REPUBLIQUE ALGERIENNEDEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENTSUPERIEUR ET DE LA RECHERCHESCIENTIFIQUE

UNIVERSITE LARBI TEBESSI - TEBESSA
FACULTE DES SCIENCES ET TECHNOLOGIES
DEPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE

DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOMEDE MASTEREN
Réseaux et Télécommunications

THEME

Application des techniques de cryptage pour la transmission sécurisée des images

Présenté par le binôme :

- **Chaker BOUAKKAZ**
- **Yassine SADOUN**

Devant le jury :

-Karim FERROUDJI	**Président**
-Riad SAIDI	**Encadreur**
-Tarek BENTAHAR	**Examinateur**

Année universitaire 2019/2020


بسم الله الرحمن الرحيم

Remerciement

Avec un grand plaisir nous remercie Allah qui nous a aidés et nous a donné

la patience, le courage et la force d'achever ce travail.

Nous tenons à remercier en cette occasion tout le corps professoral et administratif de Département de Génie Electrique de l'université CHIKH LARBI TEBESSI de Tébessa pour la richesse et la qualité de leurs enseignements et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Je tiens à remercier sincèrement **Dr SAIDI Riad**, qui, en tant qu'encadreur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'orientation, la confiance, l'aide et le temps qu'il a bien voulu me consacrer et sans lui ce mémoire n'aurait jamais vu le jour.

J'exprime également ma gratitude aux membres du jury, le président Mr FEROUDJI Karim, l'examinateur Mr BENTAHAR Tarek, qui m'ont honoré en acceptant de juger ce modeste travail.

Nous tenons à remercier sincèrement nos parents, tout la famille BOUAKKAZ et SADOUN, et nos amis, qui nous ont donné le courage.

Nous souhaitons d'adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

Thank

You

Dédicace

A l'aide du DIEU le tout puissant, qui trace le chemin de ma vie, j'ai puréaliser ce modeste travail que je dédie :

A mon très cher père Abdallah et ma très chère mère Akila qui n'ont pas cessé de m'encourager et de se sacrifier pour que je puisse franchir tout obstacle durant toutes mes années d'étude que Dieu me les garde en très bonne santé ; Aucune dédicace ne pourra compenser les sacrifices de mes parents, celui à qui je souhaite une longue vie pleine de joie, de bonheur et de santé ;

A mon frère Djamel et mon petit chère frère Ihab ;

A mes sœurs Rimal, Asma et Basma ;

A mon collègue, mon frère Yassine SADOUN, celui à qui je souhaite une vie pleine de succès.

Et à mes collègues et tous ceux qui m'aiment et qui me cannaient de proche ou de loin.

Chaker

Dédicace

Grâce au DIEU le tout puissant, qui illumine notre chemin et le rend plus paisible et clair , j'ai pu en faire un modeste travail tel que celui qui est rédigé en ce mémoire dont je le dédie :

A ceux qui m'ont permis d'accéder à ce monde et de voir la lumière du jour : mon père et ma mère qui n'ont jamais cessé de m'encourager et de se sacrifier pour que je puisse franchir tout obstacle durant toutes mes années d'étude que Dieu me les garde en très bonne santé ; Aucune dédicace ne pourra compenser les sacrifices de mes parents, celui à qui je souhaite une longue vie pleine de joie, de bonheur et de santé ;

A mon frère Sarem

A mes sœurs Ilham et samiha ;

A mon collègue, mon frère Chaker BOUAKKAZ, celui à qui je souhaite une vie pleine de réussite.

Et à mes collègues et tous ceux qui m'aiment et qui me cannaient de proche ou de loin.

Yassine

Tables de matières

Liste des figures

Liste des tableaux

Liste des Acronyme

- **A**
  - **AES :** Advanced Encryption Standard
  - **ASCII :** American Standard Code for Information Interchange
- **C**
  - **CBC :** Cipher Block Chaining
  - **CFB :** Choper FeedBack
  - **CTR :** CounTeR
- **D**
  - **DES :** Data Encryption Standard
- **E**
  - **ECB :** Electronic Code Book
  - **ECC :** Elliptic Curve Cryptography
- **L**
  - **LSFR :** Linear Feedback Shift Register
- **M**
  - **MSE :** Mean Squared Error
- **O**
  - **OFB :** Output FeedBack
- **P**
  - **PSNR :** Peak Signal to Noise Ratio
- **R**
  - **RC4 :** Rivest Cipher
  - **RSA :** Rivest Shamir Adelman
- **S**
  - **SSIM :**Structural Similarité Mesure
- **3**
  - **3DES :** Triple Data Encryption Standard

Résumé

A Nos jours, il est difficile de s'en passer de l'utilisation des moyens informatiques pour l'échange de l'information, que ce soit de la voix, des images ou d'autres. Souvent ces moyens de communication sont liés à des réseaux ouverts via des liaisons sans fil non sécurisés. Ce qui rend l'information échangée plus vulnérables. En fait, l'évolution des systèmes embarqués permet de développer des crypto-systèmes de plus en plus complexes

Ce thème entre dans le domaine cryptographique, à travers une application des techniques de cryptage pour la transmission sécurisée des images, en se basant sur le chiffrement avec l'algorithme AES, qui va nous garantir la confidentialité.

Dans ce travail en ces penchés sur le côté cryptographique, à travers une l'application se basant sur les techniques de cryptage pour la transmission sécurisée des images, en utilisant l'algorithme de chiffrements AES avec les modes ECB, CBC, OFB et CFB, ainsi que la conception d'une interface graphique qui nous facilite les opérations de chiffrement et déchiffrement.

Abstract

Nowadays, it is difficult to do without the use of computer means for the exchange of information, be it voice, images or others. Often these means of communication are linked to open networks via unsecured wireless links. This makes the information exchanged more vulnerable. In fact, the evolution of embedded systems makes it possible to develop increasingly complex cryptosystems. This theme enters the cryptographic domain, through an application of encryption techniques for the secure transmission of images, based on encryption with the AES algorithm, which will guarantee us confidentiality.
4In this work in these examined on the cryptographic side, through an application based on encryption techniques for the secure transmission of images, using the AES encryption algorithm with ECB modes, CBC, OFB and CFB, as well as the design of a graphical interface that facilitates encryption and decryption operations.

ملخص

في الوقت الحاضر، من الصعب القيام بذلك دون استخدام وسائل الحاسوب لتبادل المعلومات، سواء كان الصوت أو الصور أو غيرها.غالبًا ما ترتبط وسائل الاتصال هذه بالشبكات المفتوحة عبر ارتباطات لاسلكية غير آمنة.وهذا من شأنه أن يجعل المعلومات المتبادلة أكثر عُرضة للخطر. والواقع أن تطور النظم المتأصلة يجعل من الممكن تطوير نظم تشفير متزايدة التعقيد.
يدخل هذا الموضوع في مجال التشفير، من خلال تطبيق تقنيات التشفير لنقل الصور بشكل آمن، استنادًا إلى التشفير باستخدام خوارزمية AES، التي تضمن لنا السرية.

وفي هذا العمل الذي تم فحصه في جانب التشفير، من خلال تطبيق يستند إلى تقنيات التشفير لنقل الصور بشكل آمن، باستخدام خوارزمية التشفير AES مع ECB، وCBC، وOFB، وCFB، بالإضافة إلى تصميم واجهة رسومية تسهل عمليات التشفير وإلغاء التشفير .

Introduction générale

Depuis le temps, l'humanité a essayé de transmettre des informations d'une façon sécurisée. Pour l'échange de données secrètes, en a toujours utilisés le chiffrement d'information comme instrument de sécurisation dans les stratégies des guerres. 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. Les images transmises sur ces réseaux sont des données particulières du fait de leur quantité importante d'information. La transmission des images rassemble un nombre important de problèmes. Nous citons, par exemple la confidentialité, l'authentification et l'intégrité des données :

- La confidentialité se base sur les concepts qui permettent de s'assurer que l'information ne puisse pas être lue par des personnes non autorisées. Elle est fortement liée à la cryptographie.
- L'authentification est l'ensemble des moyens qui permettent d'assurer que les données reçues et envoyées proviennent bien des entités déclarées.
- L'intégrité des données concerne les techniques qui nous permettent d'avoir la possibilité de la vérification du contrôle du contenu.

De nombreux algorithmes de chiffrement existes pour répondre aux besoins de la sécurités ; citons parmi elles, l'algorithme public symétrique AES (Advanced Encryption Standard) [1] [2] qui a prouvé de nos jours sa robustesse contre les différents types d'attaques, l'algorithme asymétrique RSA (Rivest, Shamir and Adleman) [2], et l'algorithme IDEA (International Data Encryption Algorithme) [3].

L'application de ces algorithmes indépendamment pour la transmission des images, ne peuvent assurer que la confidentialité, donc il est toujours recommander des Cryptosystème de chiffrement hybride basé sur deux algorithmes ou plus pour avoir plus de sécurité : la confidentialité, l'authenticité, l'intégrité.

Dans notre travail est comme un premier pas, nous voulons assurer la confidentialité, pour cela notre choix est tombé sur l'algorithme AES avec ses cinq modes de fonctionnement [2] [3] [4], car il consomme peu de mémoire et n'étant pas basé sur un schéma de Feistel. Sa complexité est moindre et est plus facile à implémenter. Il est très rapide et n'est pas cassé jusqu'à nos jours. Il est le successeur de DES [5]. Nous analysons les résultats de chiffrement en utilisons quelques métrises.

Ce mémoire s'articule autour de quatre chapitres :

Le premier chapitre traite le domaine de la cryptographie en générale, ou nous avons abordé deux grandes classes de la cryptographie telle que cryptographie classique et moderne.

Le deuxième chapitre représente généralement la cryptographie symétrique et la cryptographie asymétrique telle que les algorithmes AES, RSE.

Le troisième chapitre est consacré à l'application des techniques des cryptages pour la transmission sécurisée des images basé sur les chiffrements AES avec les modes ECB, CBC, OFB et CFB, ainsi de la conception d'une interface graphique qui nous facilite les opérations de chiffrement et déchiffrement

Finalement nous terminerons notre travaille par une conclusion.

1.1 Introduction

La sécurité informatique est un domaine très important dans notre vie qui protège les informations et les données dans les réseaux de communication, donc pour garantir la sécurité d'informatique en utilise la cryptographie.

La cryptographie est l'art du secret désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles.

Dans ce chapitre, nous expliquons les terminologies de base de la cryptographie, puis nous allons parler sur leurs objectifs et ces différents types. Enfin, en termine par les types des attaques.

1.2 Historique de cryptographie

Les origines de la cryptologie remontent à l'antiquité ; son utilisation était alors très rudimentaire. Cependant la cryptologie a rapidement révélé sa double nature, celle de l'étude combinée de deux arts opposés mais complémentaires : la cryptographie, qui consiste à protéger des informations, et la cryptanalyse, qui est l'art de les retrouver ou de les exploiter. La clé breouvrage de Kahn, « The Code breakers » [6], retrace la longue histoire de la cryptologie à travers les âges.

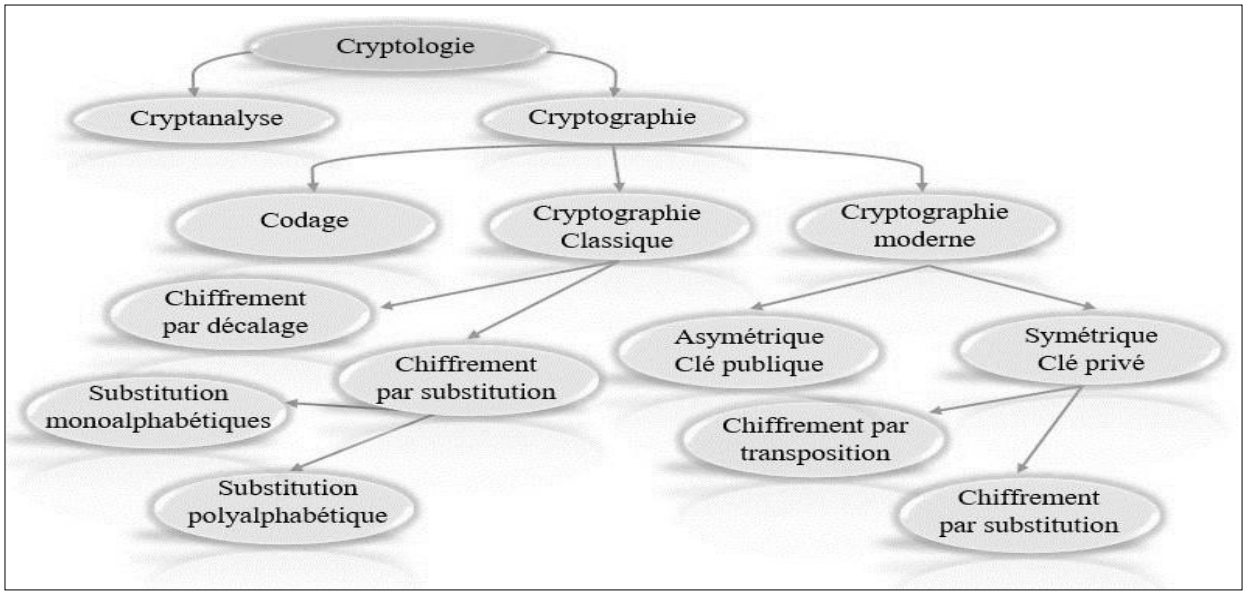


Figure 1.1: Les Domaines de Cryptographie.

La cryptologie a été utilisée initialement dans le but de protéger les communications sensibles. Elle était alors considérée davantage comme un art que comme une science ; ainsi les procédés des Spartiates ou de César assuraient la confidentialité. Ce n'est que plus tard qu'ont été définis d'autres besoins, comme l'authentification.

Le chemin si dessue présente la structure de l'histoire de la cryptologie :

- Vers1900av.J.-C, un scribe égyptien ployé des hiéroglyphes non conformes à la langue correcte dans une inscription.
- Quatre siècles plus tard, vers 1500 av .J.-C., une tablette mésopotamienne contient une formule chiffrée pour la fabrication de vernis pour les poteries.
- Cinq siècles avant notre ère, des scribes hébreux mettant par écrit le livre de Jérémie ont employé un simple chiffre de substitution connu sous le nom d'Atbash.
- En 487 av .J .-C . les grecs emploient un dispositif appelé la scytale, un bâton autour du quel une bande longue et min ce de cuir était enveloppée et sur la quelle on écrivait le message. Le cuir était ensuite porté comme une ceinture par le message [7]. Le destinataire avait un bâton identique permettant d'enrouler le cuir a fin de déchiffrer le message.

Citons quelque repère de la cryptographie moderne :

- 1975 conception du standard de chiffrement de données adopté 1977
- 1976 déifié et Hellman introduisant l'idée de système a clé publique
- 1978 inventions de RSA le premier système concret de cryptographie a clé publique.
- 1985 inventions du système cryptographie ElGamal
- 1991 adoptions du premier standard de signature basé sur l'algorithme discret.
- 2000 adoptions du Rijndael comme AES.

1.3 Vocabulaire de base

La cryptographie utilise des concepts issus de nombreux domaines (informatique, mathématique, électrique). Toutes fois, les techniques évoluent et trouvent aujourd'hui régulièrement racine dans d'autres branches (biologie, physique, etc.

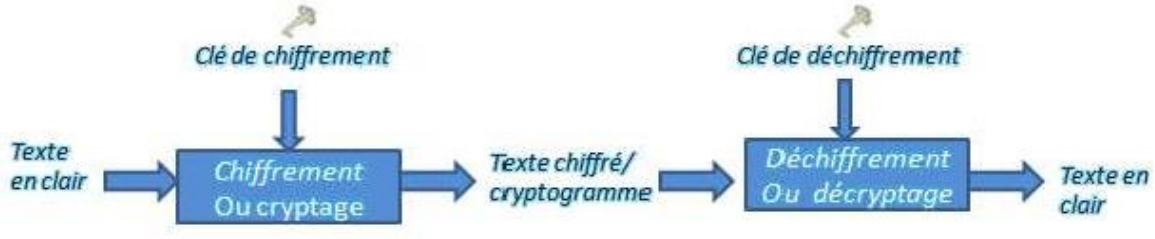


Figure 1.2:Principe algorithme de chiffrement

- **Cryptologie** : Il s'agit d'une science mathématique comportant deux branches : englobe à la fois la cryptographie et la cryptanalyse.

- **Cryptographie :** est l'étude des méthodes des principes et techniques mathématiques donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.
  Elle aussi définie comme étant " la science du secret '' [9].

- **Chiffrement:** Le chiffrement consiste à transformer une donnée (texte , message,...)afin de la rendre in compréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement.

- **Le déchiffrement :** Opération mathématique, effectuée à l'aide de la clé privée, qui consiste à décoder le message crypté à l'aide de la clé publique associée. Cette opération ne peut être effectuée que par le détenteur de la clé privée.

- **Texte chiffré :** Appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.

- **Cryptanalyse:** étudie la sécurité des procédés de chiffrement utilisés en cryptographie. Elle consiste alors à casser des fonctions cryptographiques existantes, c'est-à-dire à démontrer leur sécurité. La cryptanalyse mêle une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, depatience, de détermination et de chance.

- **Cryptosystème :** Il est définie comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.

- **Clef:** Il s 'agit du paramètre impliqué et autorisant des opérations de chiffrement/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle différente pour les deux opérations.

- **Le code :** ensemble de procédés et ensemble de symboles (lettres, nombres, signes, etc.) employés pour remplacer les mots du message à coder.

- **Le protocole :** un protocole est une série de pas, impliquant deux parties ou plus conçues pour accomplir une tâche. C'est une définition importante. "Une série de pas" signifie que le protocole a un ordre, du début à la fin. Chaque pas doit être exécuté à son tour et aucun pas ne peut être entamé avant que le pas précédent ne soit fini. "L'implication de deux parties ou plu signifie qu'on exige au moins que deux personnes 'achève le protocole ; une personne seule ne fait pas de protocole. Une personne seule peut exécuter une série de pas pour accomplir une tâche, mais ce n'est pas un protocole. Finalement, "conçu pour accomplir une tâche" signifie que le protocole doit réaliser quelque chose. Quelque chose qui ressemble à un protocole, mais qui n'accomplit aucune tâche n'est pas un protocole [10].

- **Le cryptographe :** a comme travail de fournir des outils pour éliminer les risques, afin de rendre les échanges d'information confidentiels, infalsifiables, authentique set inaltérables.

L'information est chaque jour échangée d'un point à un autre et se trouve susceptible d'être lue, copiée, supprimée, altérée ou falsifiée.

- **La stéganographie :** est une branche particulière de la cryptographie qui consiste non pas à rendre le message inintelligible, mais à le camoufler dans un support (texte, image, séquence vidéo,…etc.) de manière à masquer sa présence [11].
- **La signature :** partie chiffrée d'un message générée dans le but d'authentifier le message (intégrité) et l'expéditeur (identité).
- **Le certificat :** document informatique délivré sous la forme d'un fichier normalisé (X509) qui permet la signature, la vérification et le cryptage de messages. Le contenu du certificat lie les données, signées ou cryptées, au détenteur du certificat : l'utilisateur autorisé. Un certificat se compose d'une clé privée, d'une clé publique et d'une signature émanant de l'autorité de certification.
- **La clé privée :** partie du certificat, sous la responsabilité de l'utilisateur autorisé. Elle est destinée à la signature et au décryptage des messages. C'est la partie sensible de la clé qui doit rester secrète. La clé privée est utilisée pour déchiffrer les messages reçus et chiffrer la signature.[11]
- **La clé publique :** partie du certificat qui sera diffusée. Elle est destinée à la vérification de signature et au cryptage des messages. C'est la partie qui est transmise aux interlocuteurs ou placée sur des serveurs de clés. La clé publique est utilisée pour chiffrer les messages à envoyer et déchiffrer la signature des messages reçus [11].
- L'autorité de certification : l'organisme qui assure la délivrance et le renouvellement des certificats, la diffusion des listes de révocation et des clés publiques.
- Le condensé : le condensé est le résultat de la transformation numérique d'une information en une suite de caractères dont le contenu garantit l'intégrité du message transféré. Cette intégrité est vérifiée à la réception du message, lors de la vérification de signature, par comparaison du condensé reçu et du condensé calculé. Le caractère mathématiquement irréversible de la transformation garantit que le contenu du message n'a pas été falsifié.

1.4 Notations

Dans cette partie en va se mettre d'accord sur quelques notions comme suit :

- M : le texte clair, 11
- C : Le texte chiffré,
- E et D_k respectivement les clefs de chiffrement et de déchiffrement,

- E(x) : la fonction de chiffrement,
- D(x) : la fonction de déchiffrement.

Ainsi, en cryptographie la propriété de base est :

$$M=D(C) \Leftrightarrow C=E(M)$$

1.5 La cryptographie classique :

Elle décrit la période avant les ordinateurs. Elle traite des systèmes reposant sur les lettres et les caractères d'une langue naturelle (allemand, anglais, français, etc.). Les principaux outils utilisés remplacent des caractères par des autres les transposent dans des ordres différents. Les meilleurs systèmes de cette classe d'algorithmes répètent ces deux opérations de base plusieurs fois. Cela suppose que les procédures (de chiffrement ou déchiffrement) soient gardées secrètes ; et sans cela comme nous avons déjà dit le système est complètement inefficace (n'importe qui peut déchiffrer le message codé). On appelle généralement cette classe de méthodes : le chiffrement à usage restreint [11].

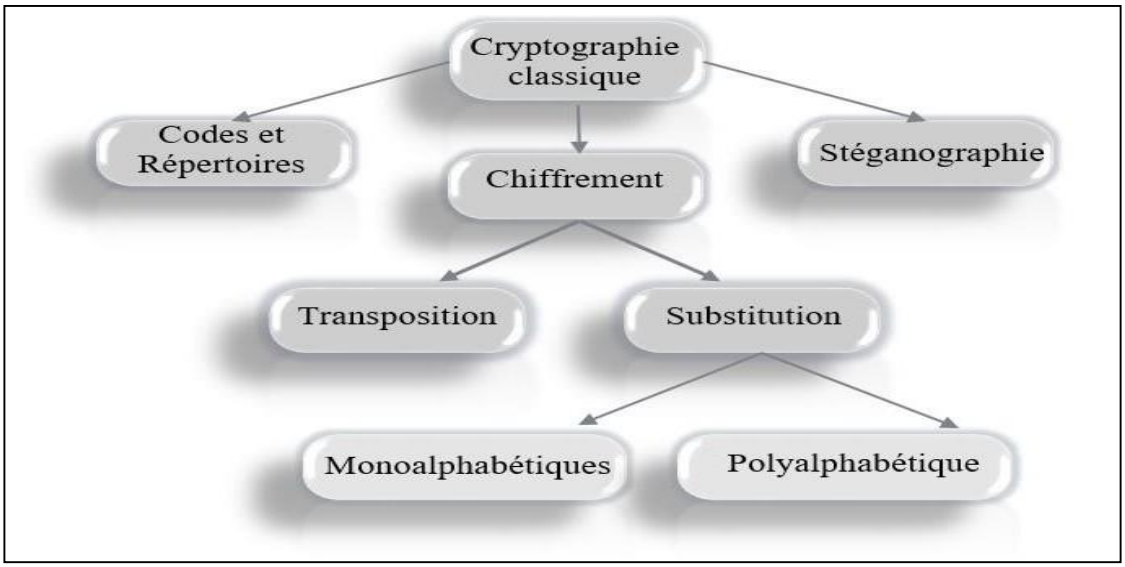


Figure 1.3: Domaines inclus dans la cryptologie.

1.6 Quelques algorithmes classiques

Dans cette partie en va citer quelque ancien algorithme utilise dans le passé lointain.

a- Les scytale des Spartiates

Un procédé de chiffrement avait été imaginé par les Spartiates, dans le souci de protéger la confidentialité de leurs informations. Le principe consistait à enrouler une lanière de cuir ou de papyrus autour d'un bâton de bois de diamètre fixé, puis à écrire le message en travers des spires, c'est-à-dire parallèlement à l'axe du bâton. Une fois déroulée, la lanière contenait donc un texte illisible, sauf pour le correspondant connaissant le diamètre adéquat.

Ce mécanisme porte le nom de permutation, puisqu'il consiste à mélanger les lettres du message, sans modifier ces lettres. Des historiens comme Thucydide ou Plutarque signalent l'utilisation de ce procédé au Vème siècle avant notre ère.

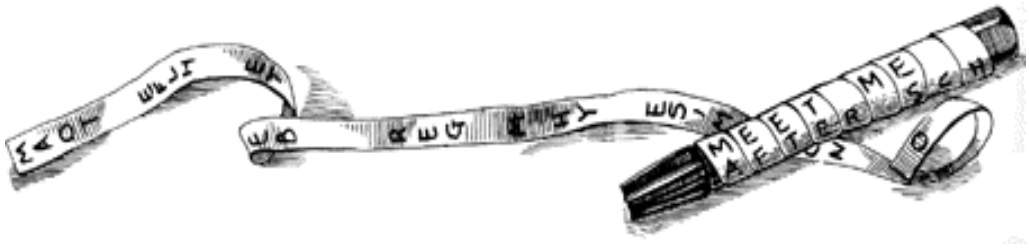


Figure 1.4: Le Scytale Spartiate

b- Chiffrement de César [12]

Le chiffrement de César est la méthode de cryptographie la plus ancienne communément mise par l'histoire. Il consiste en une substitution mono-alphabétique : chaque lettre est remplacée ("substitution") par une seule autre ("mono-alphabétique"), selon un certain décalage dans l'alphabet ou de façon arbitraire. D'après Suétone, César avait coutume d'utiliser un décalage de 3 lettres : A devient D, B devient E, C devient F, etc. Il écrivait donc son message normalement, puis remplaçait chaque lettre par celle qui lui correspondait Dans les formules ci-dessous, p est l'indice de la lettre de l'alphabet, k est le décalage.

- Pour le chiffrement, on aura la formule : $C = E(p) = (p + k) \bmod 26$
- Pour le déchiffrement, il viendra : $p = D(C) = (C - k) \bmod 26$

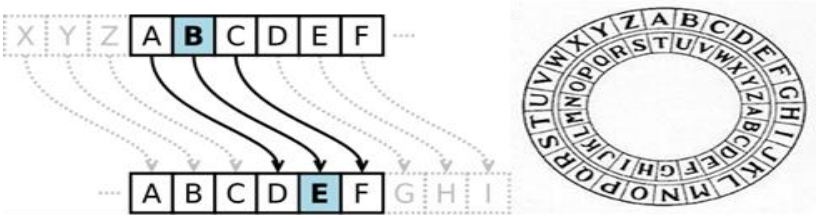


Figure 1.5: Chiffrement de César

c- Le chiffre de Hill [13]

Le chiffre publié en 1929 par Lester S. Hill (1891-1961) est un chiffre polygraphique), c'est-à-dire qu'on ne (dé) chiffre pas les lettres les unes après les autres, mais par paquets. Nous étudierons un seul cas qui est le cas de groupement biographique du chiffre de Hill, puisque nous grouperons les lettres deux par deux, mais on peut imaginer des paquets plus grands, par exemple des paquets de trois lettres. On a deux étapes qui sont les suivantes :

1- Chiffrement

Les lettres sont d'abord remplacées par leur rang dans l'alphabet. Les lettres et Pk+1 du texte clair seront chiffrées $C_k$et C_{k+1} avec la formule ci-dessous :

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} mod(26) \qquad (1.1)$$

Ce qui signifie, pour fixer les idées, que les deux premières lettres du message clair (*P1*et *P2*) seront chiffrées (C_1et C_2) selon les deux équations suivantes :

$$\begin{matrix} C_1 = aP_1 + bP_1(mod\ 26) \\ C_2 = cP_2 + dP_2(mod\ 26) \end{matrix} \quad (1.2)$$

2- Déchiffrement

Pour déchiffrer, le principe est le même que pour le chiffrement :

On prend les lettres deux par deux, puis on les multiplie par une matrice.

$$\begin{pmatrix} p1 \\ p2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} C1 \\ C2 \end{pmatrix} \quad (1.3)$$

Cette matrice doit être l'inverse de la matrice de chiffrement (modulo 26). Ordinairement l'inverse de la matrice est :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \quad (1.4)$$

d- Chiffrement de VIGENERE (1523-1596)

Le Chiffre de Vigenère est un système de chiffrement, élaboré par Blaise de Vigenère (1523-1596), diplomate français du XVIe siècle. C'est un système de substitution polyalphabétique. Cela signifie qu'il permet de remplacer une lettre par une autre qui n'est pas toujours la même, contrairement aux chiffres vu précédemment qui se contentaient d'utiliser la même lettre de substitution. C'est donc un système relativement plus « solide».L'outil indispensable du chiffrement de Vigenère est : « La table de Vigenère » ou « carré de Vigenère». On l'obtient en écrivant 26 fois l'alphabet, et en décalant chaque ligne d'une lettre. Pour la 1ère ligne :

ABCDE ...XYZ

Pour la 2eme :

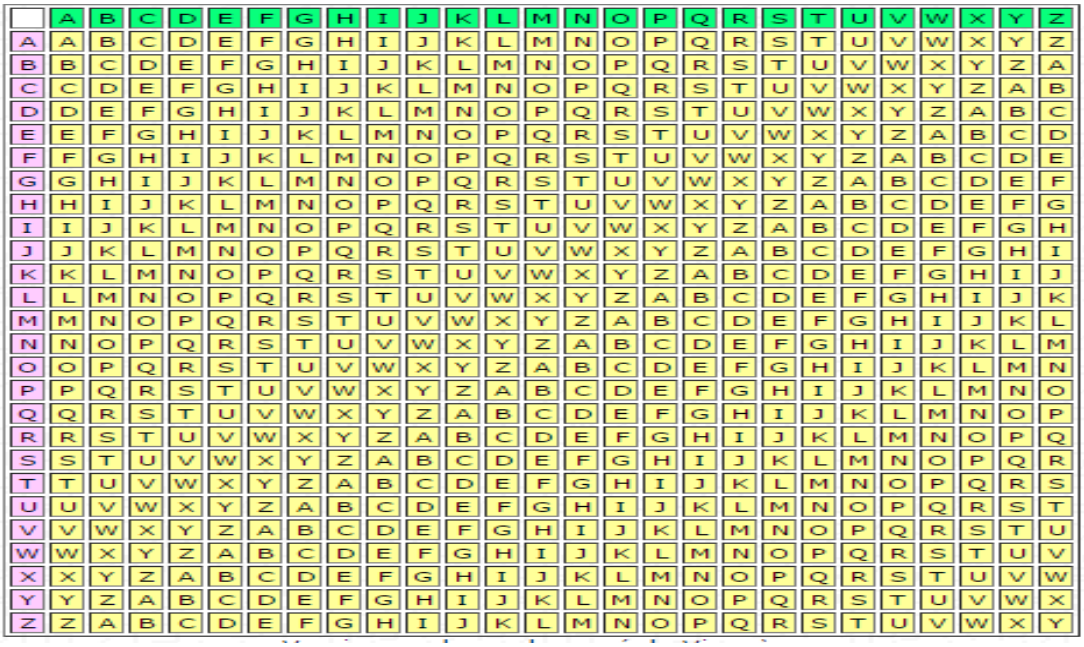
BCDEF... YZA La

3eme : CDEFG...ZAB

…Etc. Chiffrement

Pour chaque lettre en clair, on sélectionne la colonne correspondante et pour une lettre de la clé on sélectionne la ligne adéquate, puis au croisement de la ligne et de la colonne on trouve la lettre codée. La lettre de la clé est à prendre dans l'ordre dans laquelle elle se présente et on répète la clé en boucle autant que nécessaire.[14]

Le tableau de Vigenère est une matrice de 26 x 26 éléments



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

Figure 1.6: La table de Vigenère.

e- Chiffrement mécanisé (La machine ENIGMA)

Ces méthodes de chiffrement utilisent principalement des machines capables de chiffrer les messages clairs instantanément et se caractérisent par la rapidité d'exécution par rapport aux méthodes manuelles. L'histoire de la machine Enigma commence en1919, quand un ingénieur hollandais, Hugo Alexander Koch, dépose un brevet de machine à chiffrer électromécanique. Ses idées sont reprises par le Dr Arthur Scherbius, qui crée à Berlin une société destinée à fabriquer et à commercialiser une machine à crypter civile: l'Enigma.

Cette société fait un fiasco vu les coûts financiers de fabrication donc de commercialisation, mais la machine Enigma a attiré l'attention des militaires.

f- Le fonctionnement d'Enigma.

Le codage effectué par la machine Enigma est à la fois simple et astucieux. Chaque lettre est remplacée par une autre, l'astuce est que la substitution change d'une lettre à l'autre. La machine est alimentée par une pile électrique. Quand on appuie sur une touche du clavier, un circuit électrique est fermé, et une lampe s'allume qui indique quelle lettre codée l'on substitue.

Concrètement, le circuit électrique est constitué de plusieurs éléments en chaîne:

- Le tableau de connexions : il permet d'échanger des paires de l'alphabet, deux à deux, au moyen de fiches. Il y a 6 fiches qui permettent donc d'échanger 12 lettres. Un tableau de connexions est donc une permutation très particulière où on a échangé au plus 6 paires. Par exemple, dans le tableau suivant (avec simplement 6 lettres), on a échangé A et C, D et F, tandis que B et E restent invariants.
- Les rotors : un rotor est également une permutation, mais cette fois quelconque.
- À chaque lettre en entrée correspond une autre lettre.

On peut composer les rotors, c'est-à-dire les mettre les uns à la suite des autres. La machine Enigma disposera, au gré de ses évolutions successives, de 3 à 6 rotors. Parmi ces rotors, seuls 3 sont utilisés pour le codage, et on a le choix de les placer dans l'ordre que l'on souhaite (ce qui constituera une partie de la clé). Surtout, les rotors sont cylindriques, et ils peuvent tourner autour de leur axe. Ainsi, à chaque fois qu'on a tapé une lettre, le premier rotor tourne d'un cran, et la permutation qu'il engendre est changée. Observons ce changement sur la figure suivante : le rotor transforme initialement D en B. Lorsqu'il tourne d'un cran, cette liaison électrique D--->B se retrouve remontée en C

Chaque rotor possède donc 26 positions. A chaque fois qu'une lettre est tapée, le premier rotor tourne d'un cran. Après 26 lettres, il est revenu à sa position initiale, et le second rotor tourne alors d'un cran. On recommence à tourner le premier rotor, et ainsi de suite... Quand le second rotor a retrouvé sa position initiale, c'est le troisième rotor qui tourne d'un cran.

Le réflecteur : Au bout des 3 rotors se situe une dernière permutation qui permet de revenir en arrière. On permute une dernière fois les lettres 2 par 2, et on le fait retraverser les rotors, et le tableau de connexion.

Résumons sur la machine simplifiée suivante (6 lettres, 2 rotors) comment est codée la lettre A :

- On traverse le tableau de connexions : on obtient C
- On traverse les 2 rotors : on obtient successivement A et F
- On traverse le réflecteur où on obtient E, puis on renvoie dans les rotors pour obtenir F, A et finalement C après le tableau de connexions.

Remarquons que si avait tapé C, le courant aurait circulé dans l'autre sens et on aurait obtenu A.

Nombre de clés possibles. Il y a trois éléments à connaître pour pouvoir coder un message avec la machine Enigma:

- La position des 6 fiches du tableau de connexion: d'abord, il faut choisir 12 lettres parmi 26. C'est donc le nombre de combinaisons de 12 parmi 26, soit 26!/(12!14!). Maintenant, il faut choisir 6 paires de lettres parmi 12, soit 12 ! /6!, et comme la paire (A,D) donne la même connexion que la paire (B,A), il faut encore diviser par 26.On trouve enfin 100391 791 500.

- L'ordre des rotors : il y a autant d'ordres que de façons d'ordonner 3 éléments :3!=6.
- La position initiale des rotors : chaque rotor ayant 26 éléments, il y a 26*26*26=17 576 choix.

On multiplie tout cela, et on obtient plus de 1016 possibilités, ce qui est énorme pour l'époque !

Il est important de remarquer que les permutations employées dans les rotors et les réflecteurs ne peuvent pas être considérées comme faisant partie du secret. En effet, toutes les machines utilisent les mêmes, et il suffit donc d'en avoir une à disposition. Les Anglais, par exemple, en ont récupéré une pendant la guerre dans un sous-marin coulé. Ceci est une illustration d'un principe général en cryptographie, dit principe de Kerckhoff, qui veut que tout le secret doive résider dans la clé secrète de chiffrement et de déchiffrement, et pas dans une quelconque confidentialité de l'algorithme (ici de la machine) qui ne peut être raisonnablement garantie.

1.7 Principe de Kerckhoff

« La sécurité ne doit pas dépendre de ce qui ne peut pas change »

En d'autres termes, aucun secret ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré. Par contre, si on connaît K, le déchiffrement est immédiat on parle aussi de la Maxime de Shannon, dérivée du principe énoncé ci-dessus : L'adversaire connait le système.

Remarque : Il faut distinguer les termes "Secret" et "Robustesse" d'un algorithme. Le secret de l'algorithme revient à cacher les concepts de celui-ci, ainsi que les méthodes utilisées (fonctions mathématiques).

La robustesse quant à elle désigne la résistance de l'algorithme à diverses attaques qui seront explicitées dans la suite de ces notes [15].

1.8 La cryptographie moderne

Si le but de la cryptographie classique est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle, la cryptographie moderne s'intéresse en fait plus généralement aux problèmes de sécurité des communications [13]. Pour cela, on utilise un certain nombre de mécanismes basés sur des algorithmes cryptographiques [14].

La sécurité des données chiffrées dépend des éléments suivants [14]:

- La robustesse de l'algorithme cryptographique (difficile à casser).
- La confidentialité de la clé.

1.9 Critères de sécurités

La sécurité se base sur trois critères essentiels qui sont :

a- La confidentialité

Permet de protéger le contenu des informations sauvegardées ou transmises sur un réseau. Seules les personnes autorisées doivent pouvoir accéder aux informations ainsi protégées. Le *chiffrement de l'information* permet de résoudre le problème de la confidentialité : une personne souhaitant transmettre un message lui applique au préalable une fonction dite de chiffrement, et transmet le résultat au destinataire. Ce dernier retrouve le message original en utilisant une fonction de déchiffrement suivant le modèle de la cryptographie utilisée. Dans le modèle de la cryptographie à clé secrète les deux parties partagent la même clé de chiffrement et de déchiffrement, qui doit être gardée secrète. Les deux personnes jouent ainsi un rôle symétrique, tandis que, dans le modèle de la cryptographie à clé publique, le chiffrement est public et le déchiffrement est confidentiel. Pour envoyer un message chiffré, on applique une fonction de chiffrement utilisant la clé publique du destinataire. Ce dernier est le seul qui peut retrouver le message original à l'aide de sa clé privée. Les deux clés sont liées mathématiquement, mais il doit être impossible dans la pratique de retrouver la clé privée à partir de la clé publique (plus de précisions, ainsi que quelques exemples de méthodes sur les deux modes de chiffrement, symétrique et asymétrique).

b- L'intégrité

Garantir l'intégrité des données assure au récepteur que les données reçues sont celles qui ont été émises.

Le critère d'intégrité des ressources physique et logique et relatif au fait qu'elles n'ont pas été détruites ou modifiées à l'insu de leurs propriétaires tant de manière intentionnelle qu'accidentelle.

Une fonction de sécurité appliquée à une ressource pour contribuer à préserver son intégrité, permettra de la protéger plus au moins efficacement contre une menace de corruption ou de destruction.

Il convient de se protéger contre l'altération des données en ayant la certitude qu'elles n'ont pas été modifiées lors de leur stockage, de leur traitement ou de leur transfert.

Les critères de disponibilité et d'intégrité sont à satisfaire par des mesures appropriées afin de pouvoir atteindre un certain niveau de confiance dans les continus et le fonctionnement des infrastructures informatiques et télécoms [16].

c- **L'authentification**

Consiste à assurer l'identité d'un utilisateur, c'est à dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il doit être. On distingue deux types d'authentification.

1.10 Type de chiffrement dans La cryptographie moderne

Les techniques de cryptographie moderne se composent de grandes parties comme le montre la figure [17] [18] [19] :

- La cryptographie à clés secrètes ou cryptographie symétrique.
- La cryptographie à clés publiques ou cryptographie asymétrique

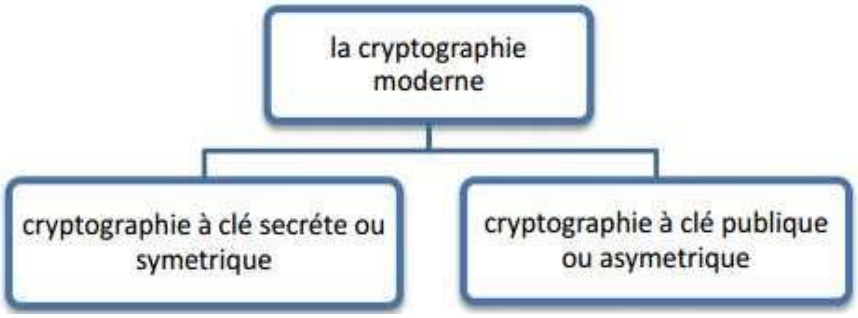


Figure 1.7:les méthodes de la cryptographie moderne [12].

1.10.1 Chiffrement symétriques (Cryptosystème à clef secrète)

La figure I.8 illustre le principe de ce type de chiffrement.

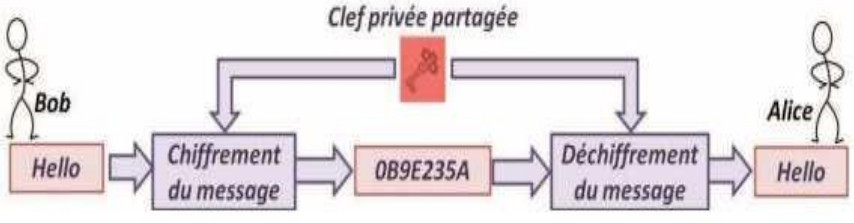


Figure 1.8: Principe de cryptographie symétrique

Principe

La cryptographie à clé secrète utilise la même clé pour les processus de chiffrement et de déchiffrement cette clé est le plus souvent appelée "secrète" (en opposition à "privée") car toute la sécurité de l'ensemble est directement liée au fait que cette clé n'est connue que par l'expéditeur et le destinataire.

On retrouve la cryptographie à clé secrète également sous les termes cryptographie asymétrique ou à clé privée. Les termes sont adéquats car c'est en effet la même clé qui est utilisée pour crypter le message par son auteur et le décrypter par son destinataire. C'est l'approche la plus authentique du chiffrement de données et mathématiquement la moins problématique

Caractérise par :

- Les clés sont identiques : $K_E = K_D = K$
- La clé doit rester secrète,
- Les algorithmes les plus répandus sont le DES, AES, RC4, 3DES,...
- Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés,
- Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texte clair en fonction de la clé,
- La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller jusque256.

a-Quelques algorithmes Chiffrement symétriques

- **DES (Data Encryption Standard)**

Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits, dont 8 bits (un octet) servent de test de parité (pour vérifier l'intégrité de la clé). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester un des octets de la clé par parité impaire, c'est-à-dire que chacun des bits de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet à qui il appartient. La clé possède donc une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée code produit.

La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés k1 à k16 Etant donné que « seuls » 56 bits servent effectivement à chiffrer, il peut exister 256 (soit 7.2*1016) clés différentes ! [20]

- **AES (Advanced Encryption Standard)**

L'algorithme prend en entrée un bloc de 128 bits (16 octets), la clé fait 128, 192 ou 256 bits. Les 16 octets en e4ntrée sont permutés selon une table définie au préalable. Ces octets sont ensuite placés dans une matrice de 4x4 éléments et ses lignes subissent une rotation vers la droite. L'incrément pour la rotation varie selon le numéro de la ligne.

Une transformation linéaire est ensuite appliquée sur la matrice, elle consiste en la multiplication binaire de chaque élément de la matrice avec des polynômes issus d'une matrice auxiliaire, cette multiplication est soumise à des règles spéciales selon GF(28) (groupe de Galois ou corps fini).

La transformation linéaire garantit une meilleure diffusion (propagation des bits dans la structure) sur plusieurs tours. Finalement, un XOR entre la matrice et une autre matrice permet d'obtenir une matrice intermédiaire. Ces différentes opérations sont répétées plusieurs fois et définissent un « tour ». Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours [21].

- **RC4**

RC4 a été conçu par Ronald Rivest (Le 'R' de RSA) en 1987. Officiellement nommé Rivest Cipher 4, l'acronyme RC est aussi surnommé Ron Code Il est utilisé dans des protocoles comme WEP, WPA ainsi que TLS. Les raisons de son succès sont liées à sa grande simplicité et à sa vitesse de chiffrement. Les implémentations matérielles ou logicielles étant faciles à mettre en œuvre.

1.10.2 Chiffrement asymétriques (Cryptosystème à clef secrète)

La figure (1.9) illustre le principe de ce type de chiffrement.

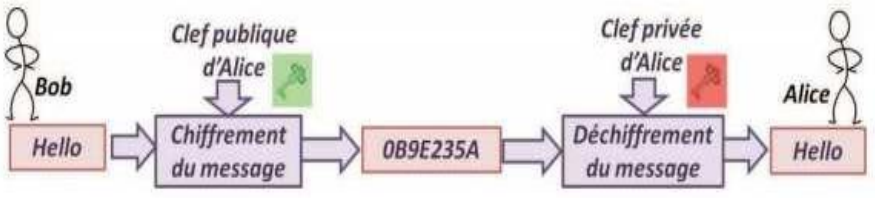


Figure 1.9: Principe de cryptographie asymétrique.

Principe

A l'ère de l'Internet, la cryptographie à clé publique s'est discrètement immiscée dans notre vie quotidienne. Elle permet notamment d'assurer la sécurité des cartes à puce ou du commerce électronique. La cryptographie à clef publique permet d'éliminer le problème de la distribution des clés. En effet, une clé publique diffusée sur un annuaire permet an n'importe qui de chiffrer un message, en revanche, il n'est pas possible de déchiffrer celui-ci par cette seule clef, il en faut une autre qui est gardée secrètement [22].

Dans un Cryptosystème asymétrique (aussi appelé Cryptosystème à clés publiques), les clés existent par paires (on parle souvent de bi-clés) :

Une clé publique pour le chiffrement

Une clé secrète pour le déchiffrement

Ainsi, dans un système de chiffrement à clé publique, les utilisateurs choisissent une clé aléatoire dont ils sont seuls connaisseurs (il s'agit de la clé privée).

A partir de cette clé, ils déduisent chacun automatiquement un algorithme (il s'agit de la clé publique). Les utilisateurs s'échangent cette clé publique au travers d'un canal non sécurisé. [16]

Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire (qu'il trouvera par exemple dans un serveur de clés tel qu'un annuaire). Ce dernier sera en mesure de déchiffrer le message à l'aide de sa clé privée (qu'il est seul à connaître).

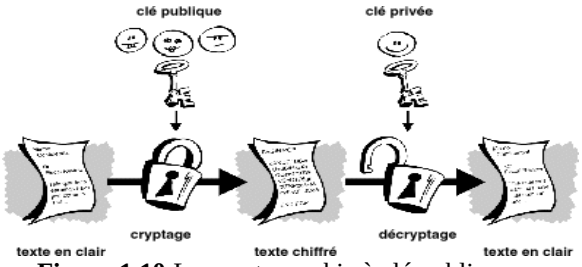


Figure 1.10: La cryptographie à clé publique

a-Quelques algorithmes Chiffrement asymétriques

Cryptage RSA : Rivest - Shamir – Adleman

Le premier système à clé publique solide à avoir été inventé, et le plus utilisé actuellement, est le système RSA. Publié en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman de l'Institut de technologie du Massachusetts(MIT), le RSA est fondé sur la difficulté de factoriser des grands nombres, et la fonction à sens unique utilisée est une fonction "puissance".

El Gamal [23]

ElGamal C'est un algorithme à clef publique présent à la base de la norme U.S. de signature électronique. Il fut inventé par Taher ElGamal en 1984. Il est basé sur la difficulté de calculer des logarithmes discrets. Le problème du logarithme discret consiste à retrouver un entier λ tel que

$$h = g^{\lambda} \bmod p \qquad (1.5)$$

1.11 Avantages et inconvénients

Dans le tableau ci-dessus en a exposé une comparaison entre la cryptographie symétrique et symétrique :

Tableau 1.1: Les avantages et les inconvénients de la cryptographie symétrique et asymétrique [24][25][26].

Type de cryptographie	Avantages	Inconvenient
Symétrique	-Système rapide de chiffrement /déchiffrement. -Clés relativement courtes (128 ou 256 bits). -Primitive de mécanismes cryptographiques, et Bonne performances et sécurité bien étudié. -Assure la confidentialité des données. -Pas de secrète à transmettre.	-Gestion des clés difficiles (nombreux clés). -Point faible : l'échange de la clé secrète. -Dans un réseau de N entités susceptibles de communiquer secrètement il faut distribuer N ∗(N-1) /2 clés.
Asymétrique	-Nombre clés à distribuer est réduit par rapport aux clés symétriques. -Très utile pour échanger des messages facilement. -La distribution est simplifiée : La clé privée n'est jamais révélée ou transmise et la clé publique est disponible à tous les utilisateurs.	-Les algorithmes à clé publique nécessitent une capacité de traitement importante, ce qui n'est pas raisonnable pour les systèmes à ressources limitées. -La relation clés publique/clés privée impose : -La taille de clés et relativement longue (généralement entre 512 et 2048 bits). -Gestion de certificats de clés publiques. -Lenteur de calcul. -Pas d'authentification de la source.

1.12 Fonction de hachage

Il s'agit de la troisième grande famille d'algorithmes utilisés en cryptographie. Le principe est qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieure celle de départ. Le message réduit portera le nom de "Haché" ou de "Condensé".

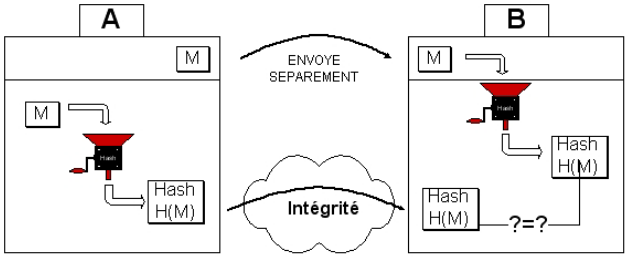


Figure 1.11: Principe de la fonction de hachage.

L'intérêt est d'utiliser ce condensé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque. Deux caractéristiques (théoriques) importantes sont les suivantes :

1. Ce sont des fonctions unidirectionnelles :

A partir de H(M) il est impossible de retrouver M.

2. Ce sont des fonctions sans collisions :

A partir de H(M) et M il est impossible de trouver M0 6= M tel que H(M0) = H(M).

1.13 Modes d'opération (chiffrement) [27]

Les modes sont des méthodes utilisés pour les chiffrements par blocs, on parle de modes opératoires. Dans le cadre d'une implémentation pratique, l'algorithme 'pur' est combiné à une série d'opérations simples en vue d'améliorer la sécurité sans pour autant pénaliser l'efficacité de l'algorithme .Cette combinaison est appelée un mode cryptographique.

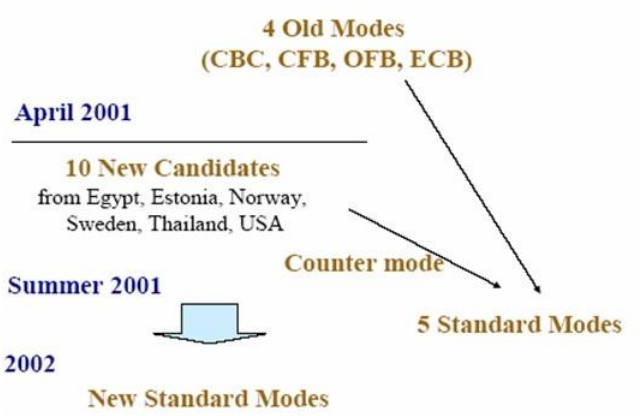


Figure 1.12: Apparition des modes cryptographiques.

L'utilisation de ces modes repose sur différents critères :

- **Sécurité :**
  - Effacement des formats standards (ex. l'introduction d'un texte).
  - Protection contre la modification de C.
  - Chiffrement de plusieurs messages avec la même clé.
- **Efficacité :**
- L'utilisation d'un mode cryptographique ne doit pas pénaliser l'efficacité du Cryptosystème.
- Limitation de la propagation des erreurs qui apparaissent dans M ou C.

a- Le mode ECB - Electronic Code Book (carnet de codage électronique)

Dans ce mode les blocs sont chiffrés indépendamment bloc par bloc. Formellement, il vient : $C_i = DES_K(P_i)$. Son usage est limité à la transmission sûre de valeurs isolées.

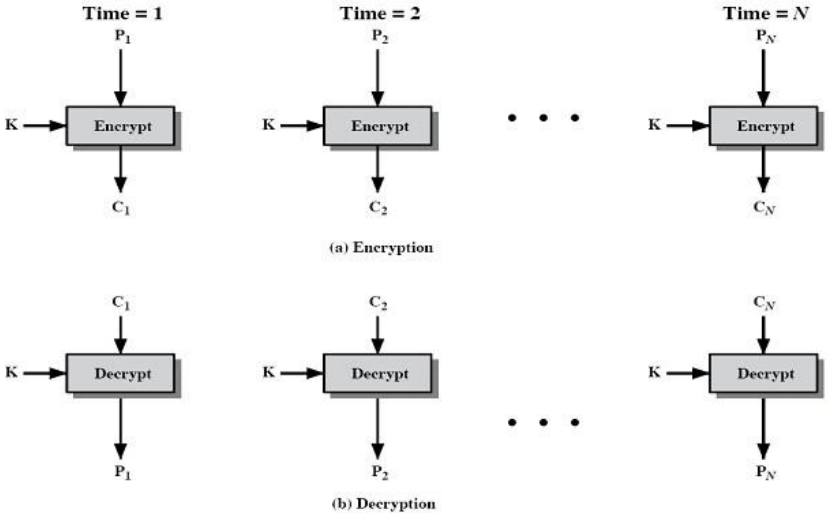


Figure 1.13: Diagramme du mode ECB.

b- *Le mode CBC - Cipher Block Chaining (chiffrement par chaînage de blocs)*

Dans ce mode, on applique sur chaque bloc d'ou exclusif avec le chiffrement du bloc précédent avant qu'il soit lui-même chiffré. De plus, afin de rendre chaque message unique, un vecteur d'initialisation est utilisé.

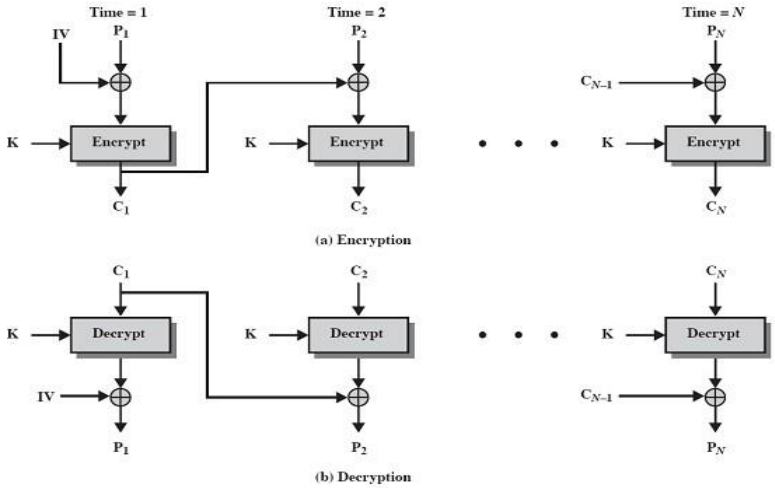


Figure 1.14: Diagramme de CBC.

c- Le mode CFB –Cipher Feed Back (chiffrement par rétroaction)

Le message est ajouté à la sortie du bloc chiffré. Le résultat sert de feedback pour l'étape suivante. Le registre peut utiliser un nombre quelconque de bits : 1, 8, 64bits (le plus souvent 64).

Il est utilisé pour le chiffrement par flux ainsi que pour l'authentification.

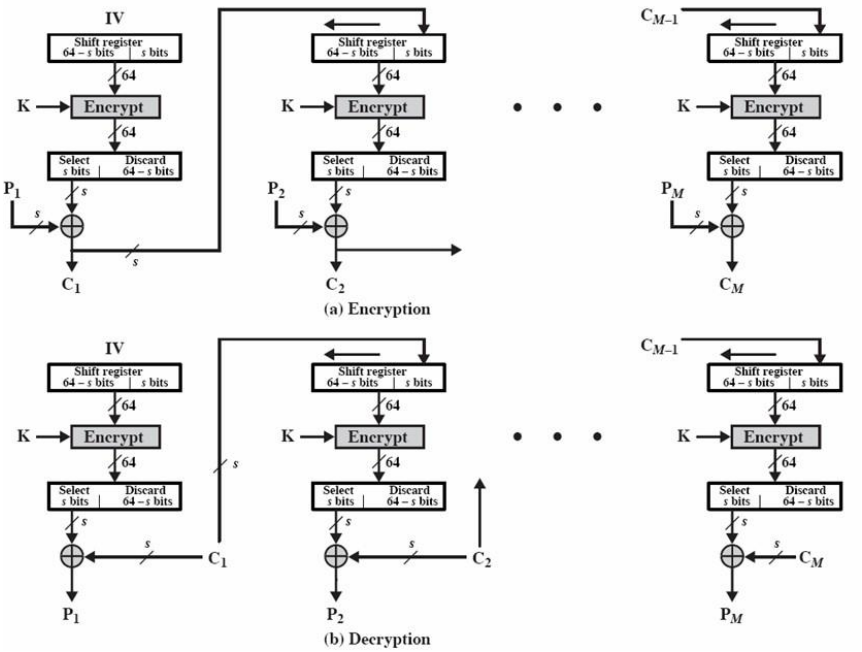


Figure 1.15: Diagramme de CFB.

d- Le mode OFB - Output FeedBack (Chiffrement de rétroaction de sortie)

Le feedback est indépendant du message. Tout le mécanisme est donc indépendant des blocs mi et ci. C'est une variante d'un chiffrement de Vernam avec réutilisation de la clé. Il est utilisé dans le cadre de chiffrement de flux sur un canal bruyant.

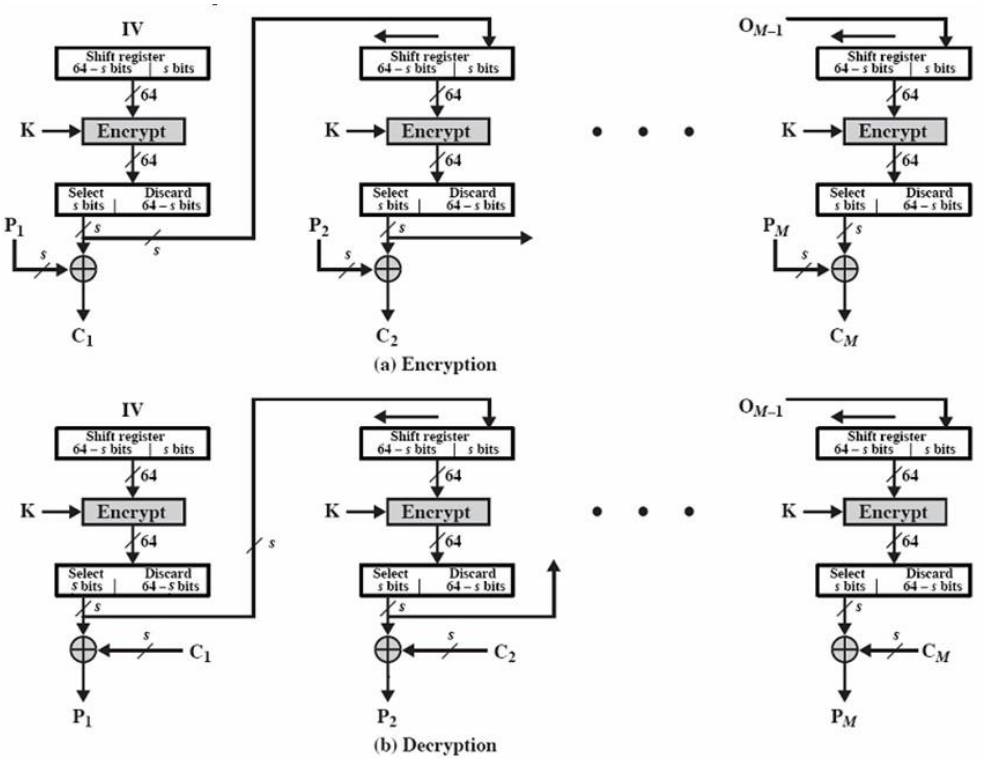


Figure 1.16: Diagramme d'OFB.

e- *CTR – CounTeR (Chiffrement basé sur un compteur)*

Ce mode est très rapide, ce qui le rend utile dans les réseaux grands vitesse. On y trouve un compteur en remplacement d'un IV et une clé différente pour chaque texte clair. Le compteur est une fonction simple mais garantissant que la séquence utilisée pour chiffrer ne sera pas réutilisée (cycle plus long que ne le nécessite le message).

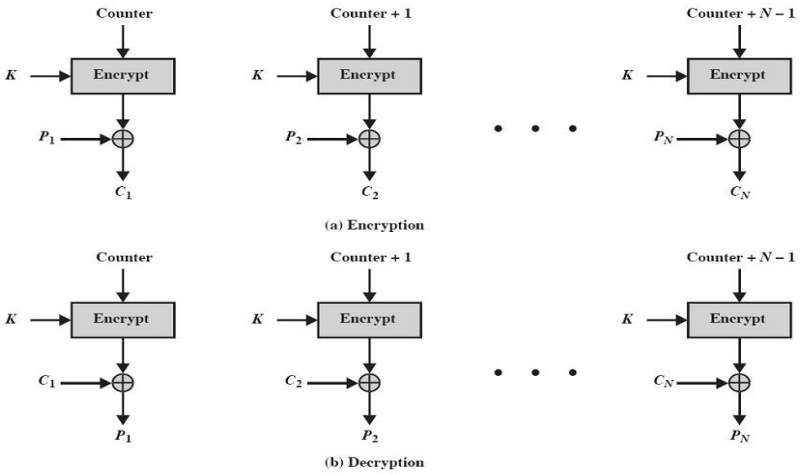


Figure 1.17: Diagramme de CTR.

1.14 Conclusion

L'origine de la cryptographie remonte sans doute aux origines de l'homme, dès que ceux-ci apprirent à communiquer. Alors, ils durent trouver des moyens d'assurer la confidentialité d'une partie de leurs communications.

Dans ce chapitre, nous avons présenté l'histoire de Cryptographie ainsi quelque algorithme classique.

Dans le chapitre suivent nous allons présenter la cryptographie moderne.

2.1 Introduction

Les systèmes de chiffrement font appel à des algorithmes de chiffrement souvent complexes qui modifient, à l'aide d'une clé de chiffrement plus ou moins longue, les caractères à protéger pour générer des données aléatoires. Ils se composent de deux principales classes : symétrique et asymétrique.

2.2 La cryptographie symétrique

La cryptographie symétrique, aussi appelée cryptographie à clé secrète, regroupe des mécanismes reposant sur la connaissance d'une clé secrète par deux personnes ou entités. Il s'agit de la branche la plus ancienne de la cryptographie. En effet, les systèmes cryptographiques historiques mettent en œuvre une même clé pour les opérations de chiffrement et de déchiffrement. La cryptographie à clé secrète remplit différentes fonctionnalités.

2.2.1 L'algorithme DES

Le D.E.S. (Data Encryptions Standard, c'est-à-dire Standard de Chiffrement de Données) est un standard mondial depuis la fin des années 1970.

Au début de cette décennie, le développement des communications entre ordinateurs a nécessité la mise en place d'un standard de chiffrement de données pour limiter la prolifération d'algorithmes différents ne pouvant pas communiquer entre eux. Pour résoudre ce problème, L'Agence Nationale de Sécurité américaine (N.S.A.) a lancé des appels d'offres. La société I.B.M. a développé alors un algorithme nommé Lucifer, relativement complexe et sophistiqué. Après quelques années de discussions et de modifications (applications de S-Boxes et réduction à des clés de 56 bits), cet algorithme, devenu alors D.E.S., fut adopté au niveau fédéral le 23 novembre 1976 **[28]**.

a. Algorithme de chiffrement

Le D.E.S. est un crypto système agissant par blocs. Cela signifie que D.E.S. ne chiffre pas les données à la volée quand les caractères arrivent, mais il découpe virtuellement le texte clair en blocs de 64 bits qu'il code séparément, puis qu'il concatène. Un bloc de 64 bits du texte clair entre par un coté de l'algorithme et un bloc de 64 bits de texte chiffré sort de l'autre côté. L'algorithme est assez simple puisqu'il ne combine en fait que des permutations et des substitutions.

C'est un algorithme de chiffrement à clef secrète. La clef sert donc à la fois à chiffrer et à déchiffrer le message. Cette clef a ici une longueur de 64 bits, c'est-à-dire 8 caractères, mais dont seulement 56 bits sont utilisés. On peut donc éventuellement imaginer un programme testant l'intégrité de la clef en exploitant ces bits inutilisés comme bits de contrôle de parité.

L'entière sécurité de l'algorithme repose sur les clefs puisque l'algorithme est parfaitement connu de tous. La clef de 64 bits est utilisée pour générer 16 autres clefs de 48 bits chacune qu'on utiliser alors de chacune des 16 itérations du D.E.S. Ces clefs sont les mêmes quel que soit le bloc qu'on code dans un message.

Cet algorithme est relativement facile à réaliser matériellement et certaines puces chiffrent jusqu'à 1 Go de données par seconde. Pour les industriels, c'est un point important notamment face à des algorithmes asymétriques, plus lents, tels que l'algorithme R.S.A**[28]**.

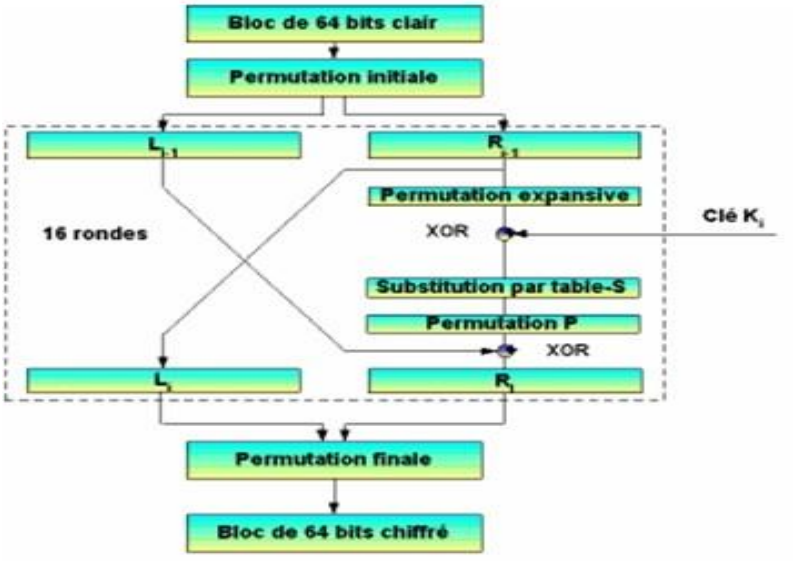


Figure2.1: Algorithme principal du DES

L'algorithme repose principalement sur 3 étapes, en plus de la gestion spécifique de la clé :

1. Permutation initiale.
2. Calcul médian (16 fois) : application d'un algorithme complexe appliqué en fonction de la clé.
3. Permutation finale.

a.1 La permutation initiale

Les 64 bits du bloc d'entrée subissent la permutation de la figure (2.2)

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Figure2.2: Matrice de permutations initiale

Cette "matrice" permet d'effectuer des changements internes au bloc (i.e. il n'y a pas d'apport de données extérieures). Le premier bit sera le bit 58, le second le bit 50, etc.

a.2 Le calcul médian

Les 64 bits initiaux de données sont divisés en 2 blocs (L et R).

Itérations :

– $Ln = Rn\text{-}1$

– $Rn = Ln\text{-}1 \oplus F(Rn\text{-}1, Kn)$

– $Kn = G(K, n)$

avec

– $Tn = LnRn$

– $Ln = t1...t32$

– $Rn = t33...t64$

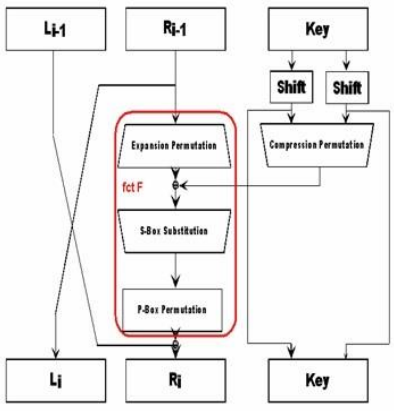


Figure 2.3 : Etape générale du calcul médian

Le calcul médian s'effectue en 16 itérations. Le détail de la fonction F est donné à la figure I.3 On traite 2 blocs simultanément : un bloc de 32 bits (données) et un bloc de 48 bits (clés). Le résultat forme un bloc de 32 bits.

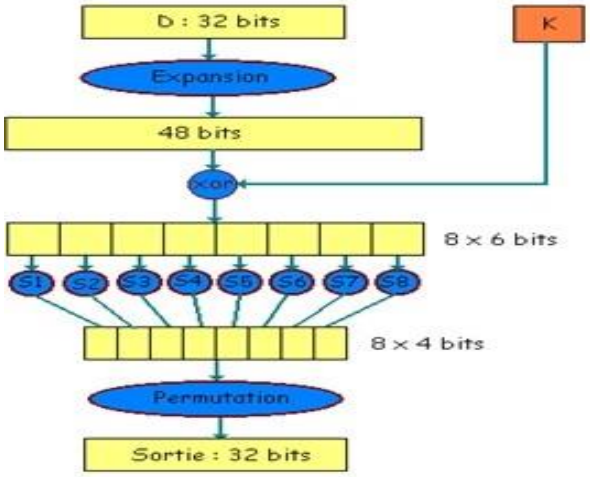


Figure 2.4: Fonction F détaillée

Expansion :

Les 32 bits sont étendus à 48 bits grâce à une table d'expansion (également appelée matrice d'extension). On retrouve ici un effet d'avalanche.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Figure2.5: Matrice d'expansion

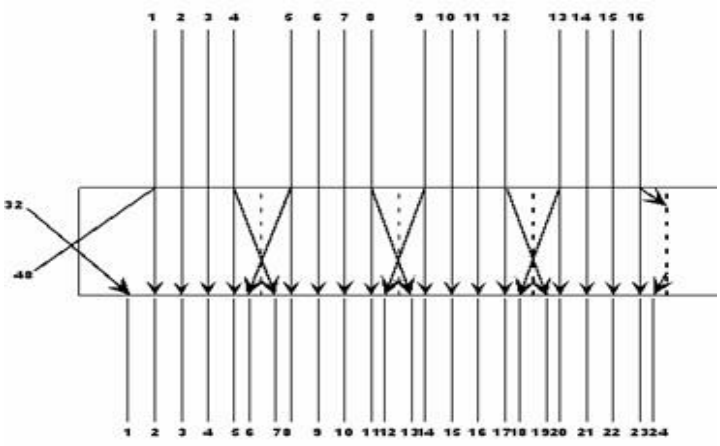


Figure 2.6: Phase d'expansion

Addition de la sous-clé :

Le résultat de l'expansion est additionné (par une opération $\oplus$) à la sous-clé *Kn* correspondant à l'itération selon la formule :

$$E(Ri\text{-}1) \oplus Ki = B1B2 \ldots B8$$

Les *B*1, *B*2, ..., *B*8 sont des blocs de 6 bits :

Bj= *b*1*b*2*b*3*b*4*b*5*b*6.

Transformations par S-Boxes :

Chaque bloc *Bj* constitue ensuite l'entrée de l'opération de substitution réalisée sur base des S-Box.

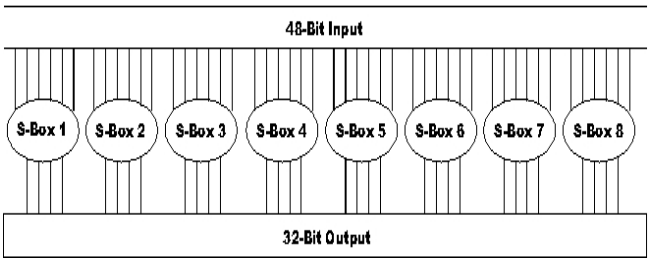


Figure 2.7: Transformations S-Box

L'opération de substitution consiste pour chaque S-box à calculer :

– *b*1 *b*6 = n° de ligne

– *b*2 *b*3 *b*4 *b*5 = n° de colonne

N° de colonne

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

N° de ligne

Figure2.8 : S-Box particulière

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
0	15	1	2	14	6	11	3	4	9	7	2	13	2	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	122	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Figure 2.9: Les 8 S-Box du DES

Transformations par P-Box (permutation du calcul médian) :

L'opération de permutation est réalisée sur le résultat de la substitution des S-box et est basée sur la table de la figure (2.10)

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Figure 2.10: Matrice de permutation du calcul médian

Le résultat de cette dernière permutation est noté *F*(*Rn*-1, *Kn*).

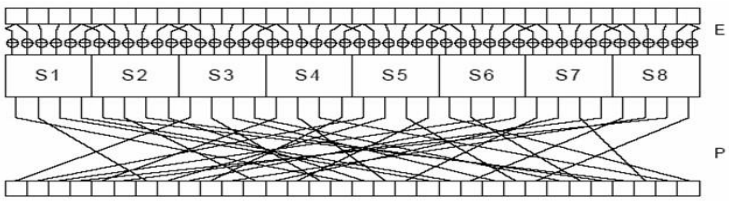


Figure 2.11: Ronde détaillée du calcul médian

a.3 Permutation finale

Une fois le calcul médian terminé, on pratique la permutation inverse de la permutation initiale. Attention toutefois : il s'agit de l'inverse de la permutation initiale, en d'autres termes, cette table permet de retrouver la position de départ. Ce n'est pas l'inverse de la "matrice" de départ !

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Figure 2.12: Permutation finale

b. Algorithme du calcul de la clé *G* (*K, n*)

La clé est constituée de 64 bits dont 56 sont utilisés dans l'algorithme. Les 8 autres peuvent être utilisés pour la détection d'erreurs où chacun de ces bits sera utilisé comme bit de parité des 7 groupes de 8 bits. Ainsi, le nombre total de clés est de 256 **[29]**.

La clé initiale est de 64 bits. Le calcul a lieu en 4 étapes :

1. Réduction à 56 bits : les bits de parité sont enlevés. On procède ensuite à une permutation semblable à celle de la figure (2.13)

57	**49**	**41**	**33**	**25**	**17**	**9**
1	**58**	**50**	**42**	**34**	**26**	**18**
10	**2**	**59**	**51**	**43**	**35**	**27**
19	**11**	**3**	**60**	**52**	**44**	**36**
63	**55**	**47**	**39**	**31**	**23**	**15**
7	**62**	**54**	**46**	**38**	**30**	**22**
14	**6**	**61**	**53**	**45**	**37**	**29**
21	**13**	**5**	**28**	**20**	**12**	**4**

Figure 2.13: Matrice de réduction de la clé

2. Division en sous-clés de 28 bits : le résultat de l'étape précédente (56 bits) est scindé en deux sous-clés de 28bits.
3. Rotation de la clé : à chaque itération, chaque sous-clé de 28 bits subit une rotation d'1 ou 2 bits vers la gauche selon la table de la figure (2.14)

Iteration i	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Figure 2.14: Rotation de la clé

4. Réduction : après concaténation des deux sous-clés précédentes, la clé résultante (56 bits) est réduite à une sous-clé de 48 bits sur base de la matrice de la figure (2.15)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Figure 2.15: Matrice de réduction de la clé

Le résultat de cette réduction est la sous-clé *Kn* additionnée avec *E*(*Rn*-1).

c. Déchiffrement

pour le déchiffrement il suffit d'appliquer le même algorithme mais inversé en tenant bien compte du fait que chaque itération du déchiffrement traite les mêmes paires de blocs utilisés dans le chiffrement. Il viendra que le déchiffrement suit cette équation :

$$R_{n-1} = L_n \; et \; L_{n-1} = R_n \oplus f(L_n, K_n)$$

2.2.2 L'algorithme AES

a. Les résultats d'un concours

La progression de la puissance des ordinateurs a causé la mort du DES. Ce dernier n'est plus jamais utilisé lorsque la sécurité demandée est forte (utilisation militaire, documents "secrets", etc.). Pour cette tâche, on préfère utiliser l'algorithme connu sous le nom générique d'AES (Advanced Encryption Standard), issu d'un concours créé en raison des faiblesses avérées du DES. Le véritable nom de l'AES est le Rijndael, nom résultant de la contraction des noms de ses inventeurs : Rijmen et Deamen **[30]**.

Le Triple DES demeure toutefois une norme acceptée pour les documents gouvernementaux aux U.S.A. Pour l'instant, il n'y a pas de projet ou d'obligation de déchiffrer les documents existants.

b. Cahier des charges

Au second tour du concours, les jurys devaient juger différents critères :

– La sécurité générale,

– Le coût en termes de calculs (rapidité),

–La simplicité de l'algorithme et ses facilités d'implémentation,

–Une lecture facile de l'algorithme, puisqu'il est destiné à être rendu public,
– La résistance aux attaques connues,
– Flexibilité - Portabilité : l'algorithme devant remplacer le DES, il est destiné à servir aussi bien dans les cartes à puces, aux processeurs 8 bits peu puissants, que dans des processeurs spécialisés pour chiffrer des milies de télécommunications à la volée.
– Techniquement, le chiffrement doit se faire par blocs de 128 bits, les clés comportant 128, 192 ou256 bits.
Au niveau du chiffrement/déchiffrement, les résultats varient assez fortement. Cependant, Serpent reste le moins bon pour la majorité des plateformes, Rijndael et RC6 étant les meilleurs.

c. Le choix : Rijndael

A la suite de nombreux tests, c'est finalement Rijndael qui a remporté la médaille, et est ainsi devenu le remplaçant officiel du DES.
Il possède les propriétés suivantes :
– Plusieurs longueurs de clef et de bloc sont possibles : 128, 192, ou 256 bits ;
– Le nombre de cycles ("rondes") varie en fonction de la longueur des blocs et des clés (de 10 à 14) ;
– La structure générale ne comprend qu'une série de transformations/permutations/sélections ;
– Il est beaucoup plus performant que le DES ;
– Il est facilement adaptable à des processeurs de 8 ou de 64 bits ;
– Le parallélisme peut être implémenté
À chaque ronde, quatre transformations sont appliquées :
1. Substitution d'octets dans le tableau d'état
2. Décalage de rangées dans le tableau d'état
3. Déplacement de colonnes dans le tableau d'état (sauf à la dernière ronde)
4. Addition d'une "clef de ronde" qui varie à chaque ronde

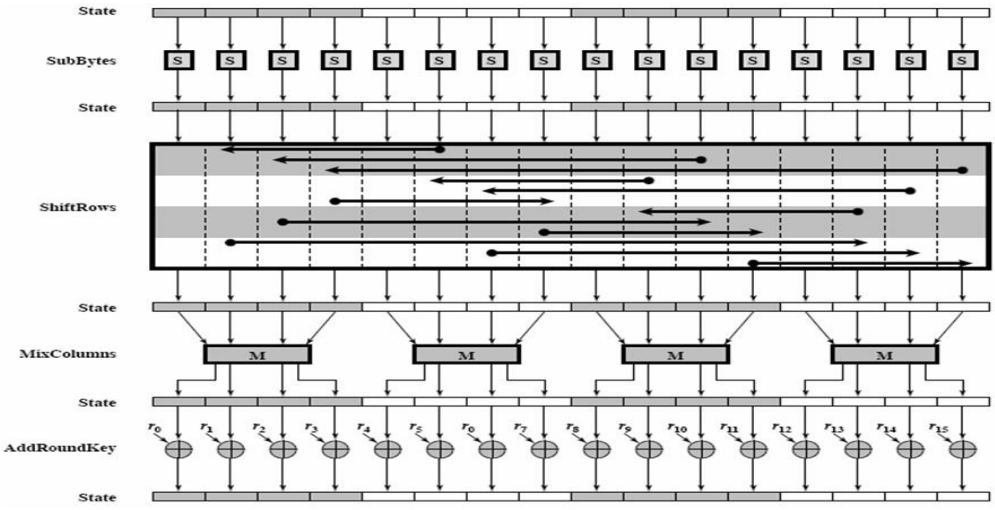


Figure2.16: Schéma général de Rijndael

d. Chiffrement et Déchiffrement

L'ordonnancement des étapes est illustré à la figure (2.17)

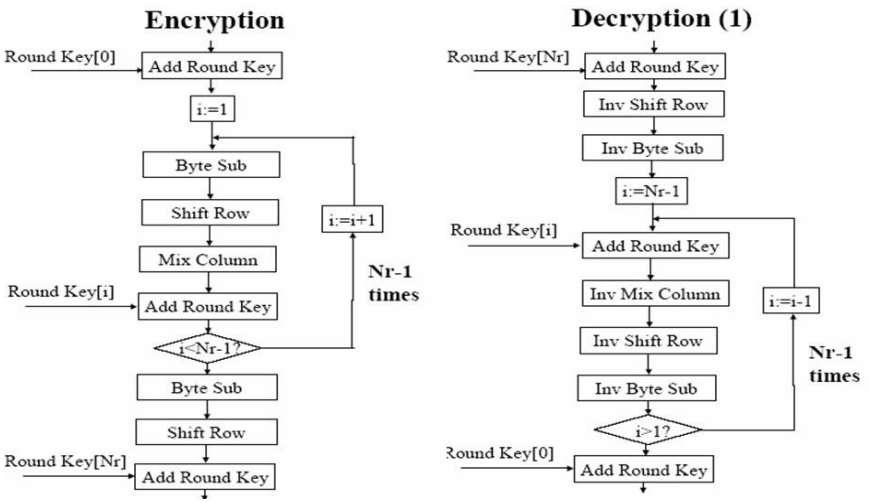


Figure2.17: Schéma des différentes étapes

e. Table d'état du texte et des clés

Le message et la clé sont conservés sous forme de tables représentées respectivement aux figures (2.19) et(2.20). Le nombre de colonnes dépend des tailles des textes et clés.

$N_b = L_{bloc}/32$

$N_k = L_{clef}/32$

Une colonne du tableau correspond à un mot de 32 bits. Ainsi, chaque petit bloc représente 8 bits, donc 1 octet. L'input et l'output sont donc gérés comme des séquences linéaires d'octets.

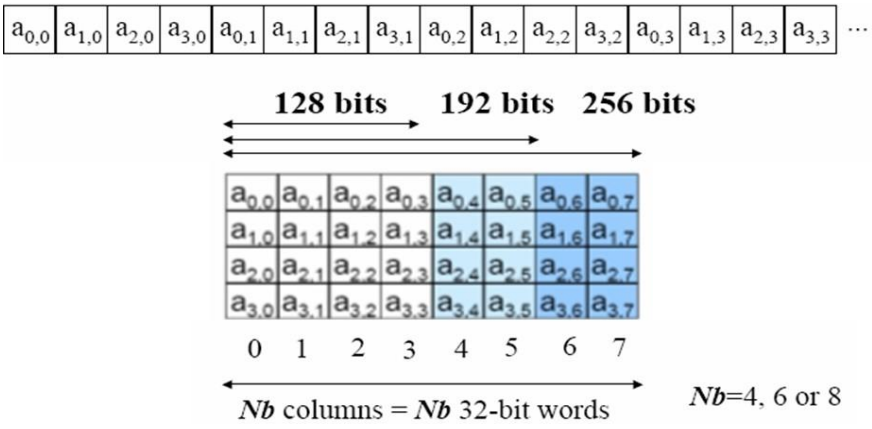


Figure 2.18: Table d'état du texte

f. SubByte

Les octets sont transformés en appliquant une S-Box inversible (afin de permettre un déchiffrement unique). Une seule S-Box est suffisante pour toute la phase de chiffrement.

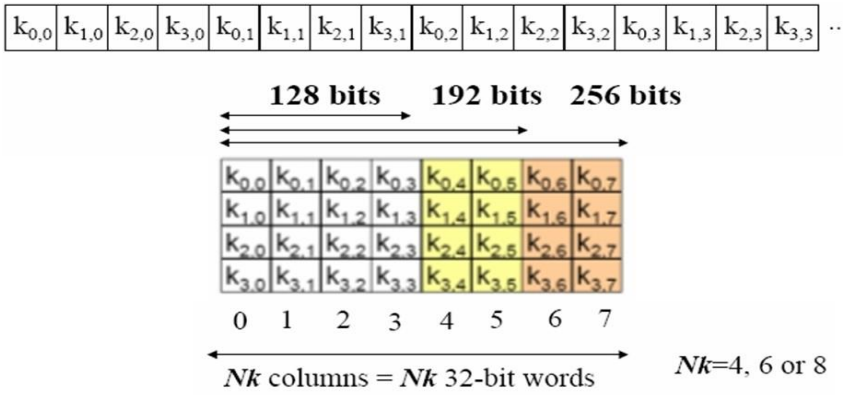


Figure 2.19: Table d'état des clés

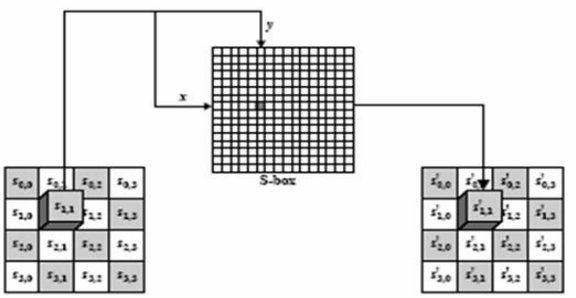


Figure 2.20: Table d'état des clés

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 2.21: S-Box inversible

g. ShiftRow

Cette étape augmente la diffusion dans la ronde.

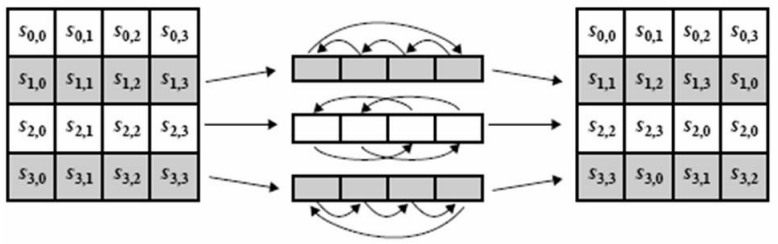


Figure 2.22: Schéma de l'étape ShiftRow

Selon la taille des blocs de message (c'est-à-dire la valeur de Nb), les décalages ne seront pas toujours identiques.

– La ligne 0 n'est jamais décalée,

– La ligne 1 est décalée de C_1,

– La ligne 2 est décalée de C_2,

– La ligne 3 est décalée de C_3.

	C_1	C_2	C_3
N_B=4	1	2	3
N_B=6	1	2	3
N_B=8	1	3	4

Figure 2.23:Décalage selon la taille des blocs de messages

h. MixColumn

Une différence sur 1 byte d'entrée se propage sur les 4 bytes de sortie. On a donc encore une étape de diffusion. La matrice utilisée est définie par Rijndael. Elle contiendra toujours ces valeurs.

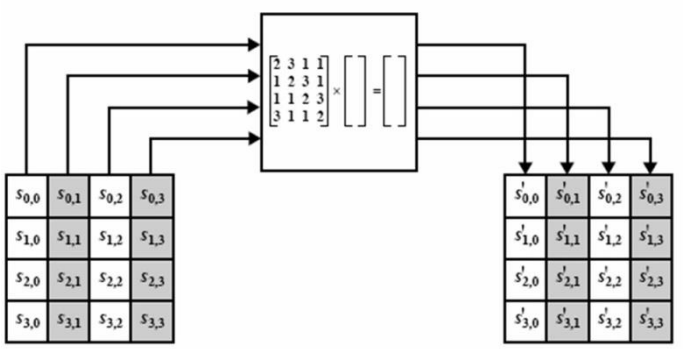


Figure 2.24: Etape du MixColumn

i. Add Round Key

C'est un simple ⊕des clés. Il s'agit d'additionner des sous-clés aux sous-blocs correspondants.

Figure 2.25: AddRound Key

j. Nombre de rondes

Selon la taille des blocs à traiter et la taille de la clé, le nombre de rondes évolue.

	Key length		
Block length	**128 bits Nk=4**	**192 bits Nk=6**	**256 bits Nk=8**
128 bits Nb=4	**10**	**12**	**14**
192 bits Nb=6	**12**	**12**	**14**
256 bits Nb=8	**14**	**14**	**14**

Figure 2.26: Nombres de rondes à effectuer

k. Calcul de la clé

Après avoir subi une extension (*Key Expansion*), la clé sera découpée en sous-clés (appelées clés de rondes), comme indiqué à la figure (2.27).

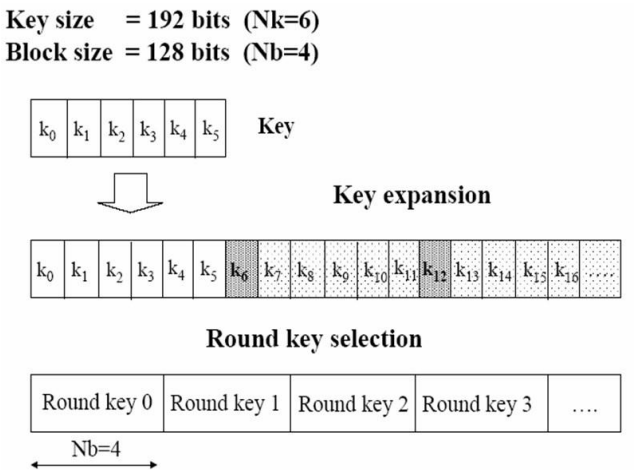


Figure 2.27: Schéma des opérations effectuées sur la clé

Le nombre de sous-blocs k_i dépendra bien sûr de la taille des clé et bloc du message.

l. Extension de la clé

Le calcul de l'expansion de la clé se fait de deux manières distinctes selon le sous-bloc de la clé concerné, comme l'illustrent les figures (2.27)et (2.28).

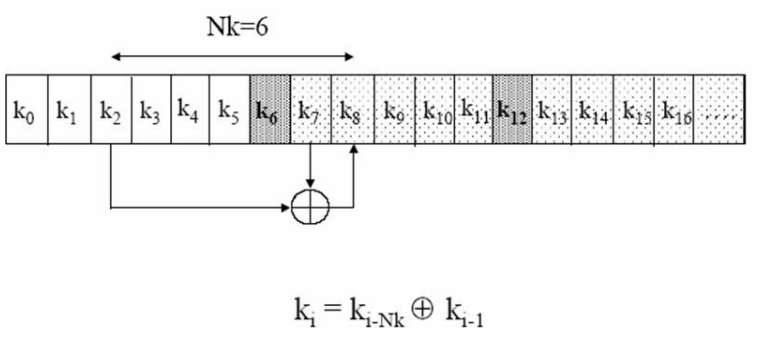


Figure 2.28: Expansion de la clé avec bloc "commun"

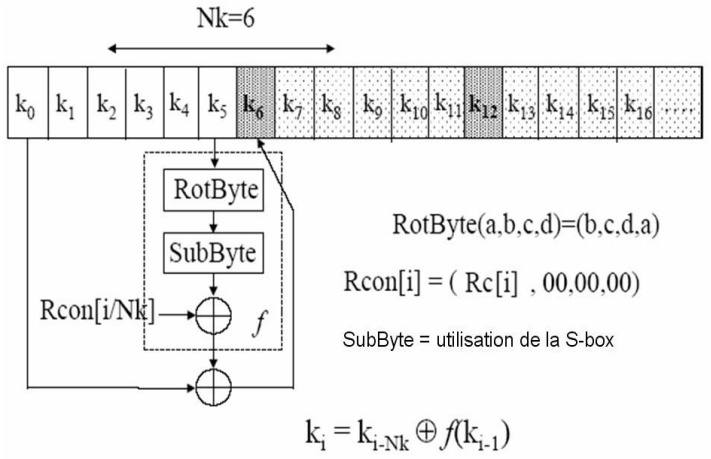


Figure 2.29: Expansion de la clé avec les blocs "multiples de Nk"

Remarques concernant la figure (2.28):

L'ajout de "Rcon[x]" donne comme résultat un $\oplus$sur les bits les plus significatifs. La table utilisée pour donner les valeurs de Rcon [] est donnée à la table (2.1).

Table2.1: Table de correspondance des Rcon []

j	1	2	3	4	5	6	7	8	9	…
RC[*j*]	01	02	04	08	10	20	40	80	1B	

La règle de construction3 de cette table est

$RC[1] = 1$

$RC[j] = 2.RC[j - 1]$

2.3 La cryptographie asymétrique

La cryptographie asymétrique repose sur une idée exposée par Diffie et Hellman en 1976dans [DH76] : le chiffrement et le déchiffrement sont deux opérations fonctionnellement différentes. Il n'y a donc aucune nécessité pour que les clés servant au chiffrement et au déchiffrement soient les mêmes. Dès lors que l'on utilise deux clés différentes pour le chiffrement et pour le déchiffrement et que la clé de déchiffrement ne peut pas être déduite de la clé de chiffrement, cette dernière peut être publique. De ce fait, la cryptographie asymétrique est également appelée cryptographie à clé publique. La clé de déchiffrement (ou clé privée) n'est connue que du destinataire. Ce dernier est l'unique détenteur de la clé privée.

2.3.1 L'algorithme RSA

2.3.1.1 Idée de base

Pour crypter un message on commence par le transformer en un –ou plusieurs– nombres. Les processus de chiffrement et déchiffrement font appel à plusieurs notions :

• On choisit deux nombres premiers p et q que l'on garde secrets et on pose n = p × q. Le principe étant que même connaissant n il est très difficile de retrouver p et q (qui sont des nombres ayant des centaines de chiffres)**[31].**

• La clé secrète et la clé publique se calculent à l'aide de l'algorithme d'Euclide et des coefficients de Bézout.

•Les calculs de cryptage se feront modulo n.

• Le déchiffrement fonctionne grâce à une variante du petit théorème de Fermat

Dans cette section, c'est Bruno qui veut envoyer un message secret à Alice. Le processus se décompose ainsi :

1. Aliceprépare une clé publique et une clé privée,

2.Bruno utilise la clé publique d'Alice pour crypter son message,

3. Alice reçoit le message crypté et le déchiffre grâce à sa clé privée.

a. Calcul de la clé publique et de la clé privée

Choix de deux nombres premiers Alice effectue, une fois pour toute, les opérations suivantes (en secret):

• elle choisit deux nombres premiers distincts p et q (dans la pratique ce sont de très grand nombres, jusqu'à des centaines de chiffres),

•Elle calcule $n = p*q$.

• Elle calcule $'(n) = (p - 1) \times (q - 1)$.

Choix d'un exposant et calcul de son inverse Alice continue:

•elle choisit un exposant e tel que pgcd $(e, '(n)) = 1$,

• elle calcule l'inverse d de e module $'(n)$: $d \times e \equiv 1 \pmod{'(n)}$. Ce calcul se fait par l'algorithme d'Euclide

étendu.

b. Clé publique

La clé publique d'Alice est constituée des deux nombres : (e,n)

Et comme son nom l'indique Alice communique sa clé publique au monde entier.

c. Clé privée

Alice garde pour elle sa clé privée: (d, n)

Alice détruit en secret p, q et $'(n)$ qui ne sont plus utiles. Elle conserve secrètement sa clé privée.

2.3.1.2 Principe de chiffrement

Bruno veut envoyer un message secret à Alice. Il se débrouille pour que son message soit un entier (quitte à découper son texte en bloc et à transformer chaque bloc en un entier).

Message chiffré

Bruno récupère la clé publique d'Alice : n et e avec laquelle il calcule, à l'aide de l'algorithme d'exponentiation rapide,

le message chiffré :

$$C = M^e \bmod n \quad (2.1)$$

Il transmet ce message Cà Alice.

2.3.1.3 Principe de déchiffrement

Alice reçoit le message *C* chiffré par Bruno, elle le décrypte à l'aide de sa clé privée *d*, par l'opération :

$$M = C^d mod n \quad (2.2)$$

2.3.2 Les courbes elliptiques

2.3.2.1 Idée de base

Il s'agit d'un concept proposé en 1985 par deux chercheurs Miller et Klobitz, de façon totalement indépendante. Ce type de cryptographie, toujours basé sur le modèle asymétrique permet aussi bien de chiffrer que de signer. On utilise souvent l'abréviation ECC, pour Elliptic Curve Cryptographie. Les clés utilisées sont plus courtes pour une sécurité égale ou supérieure. La théorie sous-jacente, ainsi que l'implémentation sont plus complexes, ce qui explique le fait que cette technologie soit moins répandue.

Toutefois, de par la nécessité de traiter plus rapidement l'information, de gérer des quantités de données importantes et de miniaturiser au maximum, les avantages de cette technique poussent la recherche.

D'une manière générale, sur R, les courbes elliptiques seront considérées comme l'ensemble des couples(x, y) tels que :

$y^2 = x^3 + ax + b$

Dont le discriminant

$-(4a3 + 27b2)$

Est non nul

Pour la dessiner, pour *a* et *b* fixés, on calcule *y* tel que :

$y = px3 + ax + b$

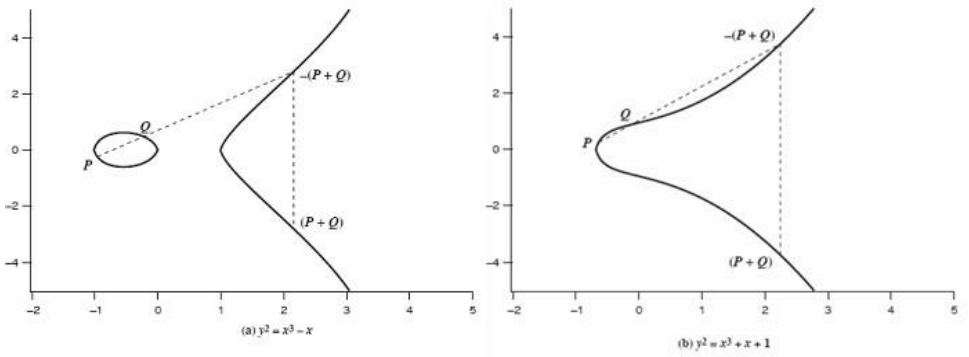


Figure 2.30: Deux exemples d'EC

a. Définition géométrique

Soit une opération (l'addition, +) pour l'ensemble *E*(*a, b*) tel que *a* et *b* répondent à la condition du discriminant.

Si 3 points sur une EC sont alignés, leur somme vaut *O* (point à l'infini).

1. *O* est l'identité pour l'addition : *O* = -*O*.

2. Pour n'importe quel point *P* + *O* = *P*.

3. L'opposé d'un point *P*(*x, y*) est *P*(*x, -y*)

4. Pour additionner 2 points *P* et *Q*, on trace la droite les reliant. Cela nous donne un point d'intersection *R*. On définit l'addition telle que *P* + *Q* = -*R*. En conséquence, on définit *P* + *Q* comme étant l'opposé de ce point *R*.

b. Les EC sur *Zp*

Les variables et coefficients prennent des valeurs dans l'ensemble [0, *p* - 1] pour un certain nombre premier *p*, et où toutes les opérations sont calculées modulo *p*. L'équation devient :

$y^2 \bmod p = (x^3 + ax + b) \bmod p$ *(2.3)*

Cette équation est par exemple satisfaite pour a = 1, b = 1, x = 9, y = 7 et p = 23.

72 mod 23 = (93 + 9 + 1) mod 23

49 mod 23 = 739 mod 23

3 = 3

On note Ep (a, b) l'ensemble des couples d'entiers (x, y) qui satisfont cette équation. On parle de groupe elliptique.

c. La cryptographie sur courbes elliptiques (ECC)

Pour utiliser les courbes elliptiques en cryptographie, il faut trouver un problème difficile (tel que la factorisation d'un produit en ses facteurs premiers dans le cas du RSA).

Considérons l'équation

$$Q = kP$$

où $Q, P \in Ep(a, b)$ et $k < p$.

Il est facile de calculer *Q* connaissant *k* et *P*, mais il est difficile de déterminer *k* si on connait *Q* et *P*. Il s'agit du problème du logarithme discret pour les courbes elliptiques : *logP*(*Q*).

Dans une utilisation réelle, le *k* est très grand, rendant l'attaque par force brute inutilisable (rappelons qu'a priori, l'attaque par force brute est toujours possible. . .).

d. ECC pour l'échange de clés

Soit un grand entier premier *q* (en considérant que l'on va utiliser les équations présentées précédemment, et non les équations dans *GF* (2*m*)) et les paramètres *a* et *b* satisfaisant l'équation *y*2 mod*q*=(*x*3 + *ax*+ *b*) mod*q*. Cela nous permet de définir *Eq*(*a, b*).

Prenons ensuite un point de départ *G*(*x*1, *y*1) dans *Eq*(*a, b*) dont l'ordre *n* est élevé. L'ordre *n* d'un point sur une EC est le plus petit entier positif tel que *nG*= *O*.

Eq (*a, b*) et *G* sont rendu publiques.

L'échange d'une clé par ECC entre deux entités A et B se déroule comme suit :

– A choisit un *nA* inférieur à *n* qui sera sa clé privée.

A génère alors sa clé publique $PA = nA \times G$.

– B choisit un *nB* inférieur à *n* qui sera sa clé privée.

B génère alors sa clé publique $PB = nB \times G$.

– A génère la clé secrète $K = nA \times PB$ et B génère la clé secrète $K = nB \times PA$.

Exemple (Schaefer, Santa Clara University) :

– Soient $p = 211$, Ep $(0, -4)(\Rightarrow y2 = x3 - 4)$ et $G = (2, 2)$. On calcule que $240G = O$ et donc $n = 240$.

– A choisit $nA = 121$, ce qui lui donne $PA = 121(2, 2) = (115, 48)$.

– B choisit $nB = 203$, ce qui lui donne $PB = 203(2, 2) = (130, 203)$.

– La clé secrète *K* générée est $121(130, 203) = 203(115, 48) = (161, 69)$.

2.3.2.2 Principe de chiffrement

Même si la cryptographie par courbes elliptiques est souvent employée pour l'échange d'une clé symétrique, elle est aussi utilisée pour chiffrer directement les données. Voici un exemple de Cryptosystème les utilisant.

Il faudra ici encoder le texte clair *m* comme un point *Pm* de coordonnées *x* et *y*. C'est ce point qui sera chiffré. Il faut ici aussi rendre publique un point *G* et un groupe elliptique *Eq* (*a, b*). Les utilisateurs doivent également choisir une clé privée et générer la clé publique correspondante. Pour chiffrer le message, A déterminé aléatoirement un nombre entier positif *k* et produit *Cm* comme un couple de points tel que :

$$Cm = \{kG, Pm + kPB\}$$

On remarquera l'utilisation de la clé publique de B.

2.3.2.3 Principe de déchiffrement

Pour déchiffrer, B devra multiplier le premier point par sa clé privée, et soustraire le résultat au second point reçu :

$$Pm + kPB - nB\,(kG) = Pm + k(nBG) - nB(kG) = Pm$$

2.4 Conclusion

Il n'y a pas de sécurité absolue, mais des éléments de sécurité relatifs. La recherche d'algorithmes, Protocoles et architectures de sécurité doit donc viser à faire en sorte qu'il ne soit pas rentable par rapport à l'information convoitée.

On constate donc qu'il y a nécessairement des solutions différentes selon la valeur de l'information à transmettre et le choix de l'algorithme est déterminant.

Pour la suite du travail nous irons vers la simulation de l'algorithme AES et son interface graphique.

3.1 Introduction

Dans ce chapitre, nous présentons la création d'une interface graphique en utilisant les GUIs de Matlab, pour le chiffrement des images en utilisent l'algorithme AES. Ce choix et pratique pour voir en mieux le chiffrement des images. Plusieurs modes de chiffrement sont utilisés pour renforce la sécurité tel que Electronic Code Book (ECB), Cipher Bloc Chaining (CBC), CipherFeedBack (CFB), Output FeedBack (OFB), Counter-mode encryption (CTR)

Des métrise sont utilisées pour évoluer la qualité des images après opération de déchiffrement

3.2 Interfaces graphiques

En informatique, une interface graphique (en anglais GUI pour graphical user interface) est un dispositif de dialogue homme-machine, dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran, de sorte que l'usager peut utiliser en imitant la manipulation physique de ces objets avec un dispositif de pointage, le plus souvent une souris.

3.3 Objets graphiques

3.3.1 Objet figure

Les objets Figure dans la création d'une interface graphique sont les conteneurs visibles où sont disposés tous les autres objets enfants. Ces objets sont couramment appelés « fenêtres ». Plusieurs objets Figure peuvent être ouverts simultanément et peuvent éventuellement communiquer entre eux [32].

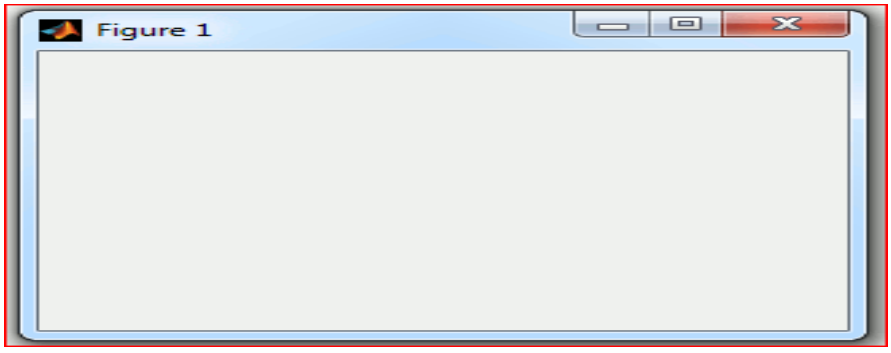


Figure 3.1: Figure 1 sur Matlab.

3.3.2 Objets Axes

Les objets Axes sont les zones de traçage des graphiques (2D ou 3D). Un objet Figure peut contenir plusieurs objets Axes simultanément.

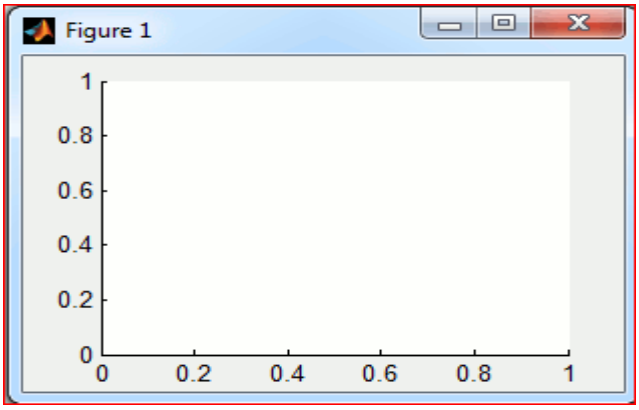


Figure 3.2: Objet axes.

3.3.3 Objets GUI

Au même niveau hiérarchique que les objets Axes, on trouve les objets GUI (pour User Interface). Certains de ces objets (comme les boutons, les menus, les cases à cocher) permettent à l'utilisateur d'interagir avec l'interface graphique grâce à la souris ou au clavier. D'autres objets (comme les panels, les tables…) servent à la mise en forme de l'interface graphique**[32].**

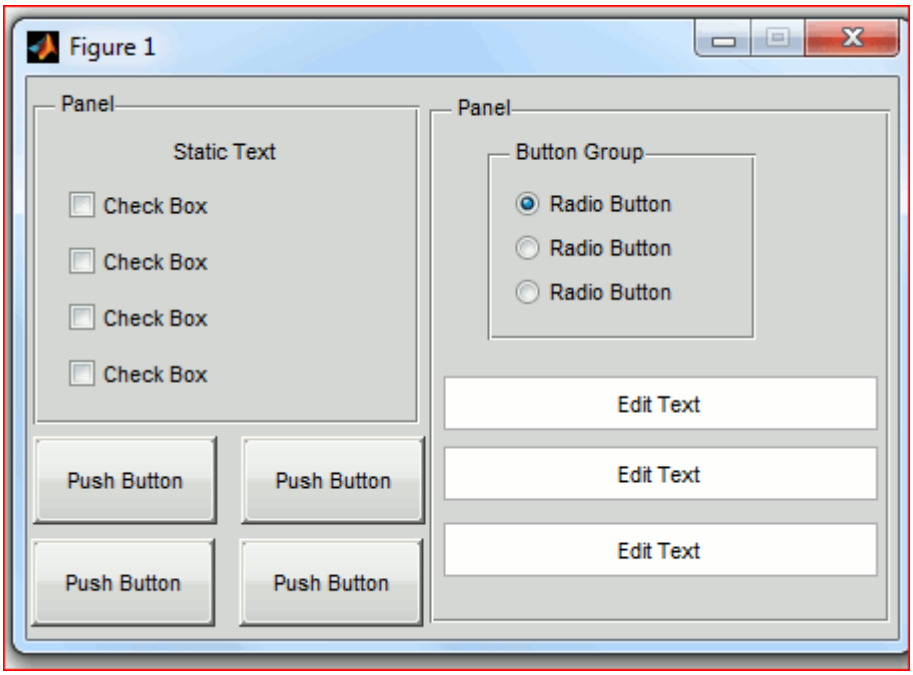


Figure 3.3: Exemple d'une interface graphique

3.4 L'algorithme AES sur des images

3.4.1 Algorithmes AES :

L'algorithme AES est un algorithme de chiffrement par blocs symétrique, qui fonctionne sur un groupe de bits de longueur fixe, appelés blocs. Il prend un bloc d'entrée de 128 bits et produit un bloc de sortie correspondant de la même taille.

L'AES nécessite une deuxième entrée, qui est la clé secrète. L'AES utilise trois tailles de clés différentes :128, 192 et 256 bits. Pour son chiffrement et son chiffrement inverse, l'algorithme AES utilise une fonction ronde composée de quatre couches orientées octets différentes :

1. Substitution d'octets à l'aide d'une table de substitution (boîte S)
2. Décalage des lignes du tableau State par différents décalages.
3. Mélanger les données dans chaque colonne du tableau d'état.
4. Ajouter une clé ronde à l'état. L'algorithme AES utilise cette fonction de round pour dix rounds, mais le premier et le dernier round de l'algorithme AES diffère des autres rounds**[33].**

Cela est illustré par la figure (**3.4**).

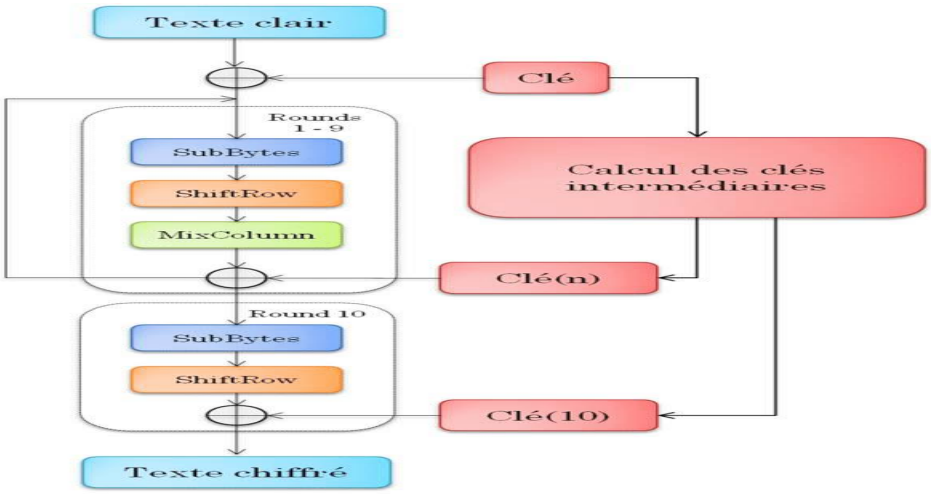


Figure 3.4: Algorithmes AES

3.4.2 Les modes de chiffrement utilisés :

Que ce soit pour DES ou des crypto-systèmes symétriques plus récents comme IDEA ou AES ou pour des crypto-systèmes asymétriques comme RSA ou El Gamal les clés sont de longueur fixée. Les messages eux peuvent avoir une longueur arbitraire. Pour adapter la taille du message à celle de la clef on décompose le message par blocs de taille fixe correspondant aux tailles des clés que l'on chiffre ensuite un a un et que l'on envoie successivement. Pour cela quatre modes de chiffrement par blocs sont possibles : ECB, CBC, CFB et OFB **[34].**

3.4.2.1 Le mode ECB :

Le mode ECB, Electronic Code Book, est le mode le plus simple **[34]**. Le message, M, est découpé en blocs, (M_i) avec i≥1, et chaque bloc est crypté séparément par *:*

$$C_i = E(M_i)$$

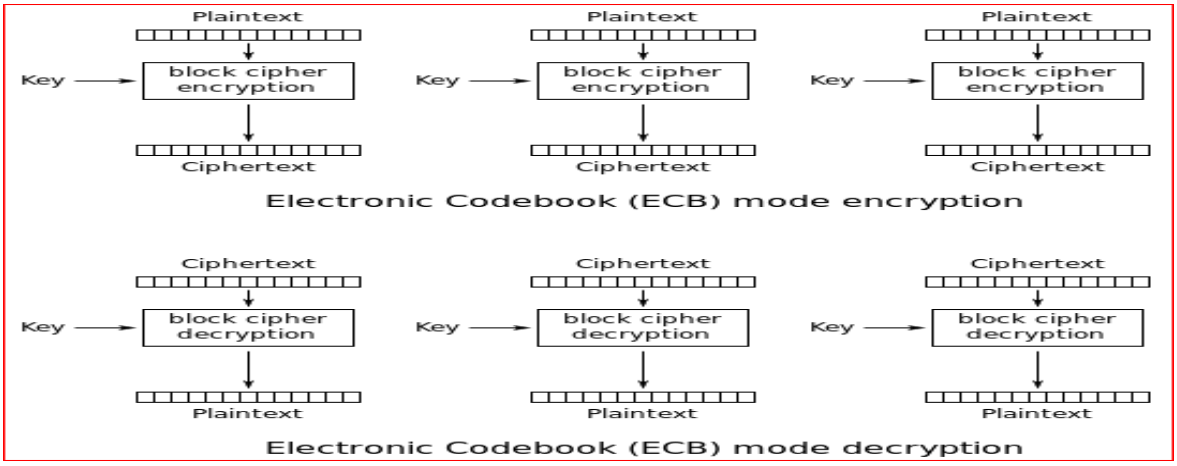


Figure 3.5: Chiffrement &déchiffrement par mode ECB.

La figure (3.5) illustre le processus de chiffrement en utilisant le mode ECB.

3.4.2.2 Le mode CBC :

Le mode CBC, Cipher Block Chaining, a été introduit pour qu'un bloc ne soit pas codé de la même manière s'il apparait dans deux messages différents ou s'il apparait deux fois dans un message**[34]**Le message, M, est découpé en blocs, (M_i) avec i≥1, et chaque bloc est crypté de la manière suivante. On commence par choisir un bloc initial C_0. Chaque bloc clair m_i est d'abord modifié en faisant un XOR de ce bloc avec le bloc crypté précédent, C_{i-1} puis on crypte le résultat obtenu par Exorcisation avec la clef par**[35]**:

$C_1 = E_k(M_1 \oplus C_0)$ $C_2 = E_k(M_2 \oplus C_1) \ldots\ldots C_i = E_k(M_i \oplus C_{i-1})$

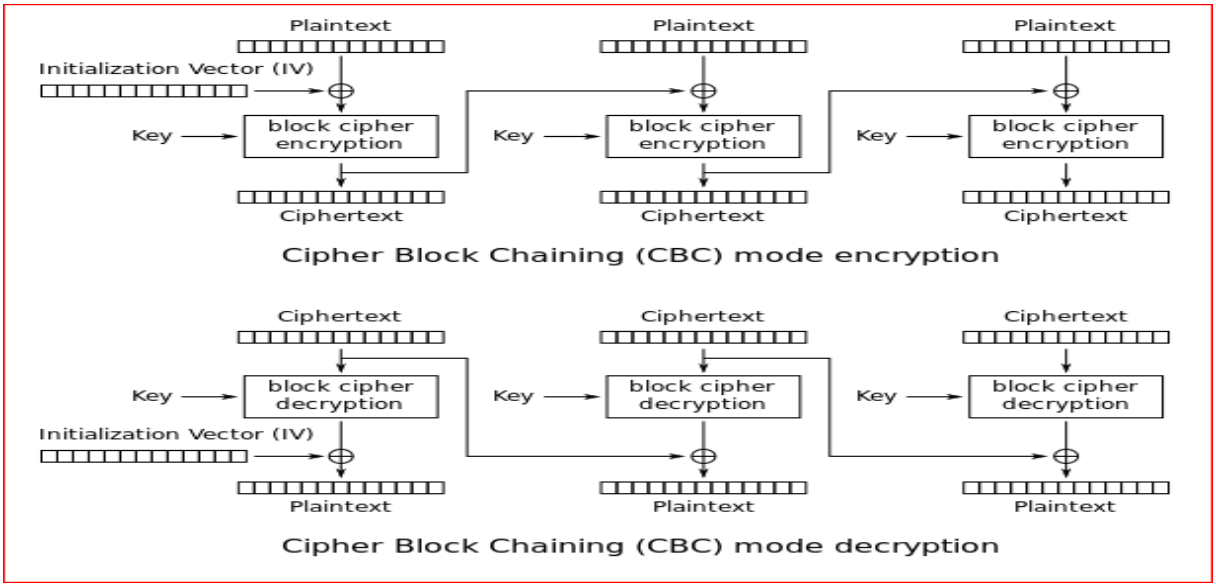


Figure 3.6: Chiffrement &déchiffrement par mode CBC.

La figure (**3.6**) illustre le processus de chiffrement en utilisant le mode CBC.

3.4.2.3 Le mode CFB :

Le mode CFB, CipherFeedBack, a été introduit pour ne pas avoir à calculer la fonction inverse, D_k, de la fonction de chiffrage E_k. Le principe est le même que celui du mode CBC**[34].**Le message, M, est découpé en blocs, (M_i) avec i≥1, et chaque bloc est crypté de la manière suivante. On commence par choisir un bloc initial m0, choisi suivant les mêmes principes que le blocC_0 en mode CBC. Chaque bloc clair mi est XORé avec le crypté du bloc de sortie précédent, C_{i-1}, suivant la figure ci-dessous :

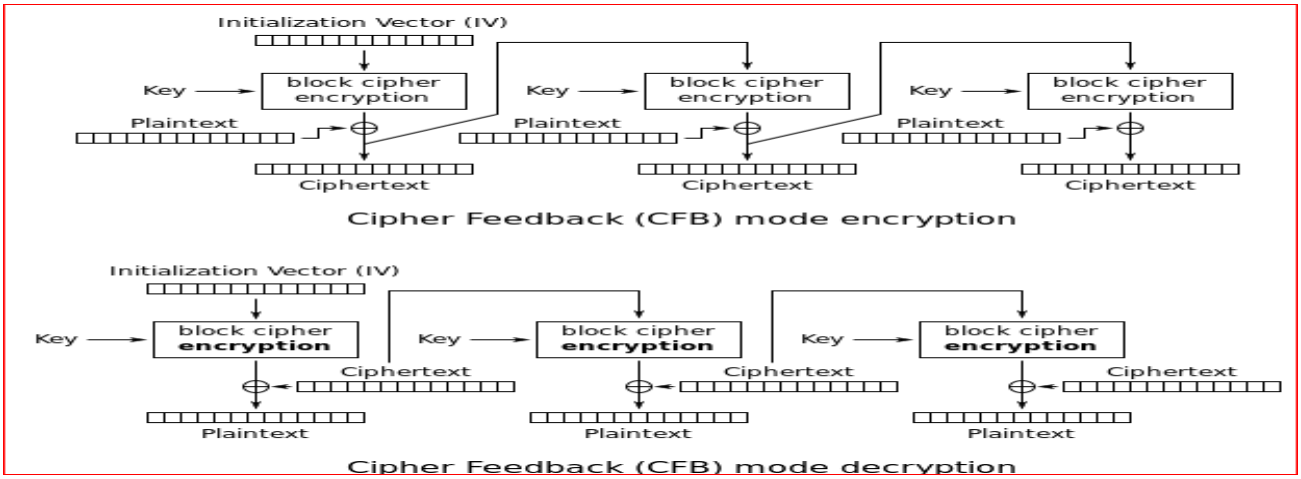


Figure 3.7: Chiffrement &déchiffrement par mode CFB.

Avec la formule **[35]**:

$$C_0 = E_k(M_0), \quad C_1 = M_1 \oplus E_k(C_0), \quad C_2 = M_2 \oplus E_k(C_1)$$

3.4.2.4 Le mode OFB :

Le mode OFB, Output FeedBack, est une variante de CFB qui permet d'avoir un cryptage et un décryptage totalement symétrique :

$Z = E_k(Z_{i-1})$; $C_i = M_i \oplus Z_i$

Ce mode est utilisé par exemple pour la sécurisation des donnés satellites. La figure (3.8) illustre le fonctionnement de ce mode**[35].**

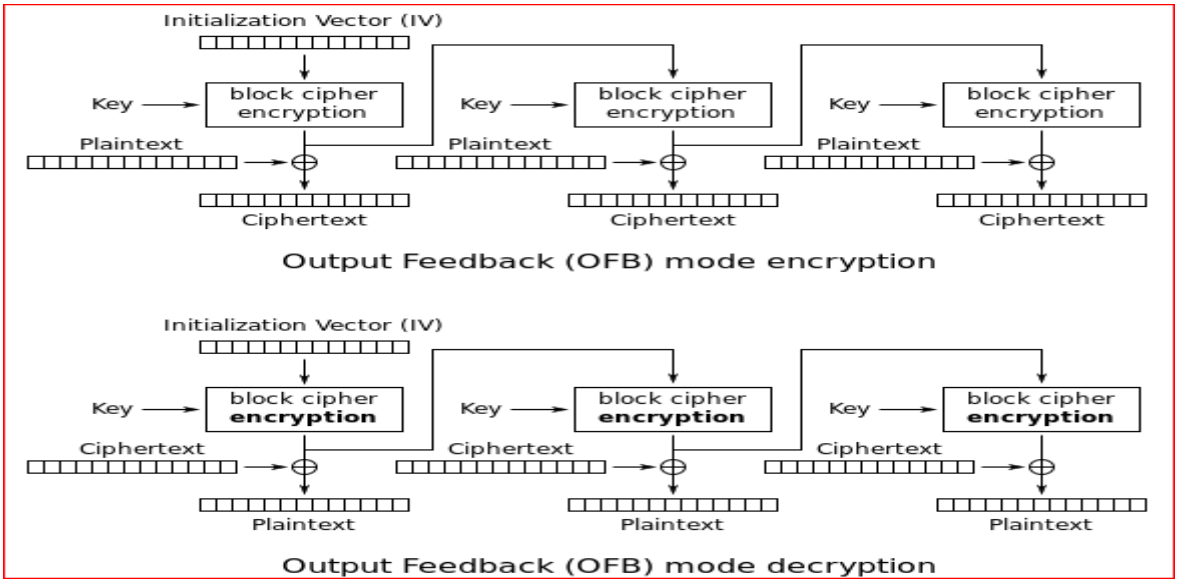


Figure 3.8: Chiffrement &déchiffrement par mode OFB.

3.4.2.5 Le mode CTR :

Counter-mode encryption. Le mode CTR, Counter-mode encryption. Ce mode de cryptage est lui aussi totalement symétrique, mais en outre facilement parallélisée. Il utilise pour le chiffrement un compteur de valeur initiale T**[34]**.

$$Ci = mi \oplus Ek(T + i)$$

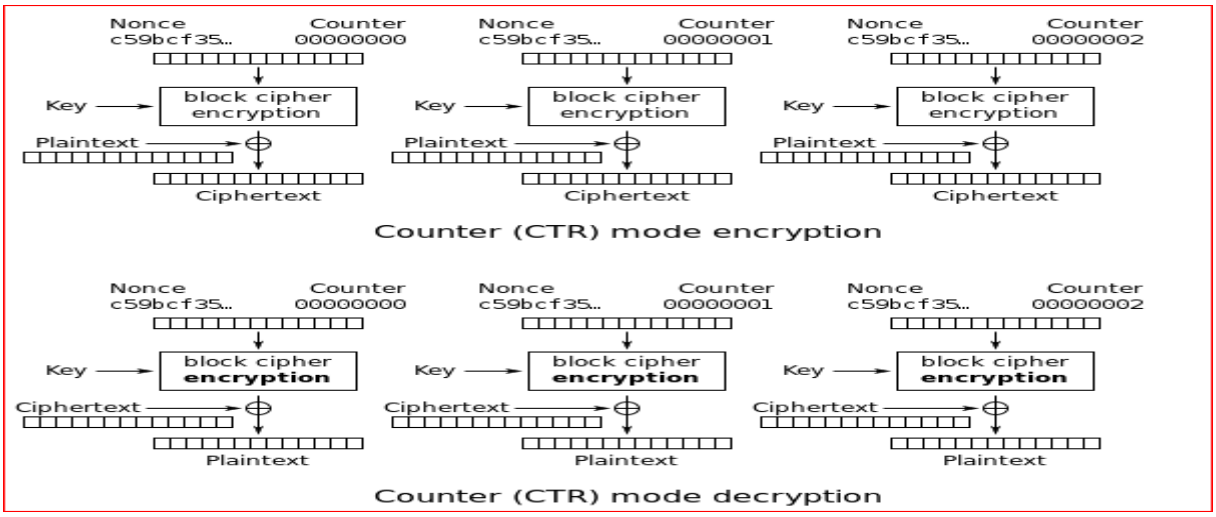


Figure 3.9: Chiffrement &déchiffrement par mode CTR.

3.4.3 Définition du Types des images étudiées :

Il existe aujourd'hui de nombreux types d'image. Des formats traditionnels comme JPG, GIF, TIF ou SVG jusqu'au format d'image propriétaire comme le format PSD du logiciel de traitement d'images Photoshop ou EPS créé par Adobe System. Chacun de ses types possèdent une vraie utilité et sont optimisés pour des cas d'utilisation précis.

Il existe trois types d'image :

Les images matricielles, les images vectorielles et les fichiers de logiciel de traitement d'images. Chacun de ces types est utilisé et créé de différente façon.

Image JPG : La définition d'une image numérique se mesure en pixel.

Toutes les images numériques sont constituées d'une "grille"de points élémentaires que l'on appelle les pixels.

Image PNG : PNG est un format d'image, ce format est agréable car il permet notamment de pouvoir mettre des fonds transparents.

Image GIF : GIF est un format d'image, il permet notamment de faire des "gifs animés".

Dans ce travail en a appliquer trois images différents pour calculer les métrises du chaque image.

Figure 3.10: Différents images utilisé pour la simulation.

3.5 Interfaces graphiques développées :

Le développement des interfaces graphiques peut être séparé en deux parties :

- Gestion de la mise en place et des propriétés des objets ;
- Programmation des interactions avec les objets.

Il existe deux méthodes de développement des interfaces graphiques sous MATLAB. La première utilise un outil graphique dédié et la seconde nécessite de programmer entièrement à la main. Nous allons donc aborder chaque méthode dans ce chapitre en l'expliquant avec illustration d'exemple simple.

Pour ce faire, nous allons prendre l'exemple d'une interface graphique contenant un objet Figure, un objet Axe et un objet Uicontrol de type Pushbutton comme le montre la figure ci-dessous**[32].**

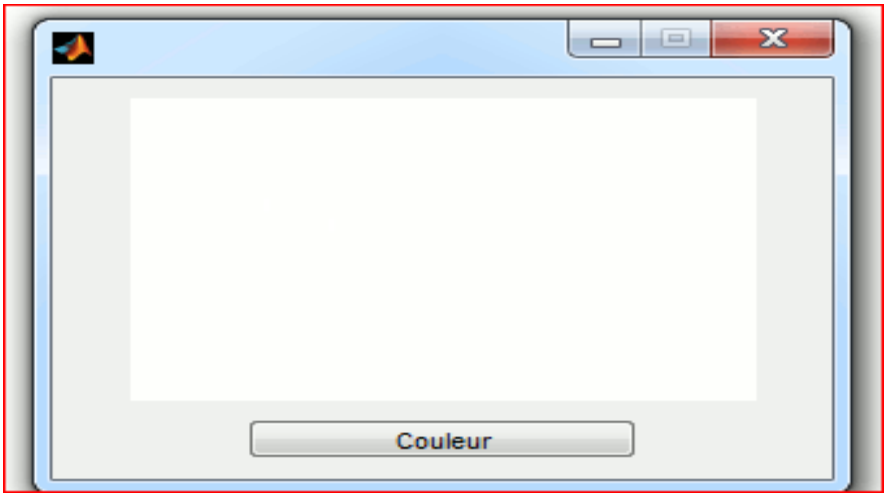


Figure 3.11: exemple d'une interface

.

Dans cette interface lorsque l'utilisateur clique sur l'objet Pushbutton couleur, l'objet Axes change de couleur de façon aléatoire.

3.5.1 Description de l'interface graphique :

Le GUIDE est un constructeur d'interface graphique qui regroupe tous les outils dont le programmeur à besoin pour créer une interface graphique de façon intuitive. Il s'ouvre, soit en cliquant sur l'icône, soit en tapant **guide** dans le Command Window de MATLAB**[32].** Le placement des objets est réalisé par sélection dans une boite à outils. Leur mise en place et leur dimensionnement se font à l'aide de la souris.

3.5.2 Présentation des différentes fenêtres de l'interface :

Matlab permet à l'utilisateur de programmer des interfaces graphiques interactives afinde présenter ses résultats. Les interfaces graphiques réalisables restent relativement simples.
Une interface graphique comprend des menus, des boutons, des "ascenseurs", des casesàcocher, des listes de choix, des zones de texte.

- Les notions principales d'une interface graphique sont :
- Les divers objets graphiques,
- Auxquels sont attribués des noms symboliques ;

La figure (**3.12**) nous montre la fenêtre principale du GUI.

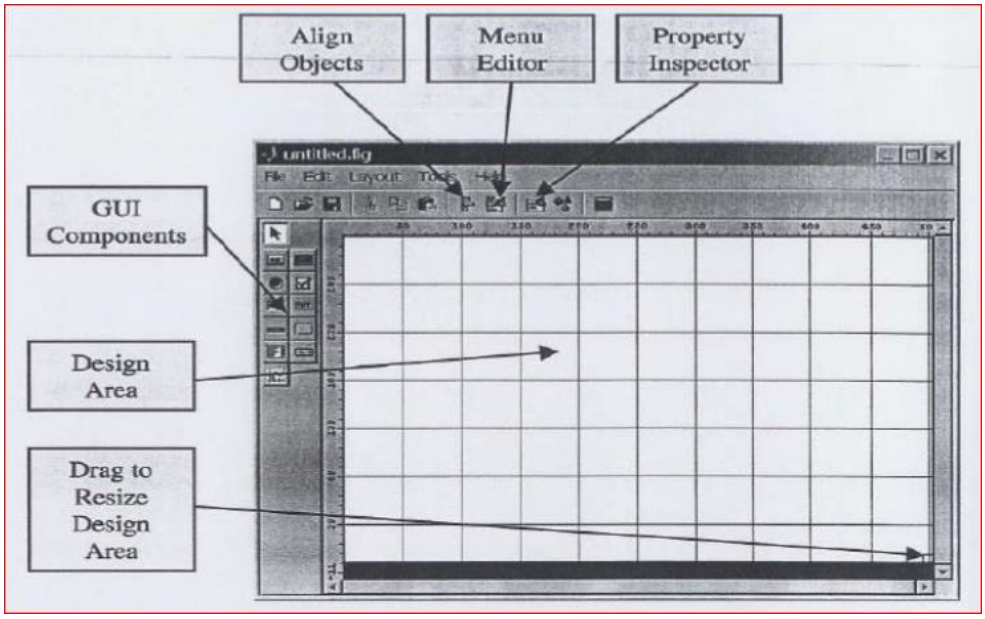


Figure 3.12: Fenêtre principale de GUI.

Un double-clique sur un objet permet de faire apparaître le PropertyInspector où les propriétés des objets sont facilement éditables. Leurs modifications et la visualisation de ces modifications sont immédiates.

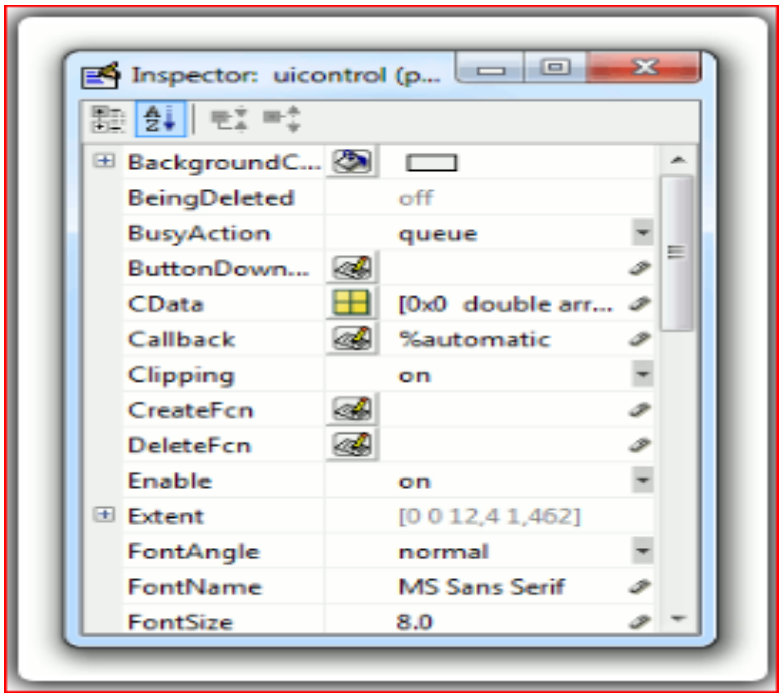


Figure 3.13: Property Inspector d'un button.

Le GUIDE possède également des outils pour gérer l'alignement des objets et pour créer des barres d'outil sou des menus.

Une fois l'interface graphique terminée, son enregistrement donne deux fichiers portant le même nommais dont les deux extensions sont**. Fig** et **.m.**

Le fichier .fig contient la définition des objets graphiques (positions et propriétés). Ce fichier peut être ouvert ultérieurement avec le GUIDE pour modifier les objets graphiques. Le fichier .m contient les lignes de code qui assurent le fonctionnement de l'interface graphique (actions des objets). Ce fichier peut être édité dans le MATLAB Editor pour y ajouter des actions à la main. C'est ce fichier qui doit être lancé pour utiliser l'interface graphique**[32].**

3.5.3 Exemple l'exécution

Lors de l'enregistrement, le GUIDE génère deux fichiers :

- Un fichier. fig (non éditable) contenant les objets graphiques Figure, Axes et Pushbutton ;
- Un fichier .m contenant le code du fonctionnement de l'interface graphique.

Il reste ensuite à ajouter au fichier .m, le code correspondant à l'action à effectuer au moment du clique sur le bouton, à savoir le changement de couleur de l'objet Axes.

Dans notre cas, il faut ajouter la ligne suivante :

```
Set (handles.axes1, 'color', rand(1,3));
```

Elle se place à la fin du code dans la fonction **pushbutton1_Callback** Le contenu du fichier .m créé par le GUIDE**[32]**

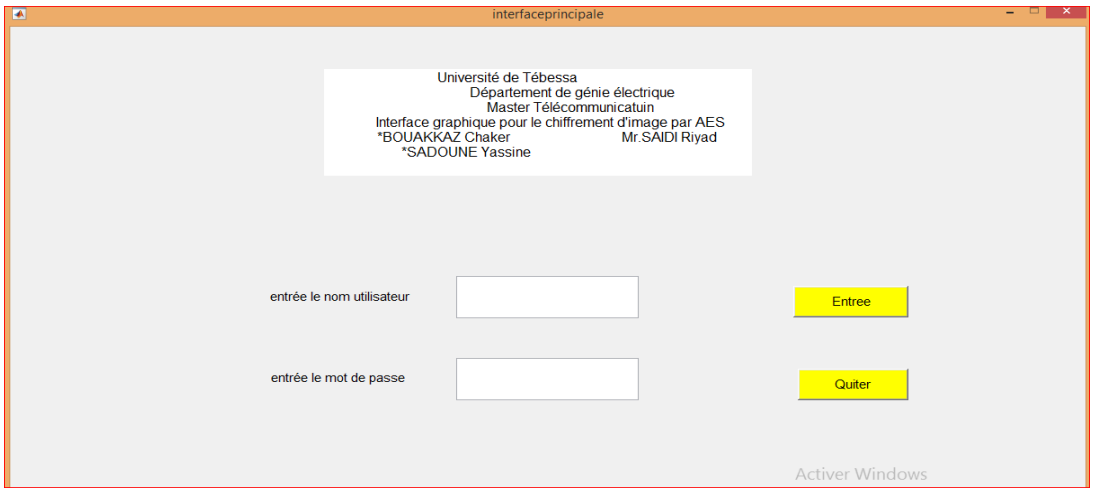


Figure 3.14: Interface principale.

La figure (**3.14**) représente l'interface principale de notre crypto système, c'est l'interface d'entré protégé par un nom d'utilisateur et un mot de passe.

3.5.4 Résultats après exécution premier interface l'algorithme AES sans mode

Cette fenêtre nous permet le choix de l'opération à réaliser avec notre crypto système soit choix des chiffrements ou fermer la fenêtre par la figure(**3.15**).

Figure 3.15: Fenêtre de choix des chiffrements.

Cette interface nous permet de ce connecté vers d'autres interfaces de la simulation.

Cette opération nous permette le choix de l'opération a effectués soit chiffrements par AES ou chiffrement avec les modes comme le montre de la figure (**3.16**) et le retour pour revenir à l'interface précédente ou fermer la fenêtre.

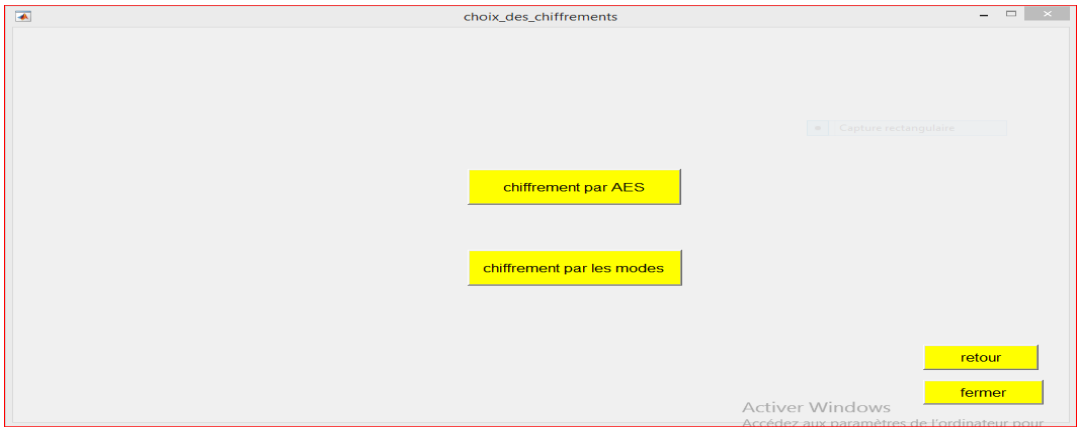


Figure 3.16: Interface pour choix les chiffrements.

Dans le premier choix c'est le chiffrement par l'algorithme AES sans utilisation des modes. Dans cette partie une image JPG 128bits est chiffrée avec l'algorithme AES sans utilisation des modes l'image est redimensionnée pour avoir un multiple de block 4x4 = 16, qui sera de même taille avec la clé. En cliquant sur le bouton chiffrement par AES qui est présente dans l'interface illustré par la figure (**3.16**), elle va nous connecter vers une autre interfacequi est illustré par la figure (**3.17**).

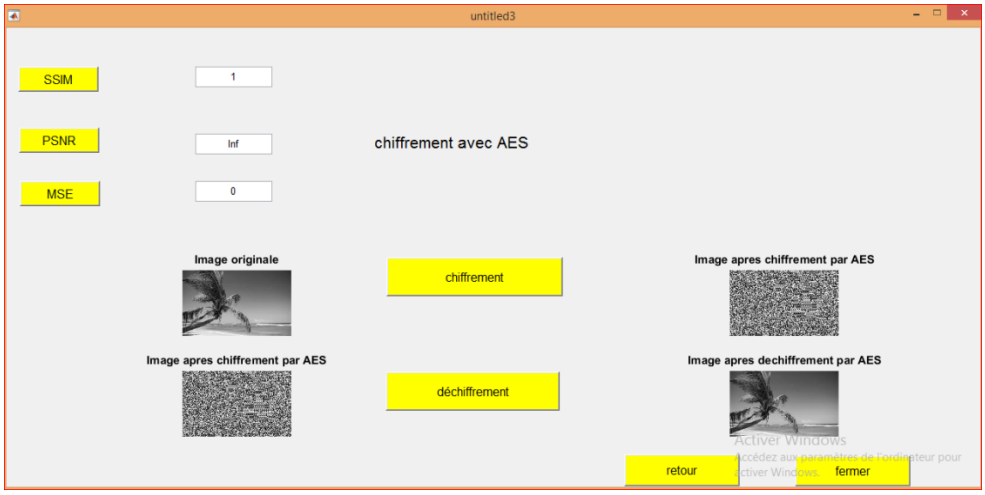


Figure 3.17: Interface de chiffrement et déchiffrement par AES.

L'utilisateur peut insérer dans cette dernière l'image à chiffré avec sa propre clé comme le montre la figure, après exécution une image chiffrée est affichée ainsi que l'image déchiffrée, avec l'affichage aussi des métrise pour faire une évaluation entre l'image en clair et l'image déchiffrée qui sont Le SSIM, MSE et le PSNR qui seront définies plus loin dans ce chapitre.

3.5.5 Résultats après exécution deuxième interface l'algorithme AES avec mode

Dans la deuxième interface en a le choix de chiffrement AES avec plusieurs modes de chiffrement et déchiffrement, en choisi cette option est présente dans l'interface illustré par la figure (**3.16**), elle va nous connecter vers une autre interface qui est illustré par la figure (**3.18**).

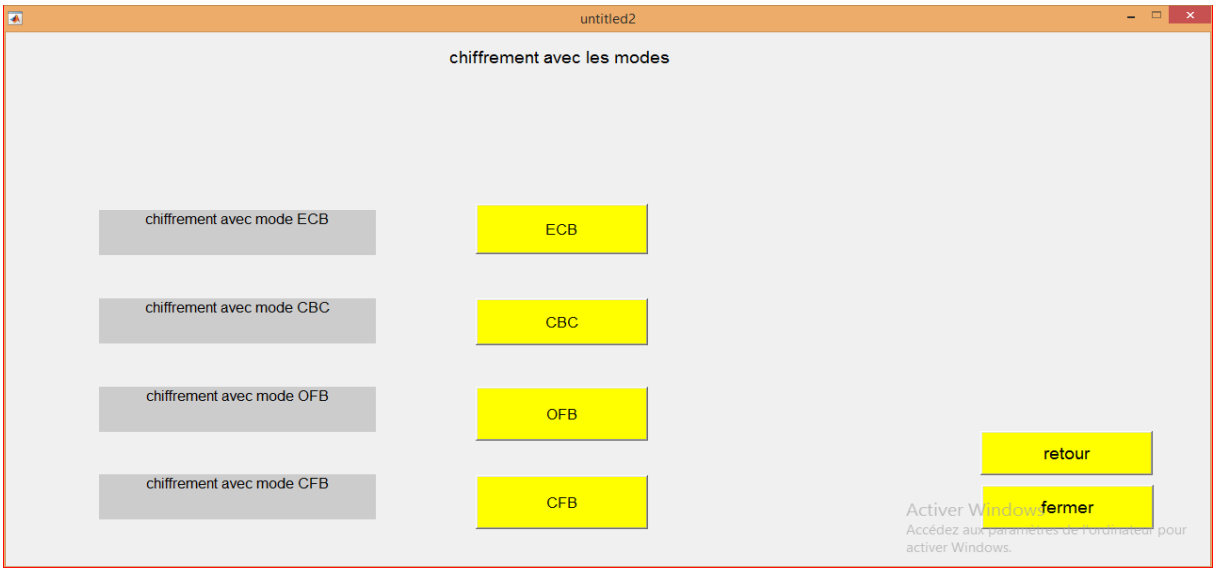


Figure 3.18: Interface pour choix les modes du chiffrement.

a) Le mode ECB : carnet de codage électronique

Le mode de chiffrement par carnet de codage électronique, qu'on trouve souvent appelé mode ECB (de l'anglais **E**lectronic **C**ode **B**ook), est la façon la plus simple de mettre en oeuvre un chiffrement par blocs. on choisi cette option dans l'interface illustré par la figure (**3.18**) qui va nous connecter vers l'interface illustré par la figure (**3.19**).

Dans cette partie pour chaque mode on a trouver trois image GPJ,PNG etGIF pour appliquer le chiffrement&déchiffrement et fait comparaison entre les images.

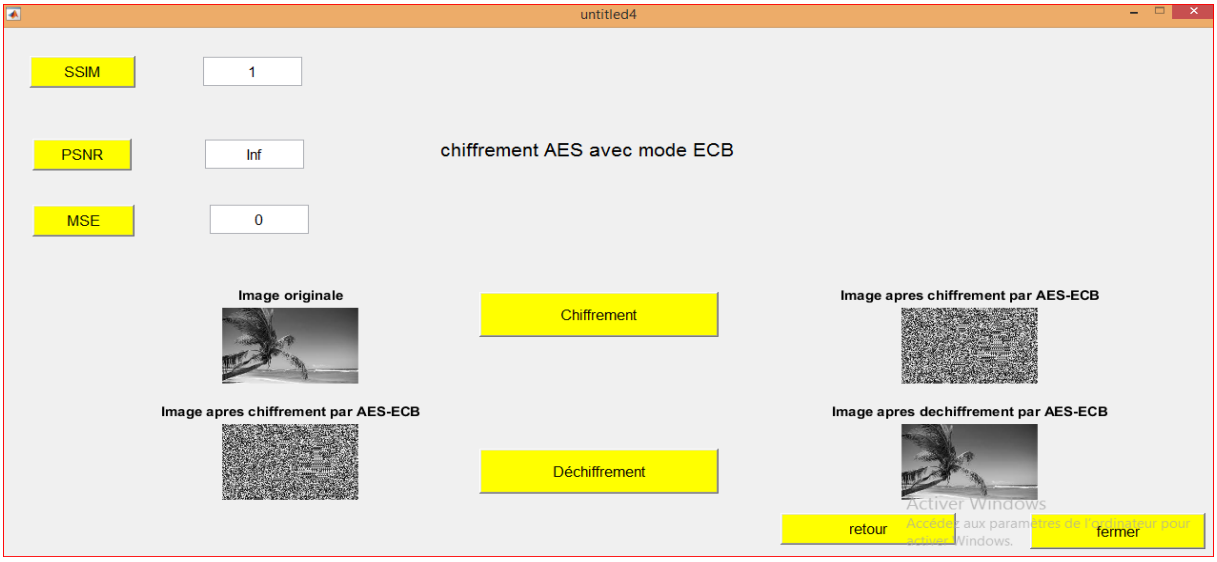


Figure 3.19: Interface de chiffrement & déchiffrement par AES mode ECB image1.

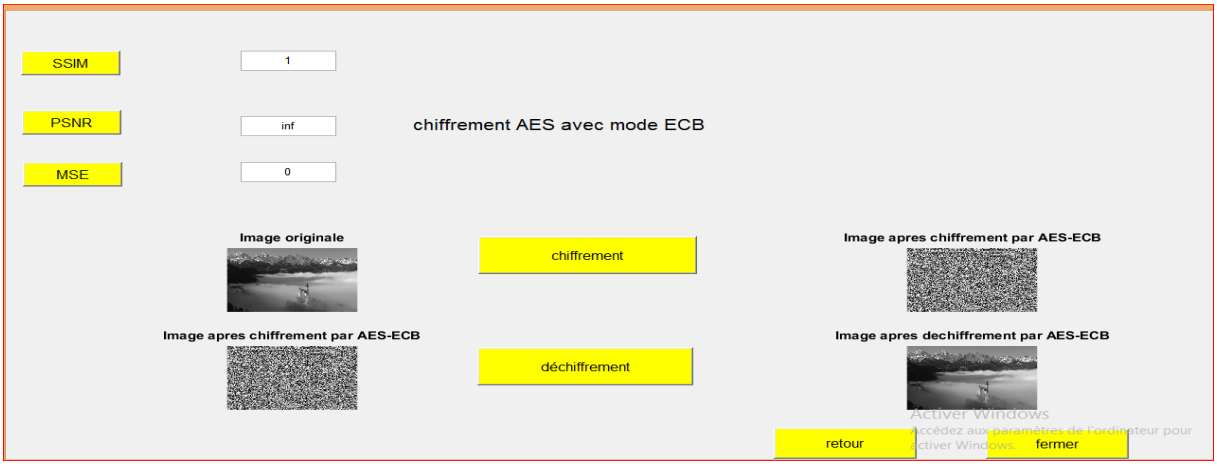


Figure 3.20: Interface de chiffrement & déchiffrement par AES mode ECB image2.

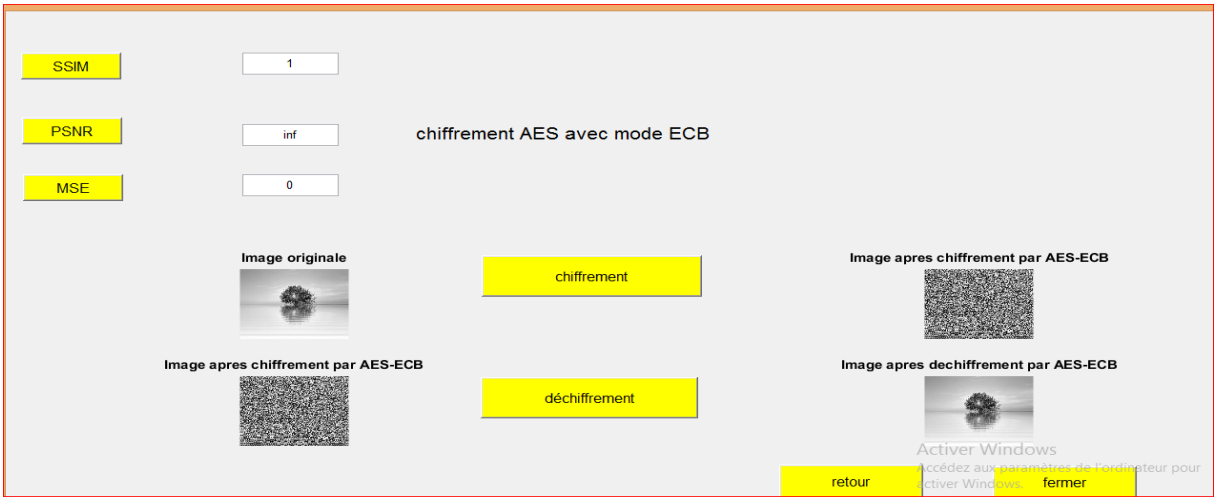


Figure 3.21: Interface de chiffrement & déchiffrement par AES mode ECB image3

.

Dans cette interface l'utilisateur peut insérer l'image à chiffré avec sa propre clé comme le montre la figure, après exécution une image chiffrée est affichée ainsi que l'image déchiffrée, avec l'affichage aussi des métrise pour faire une évaluation entre l'image en clair et l'image déchiffrée pour ce mode de chiffrement, qui sont Le SSIM, MSE et le PSNR.

Ce mode souffre de plusieurs défauts de sécurité :

- Deux blocs clairs identiques sont chiffrés de la même façon.
- Le mode ECB ne respecte pas l'intégrité des données.

b) Le mode CBC : chiffrement par chaînage de blocs

Le mode de chiffrement par chaînage de blocs, en anglais **C**ipher **B**lock **C**haining, évite les deux problèmes précédents, car désormais le chiffrement dépend du contexte. On commence par fixer un mot initial de n bits VI, et on pose C_0=VI, comme définie précédemment.

En choisi cette option dans l'interface illustré par la figure (**3.18**) qui va nous connecter vers l'interface illustré par la figure (**3.22**).

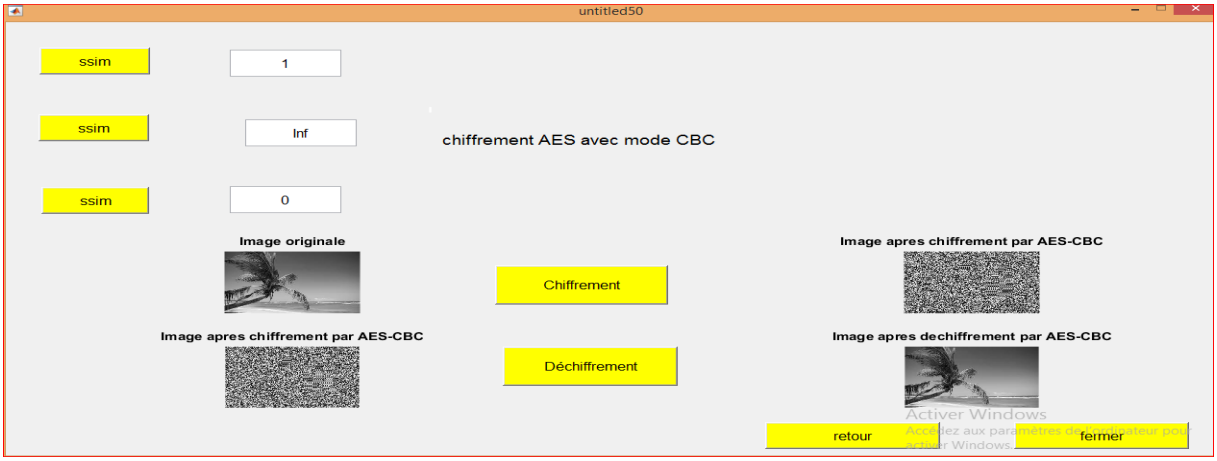


Figure 3.22: Interface de chiffrement & déchiffrement par AES mode CBC image1.

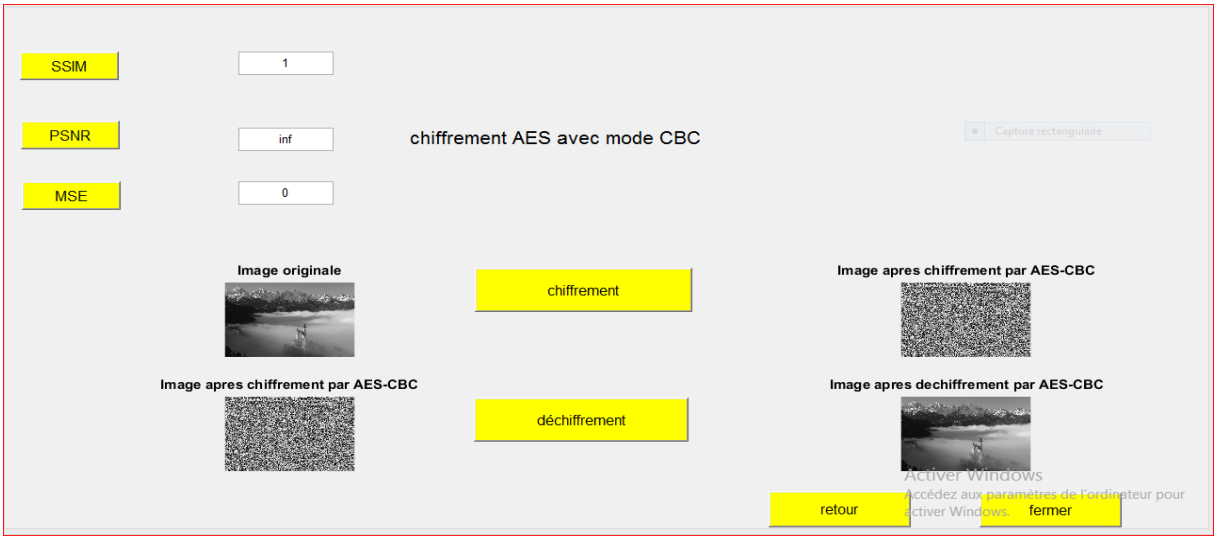


Figure 3.23: Interface de chiffrement & déchiffrement par AES mode CBC image2.

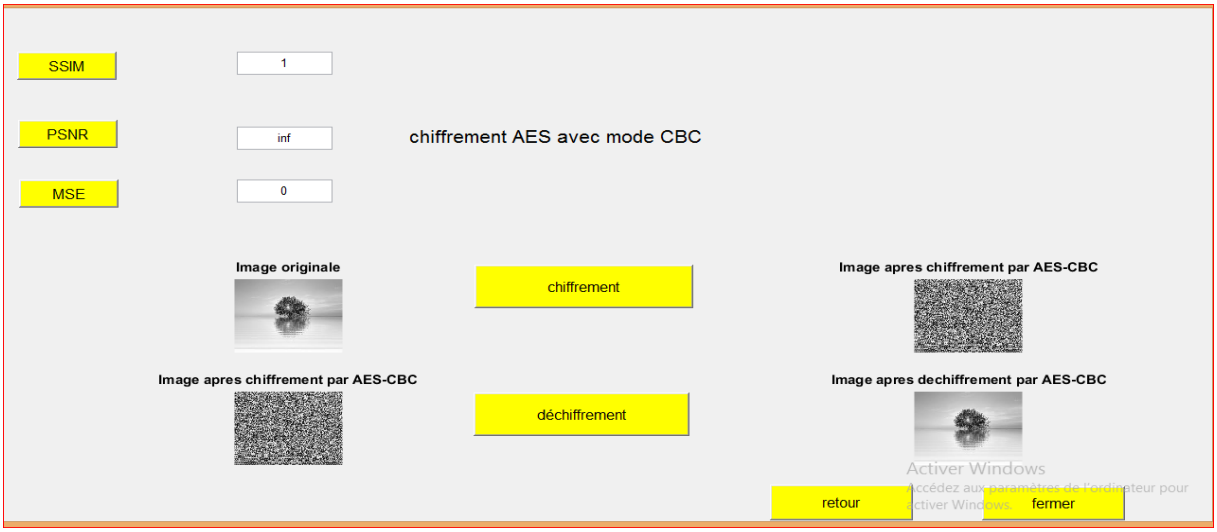


Figure 3.24: Interface de chiffrement & déchiffrement par AES mode CBC image3.

Cette interface représente l'opération de chiffrement et de déchiffrement d'une image par AES avec mode CBC.

Dans cette interface l'utilisateur peut insérer l'image à chiffré avec sa propre clé comme le montre la figure, après exécution une image chiffrée est affichée ainsi que l'image déchiffrée, avec l'affichage aussi des métrise pour faire une évaluation entre l'image en clair et l'image déchiffrée pour ce mode de chiffrement, qui sont Le SSIM, MSE et le PSNR.

Ce mode a plusieurs avantages, Le mode CBC chiffre le même message clair différemment avec des blocs d'initialisation différents.

c) Les modes CFB : chiffrement par rétroaction

Le mode de chiffrement de rétroaction (mode CFB, Cipher Feed Back) a un esprit très différent des modes précédents. Il s'agit cette fois d'utiliser la fonction de chiffrement CK comme un générateur pseudo-aléatoire de clés.

En choisi cette option dans l'interface illustré par la figure (**3.18**) qui va nous connecter vers l'interface illustré par la figure (**3.25**).

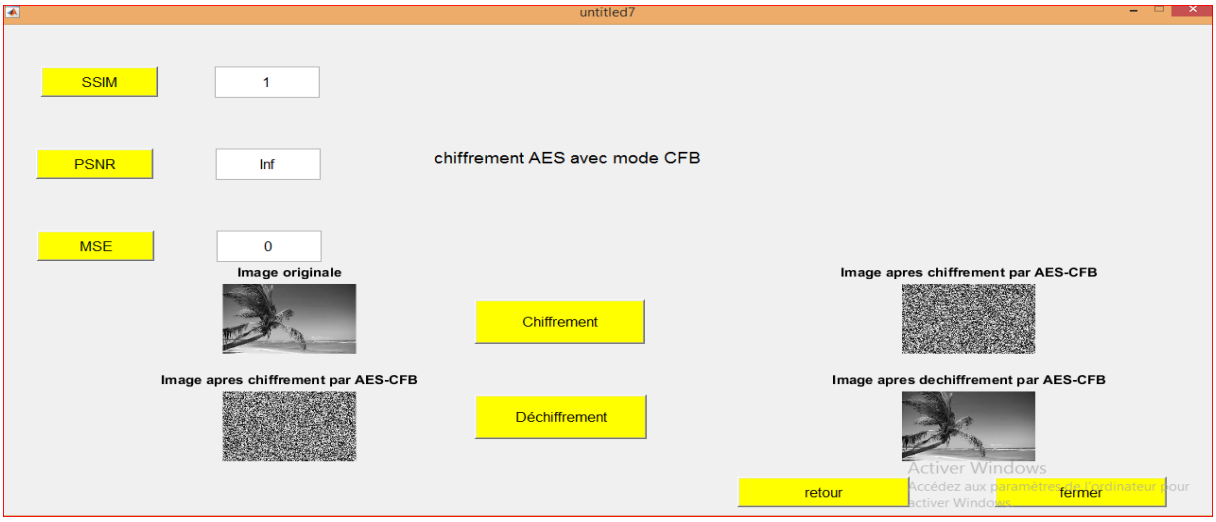


Figure 3.25: Interface de chiffrement & déchiffrement par AES mode CFB image1.

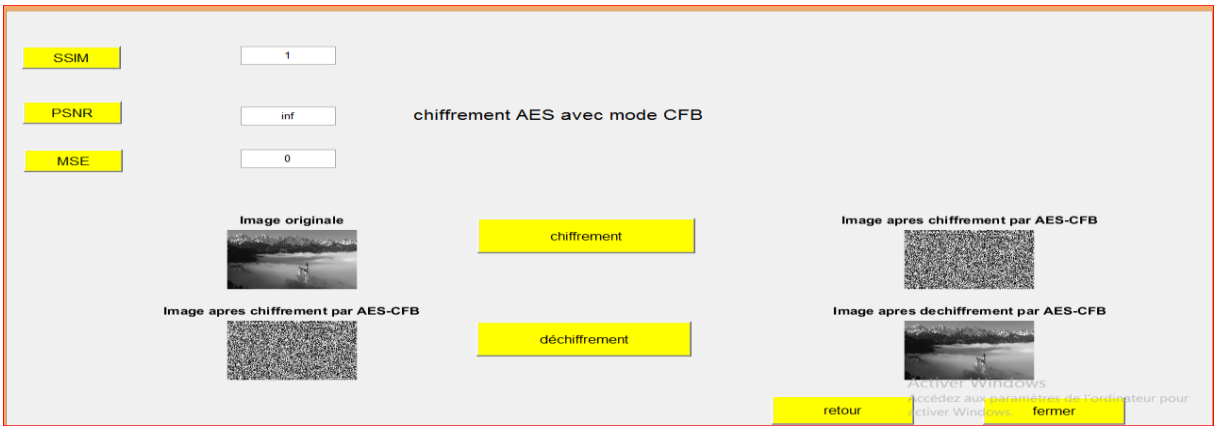


Figure 3.26: Interface de chiffrement & déchiffrement par AES mode CFB image2**.**

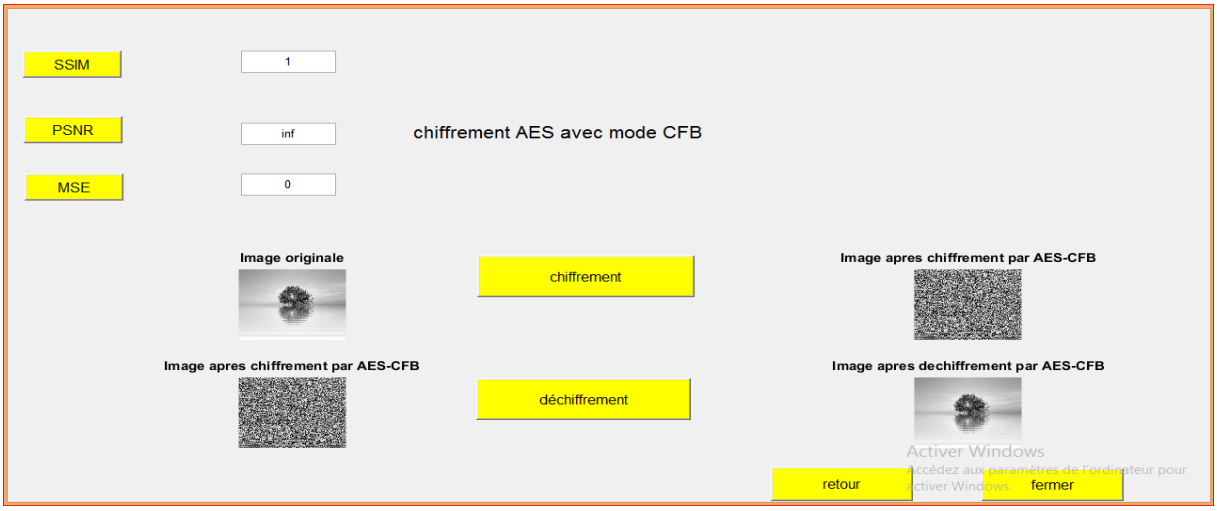


Figure 3.27: Interface de chiffrement & déchiffrement par AES mode CFB image3.

Ces interfaces représentent l'opération de chiffrement et de déchiffrement de trois images différentes par AES avec mode CFB.

d) Les modes OFB : chiffrement par rétroaction

Le mode de chiffrement de rétroaction de sortie (mode OFB, Output Feed Back) a aussi un esprit très différent des modes ECB et CBC. Il s'agit-il aussi d'utiliser la fonction de chiffrement CK comme un générateur pseudo-aléatoire de clés. En choisi cette option dans l'interface illustré par la figure (**3.18**) qui va nous connecter vers l'interface illustré par la figure (**3.28**).

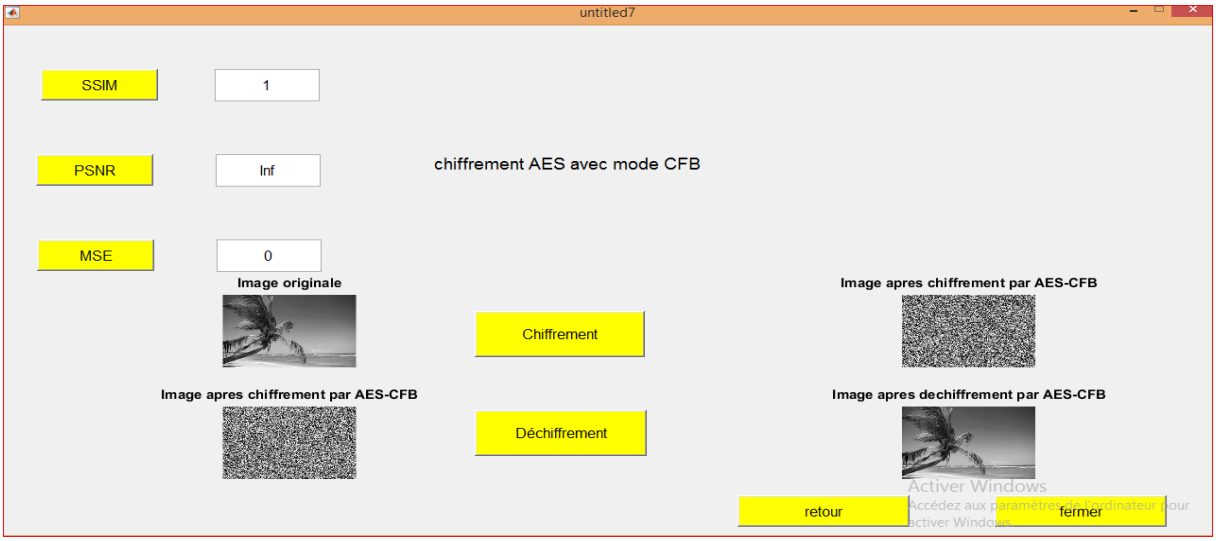


Figure 3.28: Interface de chiffrement & déchiffrement par AES mode OFB image1.

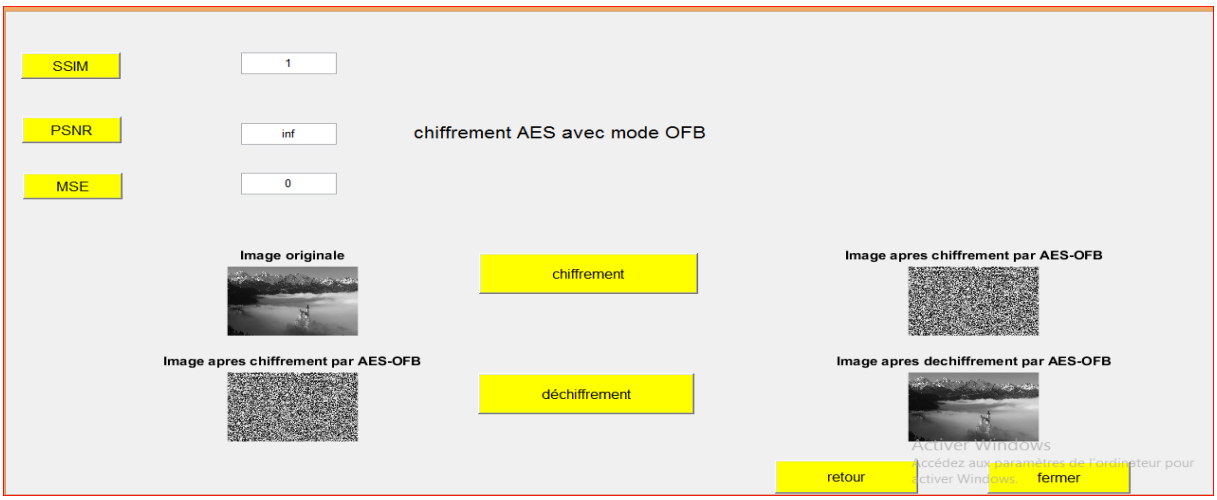


Figure 3.29: Interface de chiffrement & déchiffrement par AES mode OFB image2.

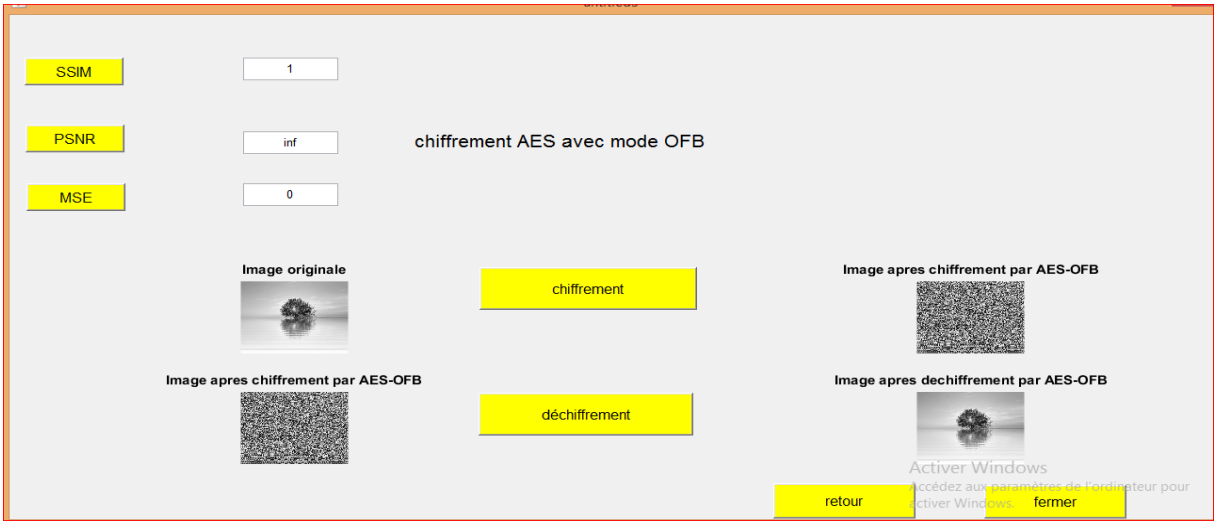


Figure 3.30: Interface de chiffrement & déchiffrement par AES mode OFB image3.

Dans les modes CFB et OFB, le bloc d'initialisation VI ne doit pas nécessairement être secret. En revanche, il doit être à usage unique. En effet, si on utilise deux fois le même bloc d'initialisation, on obtient les mêmes problèmes qu'en utilisant deux fois la même clé dans le chiffrement par masque jetable.

3.6 Analyse des Métries des résultats de chiffrement

3.6.1 Analyse des performances

Pour pouvoir analyser les performances des résultats issues du chiffrement avec l'algorithme AES seul ou avec les différents modes, en a utilisés différentes métrise d'évaluations des qualités des résultats des images chiffres par rapport des images original ou en clair. Les outils utilisés dans ce travail sont :

L'erreur quadratique moyenne (**MSE**),

Le rapport crête signal sur bruit (**PSNR**),

Indice de Similarité structurelle (**SSIM)**.

3.6.2 Erreur quadratique moyenne (MSE)

L'image déchiffrée $\hat{I}$ est toujours comparée à l'image originale I ou en claire pour déterminer son rapport de similitude. Ce critère est le plus utilisé. Il est basé sur la mesure de l'erreur quadratique moyenne (MSE) calculée entre les pixels originaux et celle déchiffrées **[36]**:

$$MSE = \frac{1}{M \times N} \sum_{m=1}^{M} \left(I(m,n) - I(\acute{m},n) \right)^2 \qquad (3.1)$$

Où (M × N) qui désigne la taille de l'image original et déchiffrée, et $I\,p$ et $\hat{I}\,p$ sont respectivement les amplitudes des pixels sur les images originale et déchiffrée. Il est vraisemblable que l'œil tienne beaucoup plus compte des erreurs à grandes amplitudes, ce qui favorise la mesure quadratique. Plus la valeur de MSE est faible, plus l'erreur est faible. Si le MSE est égal à zéro, cela signifie que l'image d'origine et l'image déchiffrée sont identiques et sa valeur PSNR sera l'infini.
Les résultats obtenus sont mentionnés dans le tableau (3.1)

Tableau 3.1 : Valeur MSE pour les quatre modes

	Image 1				**Image 2**				**Image 3**			
Mode de chiffrement	**ECB**	**CBC**	**CFB**	**OFB**	**ECB**	**CBC**	**CFB**	**OFB**	**ECB**	**CBC**	**CFB**	**OFB**
MSE	**0**				**0**				**0**			

On remarque selon le tableau (3.1) que la valeur du MSE des trois images est égale à zéro dans tous les modes, qui impliquent que les images déchiffrées et l'originales sont identiques.

3.6.3 Rapport crête signal sur bruit (PSNR)

Pour ce critère d'évaluation, au lieu de mesurer la distorsion, la valeur (Peak Signal to Noise Ratio, PSNR) mesure la fidélité de l'image déchiffrée par rapport à l'original ou en clair, puisqu'elle est proportionnelle à la qualité. Comme nous pouvons le voir selon l'équation III.3, elle est une fonction de MSE ; sa définition et son utilisation proviennent du domaine du traitement de signal **[36]**:

$$PSNR = 10\,log_{10}\left[\frac{\frac{1}{N}\sum I^2}{MSE}\right] \qquad (3.2)$$

Pour une image à niveau de gris, *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**[36]**.

Plus le PSNR est élevé, l'image déchiffrée est similaire a l'originale.

L'erreur quadratique moyenne (MSE) et le rapport signal / bruit de crête (PSNR) sont utilisés à l'origine pour comparer la qualité de compression d'image nous les avons utilisés pour évalués la qualité des images déchiffrées par rapport à l'original.

Les résultats obtenus après opération de chiffrement et déchiffrement utilisant les différents modes pour les trois images utilisées sont donnes par le tableau (3.2).

Tableau 3.2 : Valeur PSNR pour les quatre modes

	Image 1				**Image 1**				**Image 1**			
Mode de chiffrement	**ECB**	**CBC**	**CFB**	**OFB**	**ECB**	**CBC**	**CFB**	**OFB**	**ECB**	**CBC**	**CFB**	**OFB**
PSNR	Inf				Inf				Inf			

Il existe une relation inverse entre le PSNR et le MSE. Ainsi, une valeur PSNR plus élevée indique la meilleure qualité de l'image.

On remarque selon les résultats indiques dans le tableau (3.2), que la valeur du PSNR des trois images pour les cinq modes est égale à l'infinie, qui veux dire que les images déchiffrées et l'originales sont identiques.

3.6.4 Indice de similarité structurelle SSIM

SSIM est une mesure de similarité entre deux images numériques. Elle a été développée pour mesurer la qualité visuelle d'une image déformée, par rapport à l'image originale. L'idée de SSIM est de mesurer la similarité de structure entre les deux images, plutôt qu'une différence pixel à pixel comme le fait par exemple le PSNR. L'hypothèse sous-jacente est que l'œil humain est plus sensible aux changements dans la structure de l'image [36].

La métrique SSIM est calculée sur plusieurs fenêtres d'une image. On dénote *x* et *y* l'image originale et l'image déformée respectivement.

La similarité compare la luminance, le contraste et structure entre chaque couple de fenêtres. La luminance est estimée par la mesure de l'intensité moyenne de chaque fenêtre **[36]**:

$$\mu_x = \frac{1}{N}\sum_{1}^{N} x_i \qquad (3.3)$$

Où :
N : le nombre de pixels de chaque fenêtre.
x_i : l'intensité d'un pixel.

Les résultats obtenus après opération de chiffrement et déchiffrement utilisant les différents modes pour les trois images utilisées sont donnes par le tableau (3.3).

Tableau 3.3: Valeur SSIM pour les quatre modes.

	Image 1				**Image 1**				**Image 1**			
Mode de chiffrement	**ECB**	**CBC**	**CFB**	**OFB**	**ECB**	**CBC**	**CFB**	**OFB**	**ECB**	**CBC**	**CFB**	**OFB**
SSIM	**1**				**1**				**1**			

On va remarquer que la valeur du SSIM des trois images est égale à 1 dans tous les modes, donc les images déchiffrées et l'originales sont identiques.

Comparaison des résultats des différents modes.

Après les chiffrements et déchiffrements on à résumer les résultats des différentes métrise (SSIM, PSNR, MSE) dans ce tableau :

Tableau 3.4:Résultats des métrise.

	Image 1											
Mode de chiffrement	**ECB**			**CBC**			**CFB**			**OFB**		
Métrise	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNER**	**SSIM**
	0	**Inf**	**1**	**0**	**Inf**	**1**	**0**	Inf.	**1**	**0**	**Inf**	**1**
	Image 2											
Mode de chiffrement	**ECB**			**CBC**			**CFB**			**OFB**		
Métrise	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNER**	**SSIM**
	0	**Inf**	**1**	**0**	**Inf**	**1**	**0**	**Inf**	**1**	**0**	**Inf**	**1**
	Image 3											
Mode de chiffrement	**ECB**			**CBC**			**CFB**			**OFB**		
Métrise	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNR**	**SSIM**	**MSE**	**PSNER**	**SSIM**
	0	**Inf**	**1**	**0**	**Inf**	**1**	**0**	**Inf**	**1**	**0**	**Inf**	**1**

Dans ce tableau on va réduire tous les métrise des tous les images, donc les trois images d'origine identique avec tous les images chiffrée par tous les modes.

3.7 Conclusion

Dans ce chapitre nous avons utilisé une application de cryptage pour chiffrer trois images différents grâce à des interfaces graphiques, ou nous avons choisir l'algorithme AES seul et aussi choisir les modes ECB, CBC, OFB et CFB. Et pour l'évaluation des résultats nous avons utilisé trois métrises tel que MSE, SSIM, et le PSNR, ou les résultats est été satisfaisantes.

Conclusion générale

Dans ce mémoire, nous avons étudié le problème lié à la protection des images. Qui concerne la transmission sécurisée d'images ainsi d'assurer la fonction de confidentialité des images transmis. Pour cela nous avons développé une interface graphique pour faire face à ce problème de confidentialité qui se base sur l'algorithme AES (Advanced Encryption Standard).

Dans un premier temps, nous avons présenté une introduction à la cryptographie du côté historique, vocabulaire utilise, notation et type. En aborder aussi les protocoles nécessaires pour assurer la sécurité tel que Confidentialité, Intégrité et Authentification, les types de chiffrement symétriques et asymétriques sont aussi aborder.

Dans un deuxième temps, nous avons évoqué la cryptographie moderne avec les deux grandes classes symétrique et asymétrique, en abordons quelques algorithmes tel que l'AES et DES pour le chiffrement symétrique, et l'algorithme RSA, les courbes elliptiques pour le chiffrement asymétriques, avec une comparaison de ces grandes classes de chiffrement.

Dans la troisième partie, nous avons présenté en détail notre travail qui se penches sur le Côté cryptographique, qui s'inscrit dans le cadre de la sécurité des images, par l'algorithme AES. Le caractère de ce travail consiste à proposer un Cryptosystème à base de l'algorithme AES pour assurer la confidentialité. Nous avons présenté en général la constitution d'une interface graphique, c'est éléments de base, une description de l'algorithme AES ainsi que les types d'images utilisé qui est décrites afin de l'adapter pour une transmission en mode chiffré. Notre Cryptosystème est composée deux étages : étage d'émission et celui de la réception. A l'émission la clef est introduite puis l'image que nous voulons transmettre est sélectionnée. Ensuite nous avons chiffré cette dernière par l'algorithme AES seul afin de garantir la confidentialité. A la réception de ces images, des fonctions réversibles sont élaborés pour les déchiffrer, ainsi les que quelques métrise. Puis l'utilisation des modes de chiffrement tel que CBC, ECB, OFB, CFB, et CTR sont introduits en association avec l'AES pour augmenter la sécurité.

Pour tester notre Cryptosystème et analyser ces résultats, nous avons exploité plusieurs métriques d'évaluation de la qualité des images déchiffrement (calcul de l'indice de similitude SSIM, le calcule erreur quadratique moyenne MSE et le calcule aussi du Rapport crête signal sur bruit PSNR) entre l'image originale et celle déchiffrée, Le nombre d'images que nous avons chiffré est de 12 images. A partir des résultats obtenus, nous avons remarqués que toutes les métrise exploites dans ce travail tel que SSIM, MSE, et PSNR pour les modes d'opérations (OFB, CFB,

CBC, CTR) sont de valeurs qui nous indiques que l'image déchiffrée et la même que l'image original, ce qui signifie que les informations que porte l'image ne sont pas altérer.

Pour un travail de perspective nous contant le développement de Cryptosystème hybride pour satisfaire d'autres critères de sécurité tel que l'intégrité et l'authentification, ainsi ajouter une étape de codage et de compression de données pour améliore la sécurité de plus, et pour optimiser le système de cryptage et de transmission d'images.

BIBLIOGRAPHIE

[1] J. Daemen, V. Rijmen, "AES proposal: the rijndael block cipher," Belgium, 2002

[2] W. Stallings, "cryptography and network security principles and practice," Pearson, United State of America, 2011, p. 900.

[3] B Schneier, "applied cryptography," CRC press, United States of America, 1996, p. 780.

[4] M. Dworkin, "recommendation for block cipher modes of operation," NIST special publication 800-38, 2001 edition.

[5] R. Stinson, "cryptography: theory and practice, (discrete mathematics and its applications)," York, November 2005.

[6] D. KAHN, "The Codebreakers: The Story of Secret Writing", New York: Macmillan Publishing Co., 1967

[7] G. Brassad, « cryptologie contemporaine, »Edition CASSINI 1999.

[8] G. Zemor, «Cours de cryptographie,» Edition MASSON, 2000.

[9] S. Jacques, "La science du secret", Editions Edile Jacob, 1998

[10] B. Schneier, "Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C", John Wiley & Sons, Inc., 01/01/96

[11] D. El Khier, « Etude et comparaison des principaux systèmes de cryptage », diplôme de magister en informatique, université M'sila, 2006.

[12] Hacini Souleyman Boumedyen, "Implémentation d'algorithmes de Cryptographie", Université AbouBakr Belkaid– Tlemcen Faculté des Sciences Département d'Informatique, 2013-2014

[13] H. Lester's, "Cryptography in an Algebraic Alphabet", American Mathematical Monthly-N36-P306à312, 1929

[14] R. Dumont, « cours de Cryptographique Sécurité Informatique, » Université de Liège (Belgique), faculté des Sciences Appliquées, 2009- 2010

[15] A. Mayank Dave, RC Joshi, "DNA cryptography: a novel paradigm for secure routing in mobile ad hoc networks (Manets)." Journal of Discrete Mathematical Sciences and Cryptography, 11(4):393–404, 2008

[16] F. Autrusseau, "Lossless compression based on a discrete and exact radon transform: A preliminary study," International Conference on Acoustics, Speech and Signal Processing, 2006

[17] M. Rideau, "Critère de sécurité des algorithmes de chiffrement à clé secrète", Thèse de Doctorat, Université de Paris 6, (France), 10. Novembre. 2005.

[18] F. OMARY, "Applications des algorithmes évolutionnistes à la cryptographie ", Thèsede Doctorat d'état, Université MOHAMMED V – AGDAL RABAT (MAROC), 26. Juillet. 2006.

[19] A. K. BENHAOUA, "Approche cryptographie base sur les algorithmes

[20] 2génétique pour la sécurité des réseaux Ad hoc ", Mémoire de Magistère, Université D'ORAN ES-SENIA, (Algérie), 2005.

[21] http://nopb.chez.com/crypto2.html

[22] J. Daemen, V. Rijmen, "AES, Proposal: The Rijndael Block Ciphe.Technical report, Proton World Int. l," Katholieke Universite it Leuven, ESAT-COSIC, Belgique, 2002.

[23] El Khier Dehmeche, "étude et comparaison des principaux systèmes de cryptage, " diplôme de magister en informatique, université M'sila, 2006.

[24] T. El Gamal, "A public key cryptosystem and signature scheme based on discrete logarithms." IEEE IT- 31, 496-473, 1976.

[25] B. Kebir, S. Rahmouni, "Développement d'une application pour l'échange des messages sécurisé ", Mémoire de fin d'études, Université de AbouBakr Belkaid, Tlemcen, Mai 2015.

[26] G. Zaidi, "Sécurisation par dynamiques0 chaotiques des réseaux locaux sans fil au niveau de la couche MAC ", Thèse de Doctorat, L'École Nationale d'Ingénieurs de Sfax, (Tunisie), 06.Décembre. 2012.

[27] S. Bekhouch, " Fondements mathématiques et fonctionnement du standard de chiffrement avancé Rijndael (AES) ", Mémoire de Magister, Université des Sciences et de la Technologie Houari Boumediene, 2006.

[28] http://sic.epfl.ch/publications/FI00/fi-sp-00/sp-00-page8.html http://csrc.nist.gov/CryptoToolkit/aes/http://www.uqtr.ca/~delisle/Crypto/prives/blocs_modes.phphttp://www.cs.ucdavis.edu/~rogaway/papers/ctr.pdf,

[29] http://www.bibmath.net/crypto

[30] National Institute of Standards and Technology: Data Encryption Standard (DES). http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf, (2001).

[31] http ://eprint.iacr.org/2009/374

[32] A.M'hamed, «systèmes cryptographiques»,.Département réseaux et services, Institut National des télécommunications.

[33] B,Jerome.https://briot-jerome.developpez.com/matlab/tutoriels/introduction-programmation-interfaces-graphiques/#LII-B. http://www.developpez.net/forums/u125006/dut/. [En ligne] 1 6 Publié le 1er juin 2007 - Mis à jour le 30 décembre 2013. [Citation : 30 12 2013.

[34] AU - Reddy, AR. T1 - Implementation of 128-bit AES algorithm in MATLAB. https://www.researchgate.net/publication/301227113_Implementation_of_128-bit_AES_algorithm_in_MATLAB/citation/download. [En ligne] 1 3 2016.

[35] https://www.math.univ-paris13.fr/~boyer/enseignement/PolyCrypto2010.pdf.[Enligne]

[36] http://www.bibmath.net/crypto/index.php?action=affiche&quoi=moderne/blocs.

[37] Z.Wang A.C. Bovik, H. R. Sheikh, and E .P. Simocelli, "Image qualityasssessment.