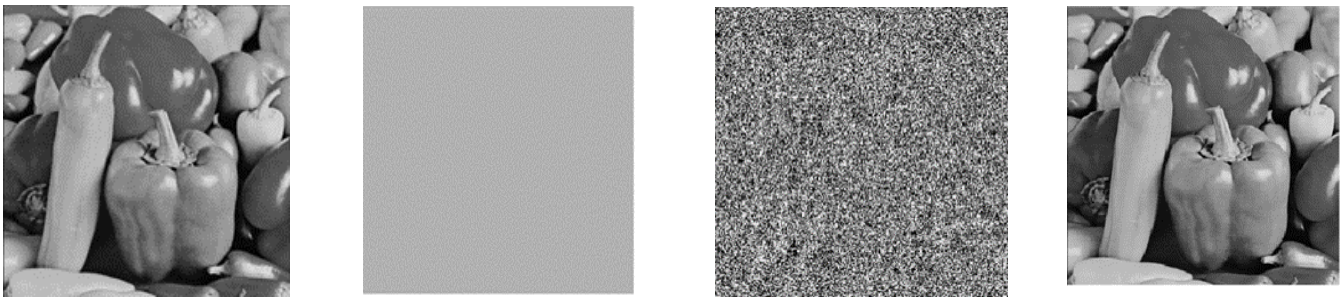
**k3 = 13**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.5.3:** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

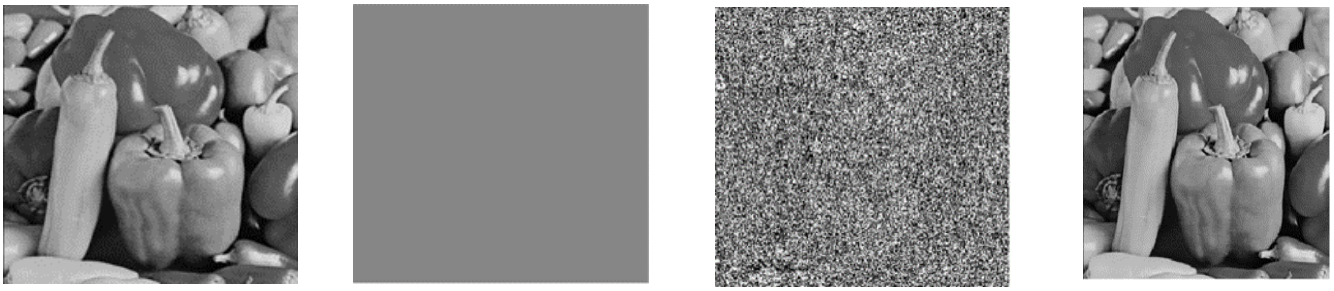
**k4 = 21**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.5.4 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

**k5 = 23**



**(a)** **(b)** **(c)** **(d)**

**Figure 4.5.5 :** (a) claire, (b) chiffrée C1, (c) chiffrée C2, (d) déchiffrée.

### 4.1.2 Valeurs de l'EQM, le PSNR et le temps de simulation

#### A) EQM

- EQM entre l'image claire et l'image chiffrée (C2) :

| sK | 7 | | | | | 9 | | | | | 13 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| k | 7 | 9 | 13 | 21 | 23 | 7 | 9 | 13 | 21 | 23 | 7 | 9 | 13 | 21 | 23 |
| baboon.png (256, 256) | 125.96 | 127.60 | 126.29 | 126.29 | 125.96 | 127.60 | 125.96 | 127.17 | 127.17 | 127.60 | 126.29 | 127.17 | 125.96 | 125.96 | 126.29 |
| barbara.bmp (512, 512) | 111.16 | 111.17 | 120.75 | 120.75 | 111.16 | 111.17 | 111.16 | 117.33 | 117.33 | 111.17 | 120.75 | 117.33 | 111.16 | 111.16 | 120.75 |
| Lena.jpeg (273, 234) | 116.46 | 119.57 | 113.96 | 11396 | 116.46 | 119.57 | 116.46 | 111.44 | 111.44 | 119.57 | 113.96 | 111.44 | 116.46 | 116.46 | 113.96 |
| cameraman.tif (256, 256) | 112.29 | 110.20 | 109.79 | 109.79 | 112.29 | 110.20 | 112.29 | 107.66 | 107.66 | 110.20 | 109.79 | 107.66 | 112.29 | 112.29 | 109.79 |
| Pepper.bmp (256, 256) | 108.70 | 104.96 | 107.75 | 107.75 | 108.70 | 104.96 | 108.70 | 107.99 | 107.99 | 104.96 | 107.75 | 107.99 | 108.70 | 108.70 | 107.75 |

**Tableau 4.1:** l'écart quadratique moyen (EQM) entre l'image claire et l'image chiffrée.

Il faut rappeler que le chiffrement est défini par le couple de valeurs (C1, C2). L'expression de C1 ne contient pas l'information (C1 = $\mathbf{a^k \bmod p}$), donc la valeur de C1 est constante quel soit l'image à chiffrée pour les valeurs données a, k et le modulo p.

Cependant, la valeur de C2 (C2 = M. $P^k$ mod p) dépend de l'mage claire à chiffrée. Pour une image donnée, l'EQM entre le clair et le chiffré C2 est pratiquement constant, quel que soient les valeurs de k et de la clé secrète sK.

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

| sK | 7 | | | | | 9 | | | | | 13 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **k** | **7** | **9** | **13** | **21** | **23** | **7** | **9** | **13** | **21** | **23** | **7** | **9** | **13** | **21** | **23** |
| **baboon.png (256, 256)** | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 | 0.44 |
| **barbara.bmp (512, 512)** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **Lena.jpeg (273, 234)** | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 | 0.55 |
| **cameraman.tif (256, 256)** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **Pepper.bmp (256, 256)** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

**Tableau 4.2 :** EQM entre l'image claire et l'image déchiffrée

De même nous constatons que d'après les résultats donnés au tableau 4.2 que l'EQM entre le clair et le déchiffré est aussi contant pour une image donnée quel que soient les valeurs de sK, a, k et p. L'image restituée M = C2. $R1^{-1}$ mod p ($R1 = C1^{s}$ mod p) est la même image claire pour les images de simulation c'est-à-dire un EQM nul pour Barbara, cameraman et Pepper. Dans notre cas de simulation, on n'a pas pris en considération les bruits introduits par le canal de transmission. Pour les EQM nuls, l'atténuation introduite par le canal rajoute un bruit, qui rend EQM non nul.

A travers la lecture des tableaux 4.1 et 4.2, il ressort que les images Barbara, cameraman et Pepper sont bien chiffrées et déchiffrées par cette méthode d'ELGAMEL.

### B) PSNR:

- Les résultats du PSNR entre l'image claire et l'image chiffrée (C2) :

| sK | 7 | | | | | 9 | | | | | 13 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| k | 7 | 9 | 13 | 21 | 23 | 7 | 9 | 13 | 21 | 23 | 7 | 9 | 13 | 21 | 23 |
| baboon.png (256, 256) | 27.12 | 27.07 | 27.11 | 27.11 | 27.12 | 27.07 | 27.12 | 27.08 | 27.08 | 27.07 | 27.11 | 27.08 | 27.12 | 27.12 | 27.11 |
| barbara.bmp (512, 512) | 27.67 | 27.67 | 27.31 | 27.31 | 27.67 | 27.67 | 27.67 | 27.43 | 27.43 | 27.67 | 27.31 | 27.43 | 27.67 | 27.67 | 27.31 |
| Lena.jpeg (273, 234) | 27.46 | 27.35 | 27.56 | 27.56 | 27.46 | 27.35 | 27.46 | 27.66 | 27.66 | 27.35 | 27.56 | 27.66 | 27.46 | 27.46 | 27.56 |
| cameraman.tif (256, 256) | 27.62 | 27.70 | 27.72 | 27.72 | 27.62 | 27.70 | 27.62 | 27.80 | 27.80 | 27.70 | 27.72 | 27.80 | 27.62 | 27.62 | 27.72 |
| Pepper.bmp (256, 256) | 27.76 | 27.92 | 27.80 | 27.80 | 27.76 | 27.92 | 27.76 | 27.79 | 27.79 | 27.92 | 27.80 | 27.79 | 27.76 | 27.76 | 27.80 |

**Tableau 4.3:** Le PSNR entre l'image claire et l'image chiffrée.

Le PSNR est inversement proportionnel à l'EQM. En cryptographie, un bon chiffrement correspond à un EQM élevé c'est à dire un PSNR faible. Dans la recherche en cryptographie un PSNR inférieur à 30 dB est appréciable pour le chiffrement, quand au déchiffrement il faudrait plus de 40dB.

Les résultats obtenus par le calcul du PSNR de chiffrement tournent autour de 27,50 dB ; c'est un bon chiffrement qui est difficilement cassable.

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

| sK | 7 | | | | | 9 | | | | | 13 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| k | 7 | 9 | 13 | 21 | 23 | 7 | 9 | 13 | 21 | 23 | 7 | 9 | 13 | 21 | 23 |
| baboon.png (256, 256) | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 | 51.62 |
| barbara.bmp (512, 512) | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> |
| Lena.jpeg (273, 234) | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 | 50.65 |
| cameraman.tif (256, 256) | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> |
| Pepper.bmp (256, 256) | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> | >> |

**Tableau 4.4 :** Le PSNR entre l'image claire et l'image déchiffré.

Quand au déchiffrement il faudrait un PSNR supérieur à 40dB pour dire qu'il ya une bonne restitution de l'image originale. Dans notre cas de simulation, les résultats donnés par le tableau 4.4 de la méthode de chiffrement à clé secrète d'ELGAMAL sont supérieur à 50 dB ce qui est très acceptables. La même remarque faite lors de l'étude de L'EQM un PSNR infini, ce qui correspond à un EQM nul, n'existe pas.

### C) Temps de simulation (seconde)

- Le temps de chiffrement **:**

| **sK** | **7** | | | | | **9** | | | | | **13** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **k** | **7** | **9** | **13** | **21** | **23** | **7** | **9** | **13** | **21** | **23** | **7** | **9** | **13** | **21** | **23** |
| **baboon.png (256, 256)** | 0.027 | 0.015 | 0.016 | 0.016 | 0.016 | 0.018 | 0.016 | 0.018 | 0.017 | 0.033 | 0.021 | 0.020 | 0.021 | 0.021 | 0.021 |
| **barbara.bmp (512, 512)** | 0.043 | 0.041 | 0.043 | 0.044 | 0.043 | 0.047 | 0.046 | 0.048 | 0.047 | 0.046 | 0.057 | 0.054 | 0.057 | 0.057 | 0.056 |
| **Lena.jpeg (273, 234)** | 0.035 | 0.033 | 0.034 | 0.035 | 0.034 | 0.039 | 0.037 | 0.039 | 0.039 | 0.038 | 0.045 | 0.043 | 0.045 | 0.045 | 0.045 |
| **cameraman.tif (256, 256)** | 0.047 | 0.034 | 0.036 | 0.036 | 0.036 | 0.039 | 0.038 | 0.041 | 0.039 | 0.039 | 0.047 | 0.045 | 0.049 | 0.047 | 0.049 |
| **Pepper.bmp (256, 256)** | 0.017 | 0.015 | 0.016 | 0.016 | 0.017 | 0.018 | 0.018 | 0.019 | 0.018 | 0.018 | 0.021 | 0.020 | 0.021 | 0.022 | 0.028 |

**Tableau 4.5:** Le temps de chiffrement.

- Le temps de déchiffrement **:**

| **sK** | **7** | | | | | **9** | | | | | **13** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **k** | **7** | **9** | **13** | **21** | **23** | **7** | **9** | **13** | **21** | **23** | **7** | **9** | **13** | **21** | **23** |
| **baboon.png (256, 256)** | 0.007 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.007 | 0.006 | 0.004 | 0.009 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 |
| **barbara.bmp (512, 512)** | 0.012 | 0.011 | 0.017 | 0.012 | 0.012 | 0.012 | 0.017 | 0.012 | 0.013 | 0.012 | 0.012 | 0.011 | 0.012 | 0.012 | 0.012 |
| **Lena.jpeg (273, 234)** | 0.009 | 0.009 | 0.009 | 0.010 | 0.009 | 0.009 | 0.014 | 0.009 | 0.009 | 0.009 | 0.013 | 0.009 | 0.010 | 0.010 | 0.010 |
| **cameraman.tif (256, 256)** | 0.010 | 0.010 | 0.011 | 0.010 | 0.010 | 0.015 | 0.010 | 0.010 | 0.012 | 0.010 | 0.010 | 0.010 | 0.010 | 0.010 | 0.010 |
| **Pepper.bmp (256, 256)** | 0.004 | 0.004 | 0.004 | 0.004 | 0.005 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.005 | 0.004 | 0.004 | 0.004 | 0.004 |

**Tableau 4.6:** Le temps de déchiffrement.

Le temps de simulation (Chiffrement ou déchiffrement) dépend de plusieurs facteurs dont le plus important est celui des dimensions de l'image. Le second facteur repose sur les niveaux de gris de l'image

Dans notre cas de simulation, l'image '**Pepper'** est pratiquement de couleur blanche (Niveaux de gris très élevés), les temps de chiffrement et de déchiffrement sont très courts. Par contre l'image **'cameraman'** est une image très noire (Niveaux de gris très faibles), les temps de simulation sont élevés.

Le troisième facteur qui est le plus important est l'outil de simulation, c'est-à-dire les caractéristiques des micro-ordinateurs utilisés (fréquence d'horloge, RAM etc.).

## D) Etude comparative avec d'autres travaux de recherche :

| | WT [11] | Chaotic Map [12] | The grayscale [13] | WIFI AVC [14] | Quadtree et AES [15] | Steno Technique [16] | Méthode proposée ELGAMEL | |
|---|---|---|---|---|---|---|---|---|
| **Images** | **Cryptage** | **Décryptage** | **Décryptage** | **Cryptage** | **Décryptage** | **Décryptage** | **Cryptage** | **Décryptage** |
| **Lena** | 29.02 | 36.5 | 48.53 | | 35.08 | 44.98 | 27.40 | 50.65 |
| **Barbara** | | | 44.70 | | | | 27.67 | > 50 |
| **Cameraman** | 28.37 | 37.4 | 44.43 | 28.9 | 35.6 | | 27.62 | >50 |
| **baboon** | | | | | | 41.78 | 27.12 | 51.62 |
| **Pepper** | | | | | | 43.68 | 27.80 | > 50 |

**Tableau 4.7:** Le PSNR de chiffrement et de déchiffrement de quelques méthodes récentes.

Dans cette partie du mémoire nous nous sommes intéressées aux résultats données par quelques articles publiés les cinq dernières années et nous les avons regroupés dans ce tableau 4.5. Pour le chiffrement de données, les PSNR des méthodes étudiées en [11] et [12] se rapproche considérablement de celle de la méthode D'ELGAMEL. Les méthodes restantes, dans leurs papiers ne donnent pas les PSNR de chiffrement. Par contre pour la restitution de l'image (Déchiffrement), dans tous les cas, les PSNR donnés par l'application de cette méthode D'ELGAMEL donne les meilleurs résultats ; un PSNR supérieur à 50 dB.

# Conclusion générale

## Conclusion générale

Ce mémoire de fin d'études nous a permis de comprendre un peu plus ce domaine de la cryptographie ; domaine qui est en plein évolution.

Le chiffrement d'ELGAMEL est un ancien protocole qui n'est utilisé que pour les transactions de courts textes. Nous l'avons exploité dans ce travail pour le chiffrement d'images et étudier son comportement pour différentes données issues d'images.

Nous constatons que ce chiffrement bi alphabétique qui est utilisé uniquement pour des messages courts pourrait être utilisé pour le chiffrement d'une masse de données importantes.

Le décryptage d'images traité dans ce mémoire utilisant cette méthode donne de très bons résultats. Cependant le problème réside dans le cryptage qui demande un espace mémoire double du au fait des calculs de C1 et C2. Ce calcul supplémentaire influe sur les temps de simulation qui devient très important. De ce fait, cette méthode D'ELGAMEL ne peut pas être utilisée pour un chiffrement en temps réel.

En perspectives nous pouvons, pour des systèmes ne travaillant pas en temps réel, améliorer la qualité du chiffrement en l'associant avec d'autres protocoles : utiliser un chiffrement hybride.

Aussi, les données chiffrées ne seront pas transférées en mode simple à savoir le mode ECB ; nous pouvons utiliser d'autres modes.

Une compression de données comme traitement préliminaire avant l'application du chiffrement est à explorer.

# Références

## Références


[1] http://www.univorleans.fr/mapmo/membres/louchet/teaching/timo/ben_hamadi/rapport_benhamadi.pdf

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

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

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

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

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

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

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

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

[10] https://medium.com/@antoine.ansel/l-algorithme-d-échange-de-clés-diffie-hellman-6f9681d1418c.

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

[12] Chaotic map, Journal of Kerbala University, Vol. 14 No.1 pages 186-198 Scientific. 2016

[13] Encrypt the gray scale image International Journal of Scientific and Research Publications, Volume 2, Issue 7, July 2012 ISSN 2250-3153.

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

ARPN Journal of Engineering and Applied Sciences.

[15] Crypto-Compression d'Images Fixes Par la méthode de Quadtree optimisée et AES S. Ftérich C. Ben Amar REGIM (Groupe de REcherche sur les Machines Intelligentes) Ecole Nationale d'Ingénieurs de Sfax (ENIS) Route de Soukra, B.P. W, 3038 Sfax – Tunisie.

[16] Data Security Using Cryptography and Steganography Techniques Cryptography and Steganography Techniques (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 6, 2016.