République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de L'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

Université **M**ouloud **M**ammeri De **T**izi-**O**uzou

Faculté De **G**énie **E**lectrique Et D'informatique
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Spécialité Automatique et Systèmes

Présenté par
Mohand-Amokrane BIR
Lyes DAHMOUNI

Mémoire dirigé par **Saïd DJENNOUNE** et co-dirigé par **Sarah KASSIM**

Thème

Etude et implémentation d'algorithmes de chiffrement à clé secrète et à clé publique : Application au cryptage de la parole.

Mémoire soutenu publiquement le 4 juillet 2018 devant le jury composé de :

M Redouane KARRA
professeur, UMMTO, Président

M Saïd DJENNOUNE
Professeur, UMMTO, Rapporteur

Mme Sarah KASSIM
Docteur, UMMTO, Rapporteur

M Hamid HAMICHE
MC-A, UMMTO, Examinateur

Mme Zedjiga YACINE
MC-A, UMMTO, Examinateur

Remerciements

Ce mémoire n'aurait pas pu être ce qu'il est sans l'aide de ALLAH le tout puissant qui nous a donné la force et le courage afin de l'accomplir.

Nous tenons à exprimer nos profonds remerciements et notre vive reconnaissance et considération à notre promoteur. **professeur DJENNOUNE**, qui a su, à sa façon, nous conseiller et nous orienter tout au long de la réalisation de ce travail.

merci à Mlle Kassim Sarah qui nous a encouragé tout au long de ce travail.

Nos remerciements particuliers sont adressés au cadre pédagogique de nous avoir consacré de leur temps et nous avoir apporté de l'aide par leurs précieux conseils.

Que toute personne ayant participé de près ou de loin dans l'élaboration de ce travail, trouve ici l'expression de notre vive reconnaissance.

Dédicace

je dédie ce travail à toutes ma famille qui ma soutenu et encourager dans touts mes décisions et durant tout mon cursus. Et à tous mes amis et toutes mes connaissances d'avoir étaient toujours présents pour moi.

Mohand – Amokrane

je dédie ce travail à toutes ma famille qui ma soutenu et encourager dans touts mes décisions et durant tout mon cursus. Mes sentiments vont également à toutes les personnes qui ont contribué de prés ou de loin à la réalisation de ce travail.

Lyes

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Depuis l'antiquité le problème de l'homme est la sécurité. Le développement de l'informatique et des télécommunications ainsi que la généralisation élargie des communications par internet ont accentués la complexité des problèmes de sécurité et de leurs solutions. En effet de nouveaux phénomènes en résultant telles les virus informatiques, accès non autorisés aux données, fausses informations, engendrent un besoin impérieux de sécurisation des informations et des technologies associées.

A l'heure actuelle, nous assistons à une évolution constante des techniques visant à sécuriser l'échange de données pour faire face aux différentes menaces. Pour répondre aux exigences de la politique de sécurité, il faut mettre en œuvre des systèmes de sécurité adéquats et robustes.[5]

La construction d'un système de sécurité fait appel inévitablement aux notions de cryptologie qui recouvrent la cryptographie et la cryptanalyse. Les cryptosystèmes modernes reposent sur le principe de Kerchkoffs, qui affirme que le secret ne doit pas résider dans l'algorithme mais plutôt dans la clé.

De nombreux systèmes de codage qui répondent à ce principe ont été proposés. Parmi les classes de ces systèmes nous pouvons citer les algorithmes de chiffrement symétrique (DES, IDEA, AES, ...) qui reposes sur le secret de la clé et les algorithmes de chiffrement asymétriques (RSA, ...) qui se basent sur la difficulté de factoriser les grandes nombres. L'émergence des nouvelles technologies de l'information, de la télécommunication et de la mondialisation des échanges ont donné naissance à de nouvelles approches pour subvenir aux contraints de sécurité. Nous pouvons citer le chiffrement quantique, basé sur les lois de la physique quantique et le chiffrement chaotique qui ce repose sur la maitrise de phénomène chaotique.

Les travaux réalisés dans ce mémoire ont pour objectif d'étudier et d'implémenter deux systèmes de cryptage/décryptage, l'un est symétrique et l'autre asymétrique. Le deuxième objectif est d'appliquer le chiffrement symétrique sur le signal de la parole et de proposer un structure hybride. Pour cela, nous avons organisé le mémoire comme suit :

Dans le première chapitre, nous avons présenté des définitions et des généralités sur des notions telles que l'intégrité, l'authentification, les menaces informatiques...etc. Ce chapitre traite des techniques cryptographiques qui ont marqué l'histoire, suivis par les techniques actuelles du monde de l'informatique. Il décrit les trois principales méthodes de chiffrement qui sont : le chiffrement classique, le chiffrement moderne et le chiffrement quantique. Le chapitre aborde aussi les modes de chiffrement de l'information.

Le chapitre 2 est dédié à la présentation des fondements théoriques de chiffrement à clé secrète DES et le chiffrement à clé publique RSA. Le chapitre abord les problèmes mathématiques sous-jacents aux cryptosystèmes à clé publique comme la factorisation des entiers, et les fonctions a sens unique. Il décrit ainsi certains aspects de chiffrement à clé secrète DES.

Le chapitre 3 est consacré au cryptage du signal parole. Ce chapitre décrit d'une part certaines performances de l'oreille, et les fondements de la numérisation des signaux audio, défini le signal parole en citant ces différentes caractéristiques. L'autre part est réservée pour l'approche proposé de chiffrement hybride (DES-RSA) sur le signal parole.

Chapitre 1

Généralités sur la cryptographie

1.1 Introduction

La transmission de l'information est omniprésente dans les télécommunications. L'information peut être de n'importe quel type de données numériques : textes, images, vidéos, son... .

La transmission de l'information commence par sa formulation par un émetteur qui sera poursuivi par un transit via un canal de transmission et se termine par le destinataire qui va reconstituer l'information.

Les canaux de transmissions peuvent être de n'import quel types (réseaux de câbles ou d'ondes). Or quel que soit le canal, il ne peut jamais être considéré sûr dans plusieurs sens du terme, d'où l'information est susceptible d'être lu ou altéré.

Partant du principe que les canaux numériques ne sont pas sûres et que l'information peut être interceptée au cours de sa transmission, il faut transformer le texte et le decorréler de sa signification. Pour cela, il faut entourer la transmission d'une confidentialité qui garantisse le secret de l'information pour faire face aux menaces qui peuvent nuire à l'efficacité, la sécurité et l'intégrité de l'information.

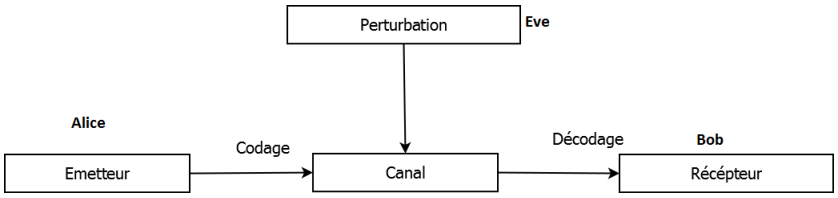

FIGURE 1.1 – Circuit de transmission

1.2 Sécurité de l'information et la cryptographie

1.2.1 La cryptographie

C'est l'art de rendre inintelligible, de crypter, coder ou cacher le sens d'un message à tous ceux qui ne sont pas autorisés à le connaitre. C'est aussi l'étude des techniques mathématiques liées aux aspects de la sécurité de l'information tels que la confidentialité, l'authentification des entités d'intégrité des données et l'authentification de l'origine des données.

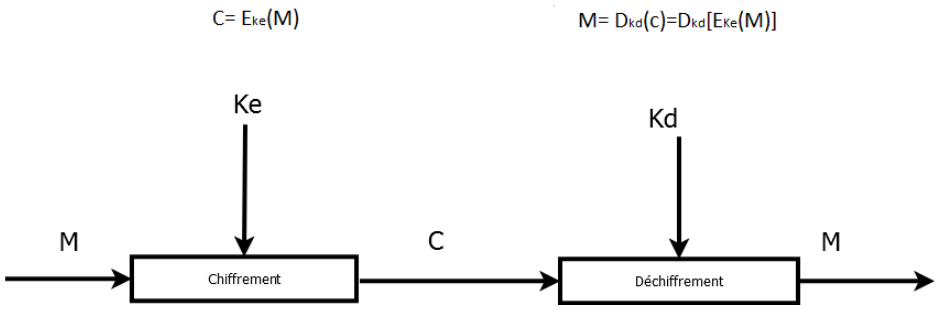

FIGURE 1.2 – Système cryptographique.

1.2.2 Historique de la cryptographie

Dès que les hommes apprirent à communiquer, ils durent trouver des moyens d'assurer la confidentialité de leurs communications c'est sans doute l'origine de la cryptographie. Vers 600 ans avant J.C, le roi de Babylon (Nabuchodonosor) écrivait ses messages sur le crane de ses esclaves, il attendait que leurs cheveux repoussent pour les envoyer, est pour lire le message il suffit juste des raser a nouveau pour les esclaves. Dans l'antiquité, un dispositif appelé syctale fait le cryptage des communications grecs. Ce dispositif est une bande étroite de parchemine sur laquelle il écrivait après l'avoir enroulé en spirale autour d'un cylindre de bois. Pour lire le message il suffisait d'enrouler la bande sur un cylindre de même diamètre.

50 ans avant J.C, Jules César pour masquer ses messages, utilisait une substitution alphabétique c'est à dire chaque lettre est remplacée par une autre lettre de l'alphabet, décalée d'une quantité fixe. Sa faiblesse réside dans le nombre de façons de chiffrer le message mais l'alphabétisation de la faible population le rend efficace. On assiste au développement plus ou moins ingénieux des techniques de chiffrement expérimentales dont la sécurité reposait essentiellement sur la confiance que leur accordaient leurs utilisateurs.

Plus tard, en 1467, Leone Battista propose la méthode de substitution polyalphabé-

tique c'est à dire de remplacer chaque lettre d'un autre alphabet. Vers les année 1500, une procédure de remplacer une lettre par un groupe de mots a été proposée par l'abbé Jean Tritheme.

L'inconvénient de la substitution alphabétique est la fréquence d'apparition de chaque lettre, c'est-à-dire, on peut décrypter le message par une attaque statistique même si la substitution alphabétique a été améliorée par Vigéner en 1586 par l'utilisation d'une clé littérale. Cependant, la non sécurisation de la clé reste un inconvénient majeur qui peut conduire au décodage du message.

En 1918, l'Allemand Arthur Scherbius a donné naissance à la fameuse machine ''Enigma". Le principe fut que chaque lettre est remplacée par une autre lettre, la règle de substitution avec cette machine est changée à chaque lettre. Cette procédure nous permet d'évincer le problème de fréquence ainsi le problème de Vigéner. Le développement de l'électronique ainsi que l'apparition des ordinateurs puissants et le développement des techniques de communications, ont fait que la sécurité de l'information devienne un nouveau problème non seulement pour la confidentialité mais pour préserver le contenu des messages est assurer l'identité de l'émetteur et du récepteur.

En 1970, Horst Feistel de compagnie IBM, propose un projet de recherche qui consiste à trouver de nouvelles méthodes de chiffrement. Des efforts de travail ont conduit à l'élaboration du DES (Data Encryption Standard). En 1976, Whit Field et Martin Helmman proposent la cryptographie à clé publique. En 1978 les trois mathématiciens américaine Rivest, Shamir, et Adleman propose le système de chiffrement à clé publique RSA ce qui a mené à l'explosion des applications civiles de chiffrement. Ces deux derniers algorithmes de chiffrement à clé publique et à clé secrète font révolutionner le monde de la cryptographie à nos jours.

1.2.3 Terminologie

— Texte Clair : L'information que Alice souhaite transmettre.
— Texte chiffré : Résultat de chiffrement.
— Chiffrement : C'est l'algorithme utilisé pour rendre le texte clair incompréhensible.
— Déchiffrement : C'est la méthode utilisé ou l'algorithme pour reconstruire le texte clair.
— Clé : C'est le secret partagé utilisé pour le chiffrement et le déchiffrement.
— Cryptographie : C'est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné .
— Cryptanalyse : Elle a pour but de retrouver le texte clair a partir de texte chiffrés

en déterminants des failles des algorithmes utilisés.

— Cryptologie : Une science mathématique qui comporte deux branches : La cryptographie et la cryptanalyse.

— Coder, décoder : C'est une méthode ou un algorithme permettant de modifier la mise en forme d'un message sans sans introduire d'élément secret.[1]

1.2.4 Buts de la cryptographie

On désigne par la sécurité informatique l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.[2]

La notion de la sécurité fait référence à la propriété d'un système, d'un service, d'une entité. Elle s'exprime par les objectifs de sécurité résumés dans le sigle CAIN, pour Confidentialité, Authentification, Intégrité, Non-répudiation.

Confidentialité

Seuls les utilisateurs autorisés peuvent accéder à l'information. Pour assurer la confidentialité des données, deux actions complémentaire sont à appliquer :[2]

1. Limiter et contrôler l'accès aux données.
2. Transformer les données par des techniques de chiffrement pour qu'elles deviennent intelligibles.

Dans le cas du chiffrement à clé privée, une même clé est utilisée pour le chiffrement et le déchiffrement. Ce type de chiffrement nécessite un échange sûr de clé entre A et B.

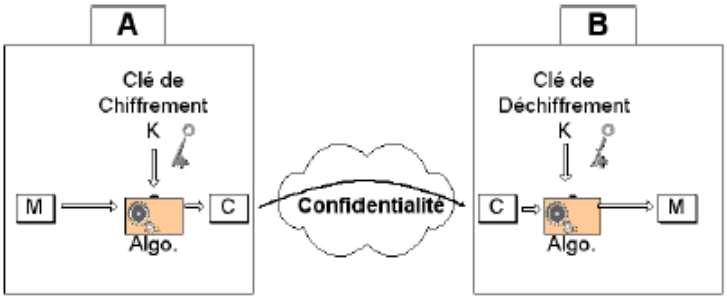

FIGURE 1.3 – Confidentialité d'un système à clé privée.

Dans le cas d'un chiffrement à clé publique, chaque entité à sa propre paire de clés.

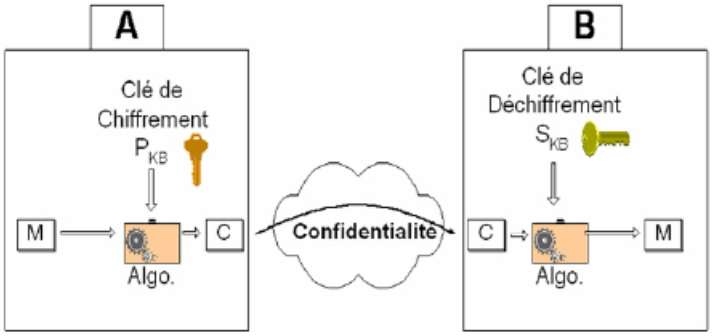

FIGURE 1.4 – Confidentialité d'un système à clé publique.

Dans le chiffrement hybride, on utilise le chiffrement à clé privée pour chiffrer le message. Par l'intermédiaire du système à clé publique, on sécurise l'échange de la clé.

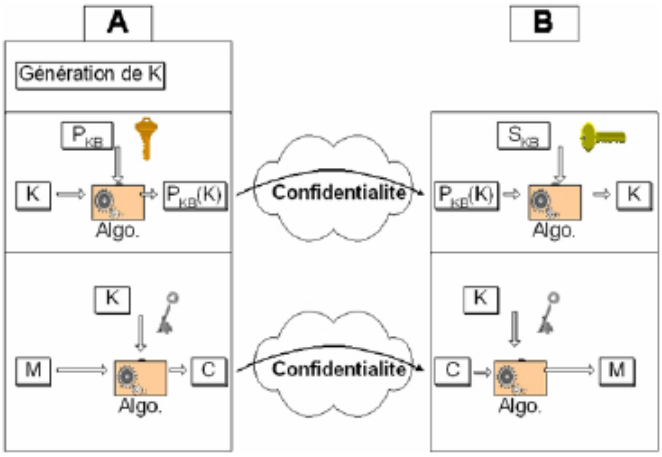

FIGURE 1.5 – Confidentialité d'un système hybride.

Intégrité

Seuls les utilisateurs autorisés peuvent modifier l'information. d'où la nécessité de vérifier si le message n'a pas subit de modifications durant la communication.

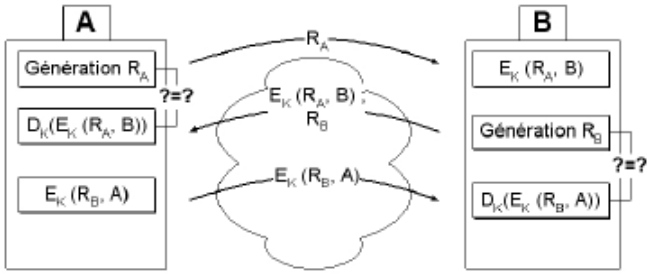

FIGURE 1.7 – Authentification dans un système à clé privée .

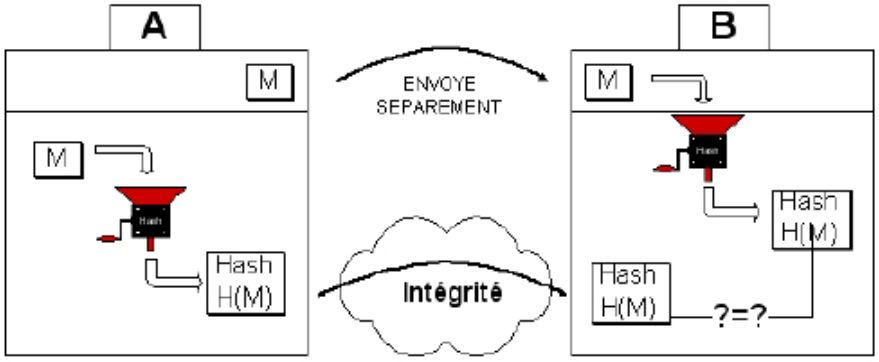

FIGURE 1.6 – Vérification de l'intégrité par fonction de hachage.

Authentification

C'est la propriété qui consiste à vérifier l'identité d'une entité avant de lui donner l'accès à une ressource. L'entité doit prouver son identité. Tous les mécanismes de contrôle d'accès logiques aux ressources informatiques nécessitent de gérer l'identification et l'authentification.[2]

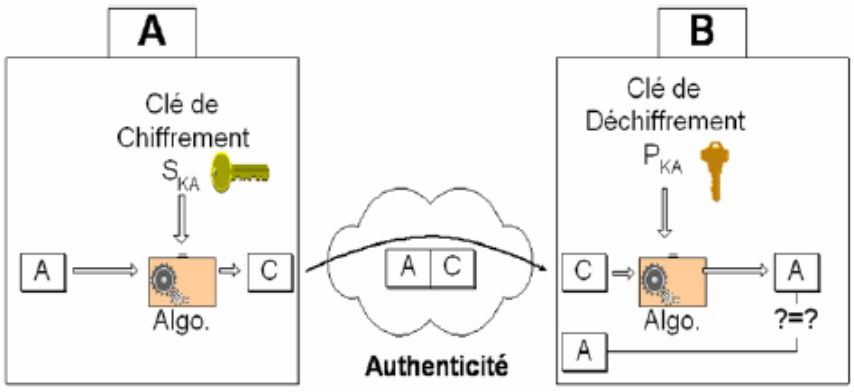

FIGURE 1.8 – Authentification dans un système à clé publique.

Non-répudiation

C'est le fait de ne pas pouvoir nier qu'un évènement (action, transaction) a eu lieu. Elle contient :[2]

1. **-Non-répudiation d'origine** : L'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est effectivement le cas.
2. **-Non-répudiation de réception** : Le receveur ne peut nier avoir reçu le message et il peut prouver qui'il ne l'a pas réutilisé si c'est effectivement le cas.
3. **-Non-répudiation de transmission** : L'émetteur du message ne peut nier avoir envoyé le message et il peut prouver qu'il ne l'a pas fait si c'est effectivement le cas.

1.3 Chiffrement symétrique

Le chiffrement symétrique consiste à utiliser la même clé pour le chiffrement et le déchiffrement. Autrement dit Alice et Bob conviennent secrètement d'une clé secrète K qui est donc utilisé à la fois pour le chiffrement et le déchiffrement. Il conviennent également d'un algorithme cryptographique de chiffrement et de déchiffrement. De tels systèmes ont l'avantage principal d'être efficaces en terme de temps de calcul, tant pour le chiffrement que pour le déchiffrement. En revanche, la faiblesse de ce système vient du secret absolu qui doit entourer la clé K. Il s'agit historiquement du premier type de chiffrement utilisé. Plusieurs exemples de ce type (le chiffrement de César et le chiffrement parfait de Vernam) .[3]

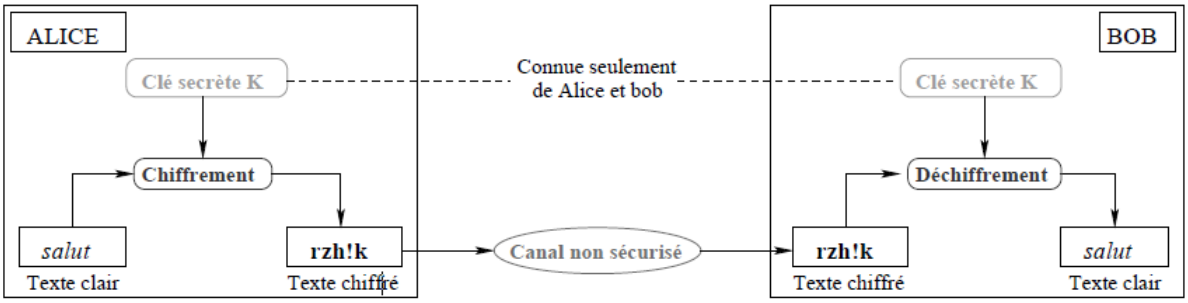

FIGURE 1.9 – Chiffrement symétrique.

1.3.1 Cryptosystème classique

Fondamentalement, la substitution et la transposition sont les deux méthodes de chiffrement. La substitution consiste à remplacer un caractère par un autre. La transposition

ou confusions consiste à mélanger l'ordre des caractères. A base de ces deux techniques, de nombreux algorithmes de cryptage plus ou moins complexes ont été développés.

Substitution mono-alphabétique

Chaque caractère du message en claire est remplacé par le caractère correspondant en alphabet de substitution.

substitution simple

Chaque lettre du message claire est remplacée par le même caractère. On obtient alors une bijection entre les lettres est les caractères de l'alphabet de chiffrement.

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

TABLE 1.1 – substitution simple.

message à chiffrer	message chiffrée
AUTOMATIQUE	DXWRPDWLTXH

TABLE 1.2 – Exemple de chiffrement pas substitution simple.

Dans l'exemple ci-dessus (chiffrement de César) on décale les lettres de 3 positions. Le chiffrement de César est un cas particulier de chiffrement mono-alphabétique.

Substitution homophonique

Le principe est le même avec la substitution simple. La différence est que la transformation n'est plus bijective. Une lettre peut correspondre à plusieurs caractères.

Substitution polyalphabétique

Le principe de la substitution polyalphabétique est d'utiliser plusieurs tables de substitutions simples. Pour chiffrer le message on change de table à toutes les lettres, et on boucle selon le nombre de tables. Par exemple si ont veut chiffrer le mot AUTOMATIQUE, on prend les trois tables C_1, C_2, C_3 les alphabets de substitution et C étant l'alphabet clair , voir la table 1.3 .

C	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
C_1	o	n	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
C_2	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
C_3	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

TABLE 1.3 – Les tables de substitution.

Dans cet exemple on boucle tous les 3 caractères, on obtient le message chiffré : NBNBTUHPKHL.

Substitution polygrammique

Elle consiste à substituer un groupe de caractères dans le message par un autre groupe de caractères. L'algorithme de Playfair est la première technique utilisable en pratique de chiffrement polygrammique.

Le chiffrement de Playfair ou carré de Playfair est inventé par Charles Weatstone en 1854 et est popularisé par Lord Playfair. Cet algorithme consiste à chiffrer des paires de lettres partir d'une matrice 5×5 construite avec une clé choisie. Le remplissage de la table ce fait avec la clé et le reste de l'alphabet dans l'ordre (i et j occupe une seule case), le mot clé utilisé peut s'écrire en ligne ou en colonne ou en spirale.

Pour chiffrer un message donné avec le carré de Playfair on doit appliquer les règles suivantes :

1. Si deux lettres sont identiques ou bien si il ne nous reste que une seul lettre à chiffré, elle sera remplacée par "X" ou "Q".
2. Si les deux lettres sont sur la même ligne, elles seront remplacées par celle de droit directement et si on atteint l'extrémité on doit boucler par la gauche.
3. Si les deux lettres sont sur la même colonne elles seront remplacées par celle d'en dessous et en bouclant par le haut si l'extrémité est atteinte.
4. Sinon, les lettres seront remplacées par d'autres situées sur la même ligne est dans un emplacement opposé.[4]

*	*	*	*	*
*	*	*	*	*
*	A	Z	R	T
*	*	*	*	*
*	*	*	*	*

*	*	*	*	*
A	*	*	*	*
B	*	*	*	*
C	*	*	*	*
D	*	*	*	*

Z	*	*	O	*
*	*	*	*	*
*	*	*	*	*
R	*	*	X	*
*	*	*	*	*

FIGURE 1.10 – Règles a appliquer pour le chiffrement Playfair.

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

TABLE 1.4 – Table de chiffrement de Playfair.

message clair	message chiffré
AU|TO|MA|TI|QU|E	MX|PR|OR|SK|LW|Q

FIGURE 1.11 – Résultat de chiffrement.

Transposition

Les algorithmes de transposition consistent à réordonner les caractères sur la base d'une permutation, en générale ils opèrent sur des blocs avec une période "d" fixée ou $\mathbb{Z}$ et l'ensemble des entiers de 1 à "d" et "f" est la permutation. La paire $K = (d,\ f)$ est la clé de l'algorithme. Les blocs sont chiffrés en permutant les caractères selon f.[11]

G	R	A	I	N
2	5	1	3	4
S	A	L	U	T
L	E	S	P	E
T	I	T	S	P
O	T	S	(A)	(B)

A	G	I	N	R
1	2	3	4	5
L	S	U	T	A
S	L	P	E	E
T	T	S	P	I
S	O	(A)	(B)	T

FIGURE 1.12 – Application d'une transposition.

1.3.2 Cryptosystèmes moderne

1.3.2.1 Schéma de Feistel

La structure de Feistel fut décrit en 1973 par Horst Feistel (employé chez IBM). Dans une construction de Feistel dans la figure 1.13, le bloc d'entrée d'un round est séparé en deux parties. La fonction de chiffrement est appliquée sur la première partie du bloc et l'opération binaire OU-Exclusif ($\oplus$) est appliquée sur la partie sortante de la fonction et la deuxième partie. Ensuite les deux parties sont permutées et le prochain round commence.[11]

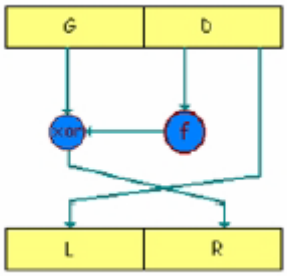

FIGURE 1.13 – Structure de Feistel.

L'avantage est que la fonction de chiffrement et la fonction de déchiffrement sont identiques. Ainsi la fonction n'a pas à être inversible, c'est la structure qui l'est.

1.3.2.2 Chiffrement par blocs

L'opération de chiffrement s'effectue sur des blocs de texte clair. L'idée générale d'un chiffrement par bloc est la suivante :

1. Remplacer les caractères par un code binaire.
2. Découper la chaine en blocs de langueur donnée.
3. Chiffrer un bloc en l'additionnant bit par bit à une clé.
4. Déplacer certains bits d'un bloc.
5. Recommencer éventuellement un certain nombre de fois l'opération 3.
6. Passer au bloc suivant et retourner au point 3 jusqu'à ce que tout le message soit chiffré.

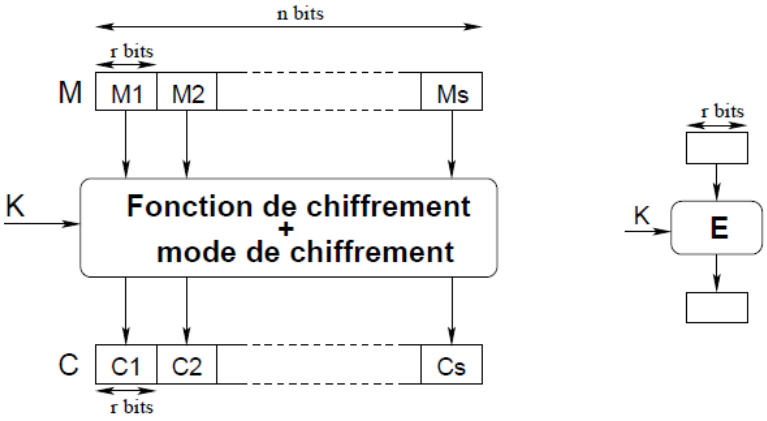

FIGURE 1.14 – Chiffrement symétrique par blocs.

1.3.2.3 Modes de chiffrements

Que ce soit pour le DES ou les nouveaux standards IDEA, AES les clés sont de longueur fixée. Or, les messages peuvent être de longueur quelconque bien sûr. il faut donc initier des chiffrements par blocs de taille fixe correspondants aux taille des clés. Pour cela, quatre modes de chiffrement par blocs sont possibles : ECB, CBC, CFB, OFB.

Mode ECB (Electronic Code Book)

Le mode ECB est le plus simple parmi les modes de chiffrement, il s'agit de découper le message M en blocs m_i de taille fixe. Dans ce mode chaque blocs est crypté séparément

$$c_i = E_k\left(m_i\right) \tag{1.1}$$

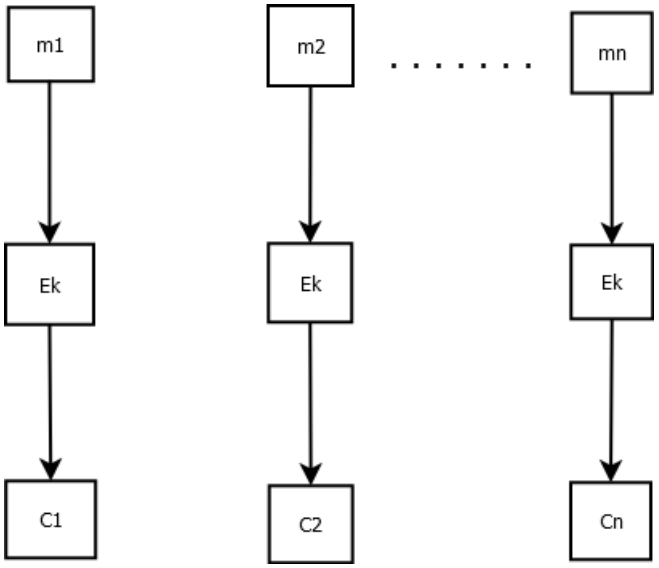

FIGURE 1.15 – Mode ECB : Electronic Code Book.

Ainsi, un bloc de message donné mais sera toujours codé de la même manière. Ce mode de chiffrement ne présente donc aucune sécurité et n'est jamais utilisé ![5].

Avantages du mode ECB :

Le travail de chiffrement ou de déchiffrement peut être parallélisé. Plusieurs machines ou CPU peuvent travailler simultanément sur des parties différentes du message. Il permet un accès aléatoire dans le texte chiffré. Une erreur de transmission d'un bit affecte uniquement le décodage du bloc courant.

Inconvénients :

Les répétitions du texte en clair ne sont pas masquées et se retrouvent sous la forme de répétitions de textes chiffrés. Des portions complètes du message peuvent être modifiées, répétées ou remplacées sans difficulté. La perte de synchronisation (perte ou ajout d'un bit) est irrécupérable.

Mode CBC (Cipher Bloc Chaining)

Pour que le message m_i ne soit pas codé de la même manière le mode CBC ajout une valeur initiale C_0, chaque blocs et modifié par XOR avec le bloc crypté précédent avant d'être lui-même crypté par :

$$C_i = E_k\left(m_i \oplus C_{i-1}\right) \tag{1.2}$$

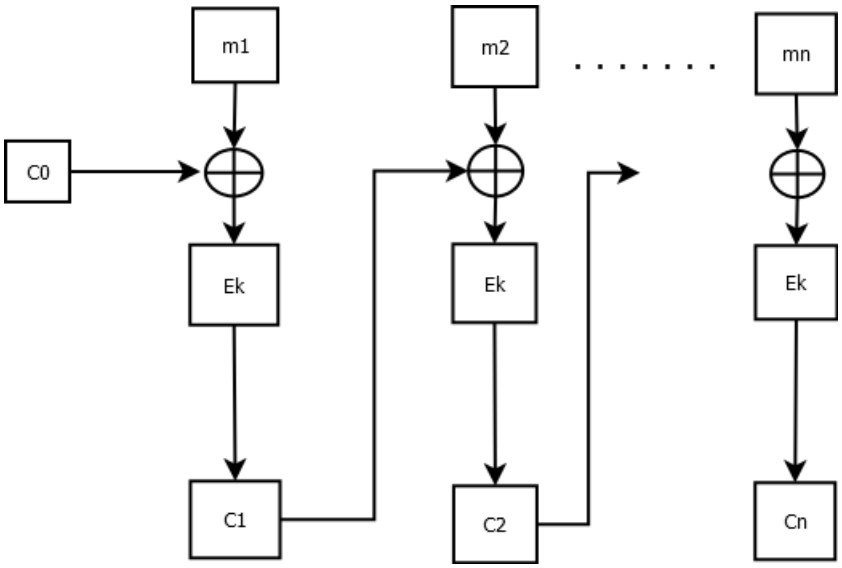

FIGURE 1.16 – Mode CBC : Cipher Bloc Chaining.

C'est le mode de chiffrement le plus utilisé. Le déchiffrement nécessite l'inverse de la fonction de codage $D_{(k)} = E^{-1}$ pour décrypter[5] :

$$m_i = C_{(i-1)} \oplus D_k(c_i) \tag{1.3}$$

Avantages du mode CBC :

Les répétitions de texte en clair sont masquées dans le texte chiffré. La valeur du vecteur d'initialisation IV n'a pas besoin d'être secrète.

Inconvénients :

Deux textes en clair commençant pareil auront le même début de texte chiffré. Une erreur de transmission d'un bit affecte uniquement le décodage du bloc courant ainsi que le décodage du même bit dans le bloc suivant. La perte de synchronisation (perte ou ajout d'un bit) est irrécupérable

Mode CFB (Cipher FeedBack)

Dans le CFB on utilise un XOR après le cryptage pour éviter la fonction inverse pour le décryptage

$$C_i = m_i \oplus E_k\left(C_{i-1}\right) \tag{1.4}$$

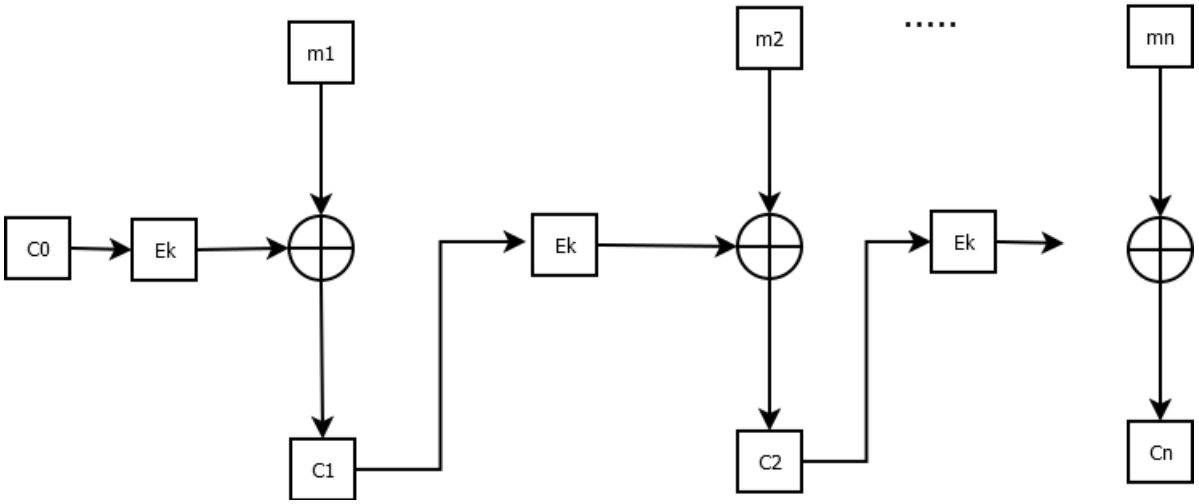

FIGURE 1.17 – Mode CFB : Cipher FeedBack.

Ce mode est moins sûr que le CBC et est utilisé par exemple pour le cryptage de réseau. L'intérêt est que le déchiffrement ne nécessite pas de fonction de déchiffrement D. [5]

$$m_i = C_i \oplus E_k(C_{(i-1)}) \tag{1.5}$$

Avantages du mode CFB :

Il est possible de chiffrer un flot de valeurs plus petites que la taille standard du bloc géré par l'algorithme. Les répétitions de texte en clair sont masquées dans le texte chiffré. La valeur du vecteur d'initialisation IV n'a pas besoin d'être secrète. La perte de synchronisation (perte ou ajout d'un bit) est récupérable.

Inconvénients :

Une erreur de transmission d'un bit affecte uniquement le décodage du bloc courant ainsi que le décodage du même bit dans le bloc suivant.

Mode OFB (Output FeedBack)

Ce mode est pour avoir un cryptage et un décryptage totalement symétrique[5].

$$Z_i = E_k\left(Z_{i-1}\right) : C_i = m_i \oplus Z_i \tag{1.6}$$

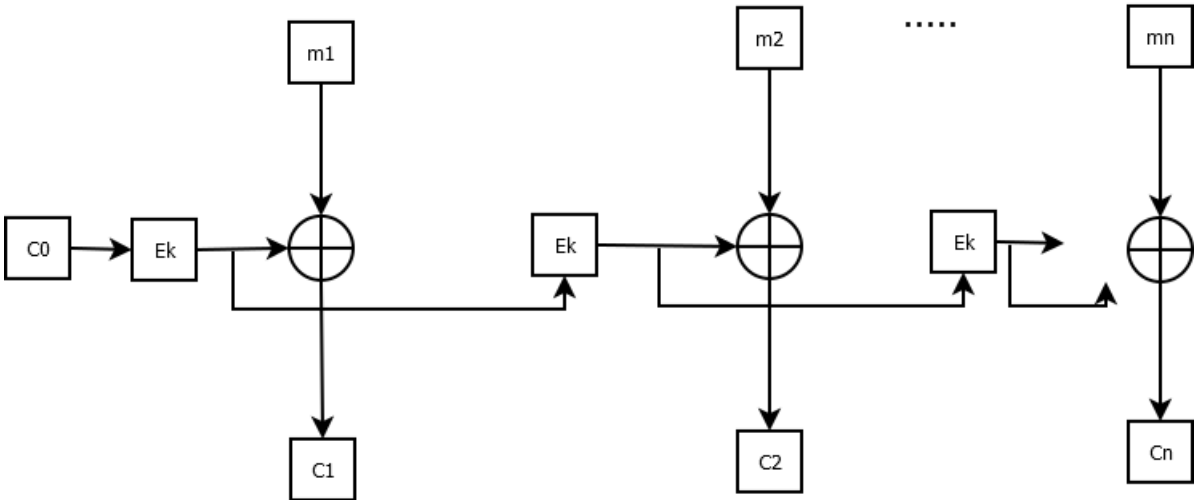

FIGURE 1.18 – Mode OFB : Output FeedBack.

Avantages du mode OFB :

Les répétitions de texte en clair sont masquées dans le texte chiffré. La valeur du vecteur d'initialisation IV n'a pas besoin d'être secrète. Ce mode n'amplifie pas les erreurs. Une erreur de transmission d'un bit affecte uniquement ce bit lors du décodage.

Inconvénients :

La perte de synchronisation (perte ou ajout d'un bit) est irrécupérable.

Mode CTR (counter-mode-encrption)

Ce dernière et un mode totalement symétrique,mais en outre facilement parallélisable, il utilise le chiffrage d'un compteur de valeur initiale T,mais en outre facilement.[5]

$$C_i = m_i \oplus E_k\left(T + i\right) \tag{1.7}$$

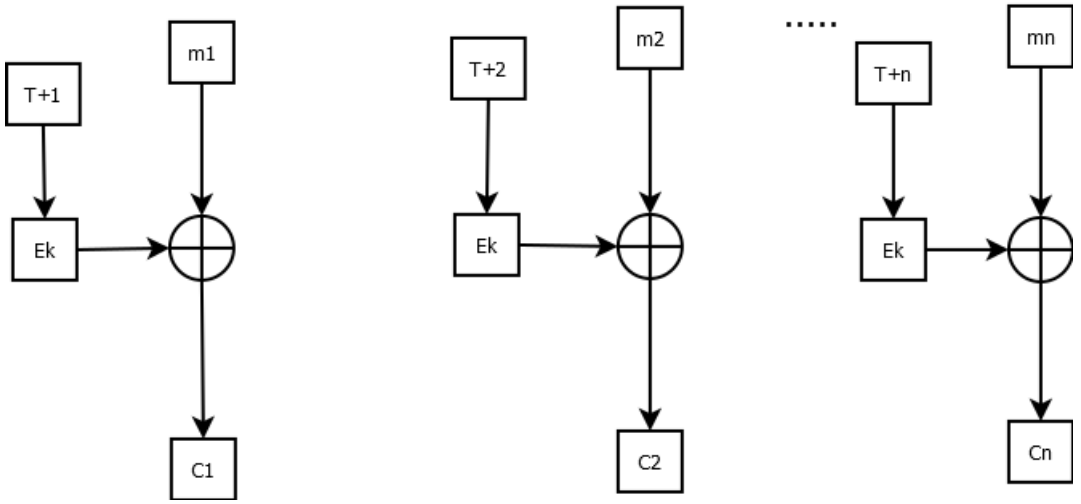

FIGURE 1.19 – Mode CTR :counter-mode-encrption.

$$m_i = C_i \oplus E_k\,(T + i) \tag{1.8}$$

L'intérêt d'un tel mode est principalement que les différents calculs de cryptage et décryptage sont indépendants,comme pour le EBC mais qu'un même bloc n'est a priori jamais codé de la même façon.[5]

1.3.2.4 Chiffrement par flots

On chiffre un bit/caractère à la fois, la structure d'un chiffrement par stream repose sur un générateur de clé qui produit une séquence de clés K_1.....K_i.

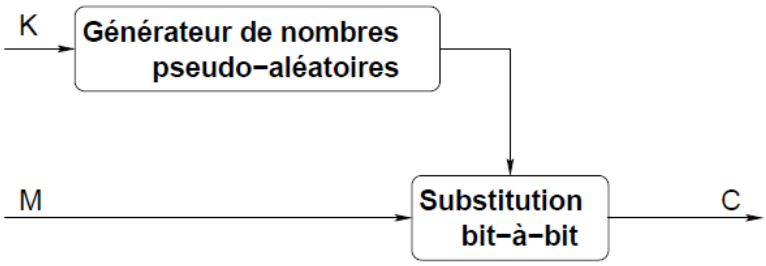

FIGURE 1.20 – Chiffrement symétrique par flot.

1.3.2.5 Comparaison des chiffrements par blocs et par flots

	par Blocs	par Flots
Avantages	- Réutilisation des clés	Moins de code d'implémentation - Rapidité
Inconvénients		- Deux utilisations d'une même clé facilite la cryptanalyse
Applications	- Transfert de fichiers	- Chiffrement de canal de communication

TABLE 1.5 – Comparaison de chiffrement par bloc et par flot.

Avantages de chiffrement symétrique

— Rapidité
— Facilité d'implémentation sur hardware
— Taille de clé mémorisable

Inconvénients de chiffrement symétrique

— Partage de clé
— Nombre de clés à gérer
— Signature difficile à réaliser

1.4 Chiffrement asymétrique

Dans le cas des systèmes symétriques, on utilise une même clé pour le chiffrement et le déchiffrement. Le problème repose dans la transmission de la clé : il faut une clé par destinataire. Dans le cas des systèmes asymétriques, chaque cryptosystème possède 2 clés distinctes (une privée, une publique) avec une grande difficulté de déduire la clé privée à partir de la clé publique. De ce fait, il est possible de distribuer librement cette dernière.[11]

Ce système est basé sur une fonction facile à calculer dans un sens (appelée fonction à trappe à sens unique ou en anglais one-way trapdoor function) et mathématiquement très difficile à inverser sans la clé privée (appelée trappe). Cette méthode rudimentaire va à l'encontre du principe de Kerchkoffs qui s'énonce ainsi : « La sécurité d'un système de chiffrement ne doit reposer que sur la clé. »

Cela se résume aussi par : « L'ennemi peut avoir connaissance du système de chiffrement. »

Voici le texte original d'Auguste Kerchkoffs de 1883 « La cryptographie militaire » paru dans le Journal des sciences militaires. Il traite notamment des enjeux de sécurité lors des correspondances :

« **Il faut distinguer un système d'écriture chiffré, imaginé pour un échange momentané de lettres entre quelques personnes isolées et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux.** » Le principe fondamental est le suivant :

> « Dans le second cas, [. . .] il faut que le système n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi. » Ce principe est novateur dans la mesure où intuitivement il semble opportun de dissimuler le maximum de choses possibles : clé et système de chiffrement utilisés. Mais l'objectif visé par Kerckhoffs est plus académique, il pense qu'un système dépendant d'un secret mais dont le mécanisme est connu de tous sera testé, attaqué, étudié, et finalement utilisé s'il s'avère intéressant et robuste.

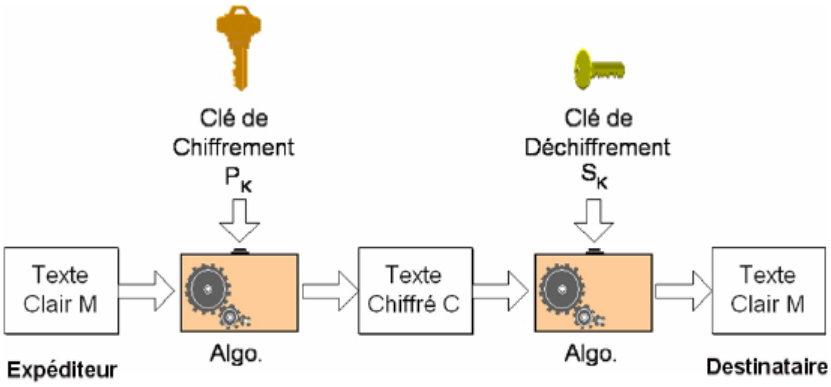

FIGURE 1.21 – Chiffrement asymétrique.

Le chiffrement asymétrique à été proposé en 1976 par Diffie et Hillman. Dans un tel schéma, la clé de chiffrement K^e et différente de celle de déchiffrement K^d. Le chiffrement asymétrique repose sur l'existence d'une fonction à sens unique. une fonction et dite à sens unique quand il est facile de calculer K^e connaissant K^d mais très difficile de calculer K^d connaissant K^e. Parfois, des fonctions à sens unique qui possèdent en plus une trappe sont utilisé. une fonction à sens unique est dite à trappe quand il est très difficile de calculer K^d à partir de K^e, sauf si on connaît une information supplémentaire.[1]

Pour que Alice veule communiquer avec Bob elle doit chiffrer le message M en texte chiffré $C = E(K^e, M)$, ou E est la fonction de chiffrement, De son coté Bot calcule $M = D(K^d, C)$ ou D est une fonction de déchiffrement.

Un exemple de chiffrement à clé publique le schéma RSA proposé par Rivest, Shamir et Adleman, en 1978 qui est largement utilisé. Il repose sur la difficulté de factoriser des grands nombres.

1.4.1 Outils mathématiques

1.4.1.1 Factorisations des entiers

La factorisation des entiers est un problème crucial en cryptographie, car elle permet de casser le RSA. La méthode la plus élémentaire pour factoriser un entier n consiste à prendre tous les entiers inférieurs à n, et à tester s'ils divisent n (algorithme de force brute). C'est bien sûr un algorithme inutilisable si n est grand. Un premier raffinement consiste à ne prendre que les entiers inférieurs à racine de n (si n n'est pas premier, n admet forcément un diviseur inférieur à racine de n). C'est beaucoup mieux, mais encore insuffisant pour les entiers de 1000 chiffres que l'on souhaite factoriser.

L'idée utilisée par les algorithmes modernes (crible quadratique, crible du corps de nombre) est due à l'arithméticien français Pierre de Fermat : si on trouve deux entiers x et y, non égaux, non opposés, tels que $x^2 = y^2 \, mod \, n$, alors $(x - y)(x + y) = 0$ mod n, et $pgcd\,(x + y, n)$ ou $pgcd\,(x - y, n)$ donne un diviseur non trivial de n

Il reste à trouver de tels nombres x et y. Posons x un nombre juste supérieur à racine de n. x^2 est juste supérieur à n, et $x^2 = amodn$, avec a petit. Il est donc facile de factoriser a, et avec un peu de chances, en essayant plusieurs x, on peut espérer trouver un a tel que $a = y^2$.

Ex : Soit à factoriser $n = 3337$. Sa racine carrée vaut57. On teste :

— $58^2 = 27 = 3^3$ mod 3337 : ne convient pas.

— $59^2 = 144 = 12^2$ mod 3337 : convient !

Alors, $pgcd\,(59 + 12, 3337) = 71$ donne un diviseur non trivial de n. Le second facteur premier est 49.

Bien sûr, dans le cas général, en prenant des x juste plus grands que racine de n, il n'est pas sûr que l'on tombe dans le cas précédent. Mais on peut combiner plusieurs équations : si $X_1^2 = a_1$ mod$n$.....X_p^2 mod n, et que a_1.....$a_p = y^2$ mod n, en posant $X = X_1$.....X_p, on retrouve $X^2 = Y^2 \, mod \, n$.

Ex : Soit à factoriser $n = 180121$, dont la racine carrée vaut 424, et des brouettes... En faisant des essais successifs avec les entiers juste supérieurs à 424, on trouve :

— $425^2 = 2^3 3^2 5^2 7 \, mod \, n$.

— $439^2 = 2^3 3^2 7^2 \, mod \, n$.

d'où $(425.439)^2 = (2^3 3^2 57)^2 \, modn$, et $pgcd(n, 425 \times 439 - 2^3 3^2 57) = 281$ donne un diviseur non trivial de n. Tout le problème réside ensuite dans le choix des bases de petits premiers sur lesquels on souhaite décomposer.[7]

1.4.1.2 Fonctions à sens unique

Il existe bien d'autres situations mathématiques asymétriques : les fonctions à sens unique. En d'autres termes, étant donnée une fonction f , il est possible connaissant x

de calculer «facilement» $f(x)$; mais connaissant un élément de l'ensemble image de f , il est «difficile» ou impossible de trouver son antécédent. Dans le cadre de la cryptographie, posséder une fonction à sens unique qui joue le rôle de chiffrement n'a que peu de sens. En effet, il est indispensable de trouver un moyen efficace afin de pouvoir déchiffrer les messages chiffrés. On parle alors de fonction à sens unique avec trappe secrète.

Prenons par exemple le cas de la fonction f suivante :

f : x$\rightarrow x^3\,(mod\,100)$

• Connaissant x, trouver $y = f\,(x)$est facile, cela nécessite deux multiplications et deux divisions.

• Connaissant y image par f d'un élément x ($y = f\,(x)$), retrouver x est difficile.

Tentons de résoudre le problème suivant : trouver x tel que $x^3 \equiv 11\,mod\,100$. On peut pour cela :

• soit faire une recherche exhaustive, c'est-à-dire essayer successivement 1, 2, 3, ..., 99, on trouve alors :

$71^3 = 357911 \equiv 11\,mod\,100$

soit utiliser la trappe secrète :$y \rightarrow y^7\,mod\,100$ qui fournit directement le résultat !

$11^7 = 19487171 \equiv 71\,mod\,100$

La morale est la suivante : le problème est dur à résoudre, sauf pour ceux qui connaissent la trappe secrète. (Attention, dans le cas de cet exemple, la fonction f n'est pas bijective.).[9]

1.4.1.3 Petit théorème de Fermat amélioré

Nous connaissons le petit théorème de Fermat :

Théorème 1.4.1 (Petit théorème de Fermat).[9] Si p est un nombre premier et$a \in \mathbb{Z}$ alors, et p ne dévise pas a .

$$a^p \equiv a\,mod\,p \tag{1.9}$$

et sa variante :

Corollaire 1.4.1[9] :

Si p ne divise pas a alors

$$a^{p-1} \equiv 1\,mod\,p \tag{1.10}$$

Nous allons voir une version améliorée de ce théorème dans le cas qui nous intéresse :

Théorème 1.4.2 (Petit théorème de Fermat amélioré).[9] Soient p et q deux nombres premiers distincts et soit $n = pq$. Pour tout $a \in \mathbb{Z}$ tel que $pgcd(a, n) = 1$ alors :

$$a^{(p-1)(q-1)} \equiv 1\,mod\,n \tag{1.11}$$

on note $\varphi(n) = (p-1)(q-1)$ la fonction d'Euler l'hypothèse $pgcd(a,n) = 1$ équivaut ici à ce que a ne soit divisible ni par p, ni par q. Par exemple pour $p = 5$ et $a = 7$ $\varphi(n) = 4.6 = 24$ Alors pour $a = 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18,$ ... on a bien

$a^{24} \equiv 1 \, mod \, 35$

Démonstration. Notons $c \equiv a^{(p-1)(q-1)}$Calculons c modulo p :

$c \equiv a^{(p-1)(q-1)} \equiv \left(a^{(p-1)}\right)^{(q-1)} \equiv 1^{(q-1)} \equiv 1 \, mod \, p$

où l'on appliquer le petit théorème de Fermat :$a^{(p-1)} \equiv 1 \, mod \, p$, car p ne divise pas a.

Calculons ce même c mais cette fois modulo q :

$c \equiv a^{(p-1)(q-1)} \equiv \left(a^{(q-1)}\right)^{(p-1)} \equiv 1^{(q-1)} \equiv 1 \, mod \, q$

où l'on appliquer le petit théorème de Fermat :$a^{q-1} \equiv 1 \, mod \, q$ car q ne divise pas a. Conclusion partielle :$c \equiv 1 \, mod \, p \, et \, c \equiv 1 \, mod \, q$

Nous allons en déduire que $c \equiv 1 \, mod \, pq$

comme $c \equiv 1 \, mod \, p$ alors il existe $\alpha \in \mathbb{Z}$ tel que $c = 1 + \alpha p$, comme $c \equiv 1 \, mod \, q$ alors il existe.[9]

1.4.1.4 L'algorithme d'Euclide étendu

L'algorithme d'Euclide étendu est une variante de l'algorithme d'Euclide qui permet, à partir de deux entiers a et b, de calculer non seulement leur plus grand commun diviseur (pgcd), mais aussi un couple de coefficients de Bézout, c'est-à-dire deux entiers u et v tels que

$$au + bv = pgcd(a, b). \tag{1.12}$$

Cet algorithme est particulièrement utilisé lorsque on souhaite calculer l'inverse multiplicatif d'un entier.

La question importante est comment calcule-t-on les coefficients u et v. L'idée principale de l'algorithme est d'effectuer les mêmes étapes que pour l'algorithme d'Euclide, mais en exprimant à chaque itération le reste comme une combinaison linéaire de a et b. Puisque le dernier reste est le pgcd, celui-ci sera alors exprimé comme une combinaison linéaire de a et b.[10]

1.4.1.5 Calcul des inverses multiplicative modulo$n$

On peut utiliser l'algorithme d'Euclide étendu afin de calculer l'inverse modulaire d'un entier. Avant de continuer, on va démontrer un résultat crucial pour la démarche.

Proposition 1.4.1 Un entier a est inversible modulo n si et seulement si $pgcd(a, n) = 1$.

Démonstration

On suppose d'abord que l'entier a est inversible modulo n. Il existe alors$b \in Z/nZ$ tel que $ab \equiv 1\ mod\, n$. De cette congruence on voit alors qu'il existe un entier v tel que $ab + nv = 1$. Par le théorème de Bézout on peut alors conclure que $pgcd(a, n) = 1$.

Réciproquement, on suppose que $pgcd(a, n) = 1$. Par le théorème de Bézout, il existe des entiers u, v tels que$au + nv = 1$. Par conséquent, $au \equiv 1\ mod\, n$ et u est alors l'inverse de a modulo n.

Supposons maintenant qu'on veut calculer l'inverse de a mod n, avec $n < a$.

On vient de montrer que si cet inverse existe, alors forcement $pgcd(a, n) = 1$. En appliquant l'algorithme d'Euclide étendu on obtient un couple (u, v) tels que$a*u+n*v = 1$. On a alors

$a * u + n * v = 1$

$a * u + 0 \equiv 1\ mod\, n.$

$a * u = 1 \equiv 1\ mod\, n$

La dernière équation est la définition de l'inverse. Ceci vaut dire que u est l'inverse de a :

$u = a^{-1}\ mod\, n.$

1.4.1.6 L'exponentiation rapide

Pour calculer des puissances modulo n rapidement, une méthode plus éfficace que de calculer a^kpuis de le réduire modulo n.

Exemple 1.4.1 Calculons $17154(mod\, 100)$. Tout d'abord on décompose l'exposant $k = 154$ en base 2 :$154 = 128 + 16 + 8 + 2 = 2^7 + 2^4 + 2^3 + 2^1$, il s'écrit donc en base 2 :$(1, 0, 0, 1, 1, 0, 1, 0)$. Ensuite on calcule $17, 17^2, 17^4, 17^8, ..., 17^{128}$ modulo 100.

$17 \equiv 17\ mod\, 100$

$17^2 \equiv 17 * 17 \equiv 289 \equiv 89\ \ (mod\, 100)$

$17^4 \equiv 17^2 * 17^2 \equiv 89 * 89 \equiv 7921 \equiv 21\ \ (mod\, 100)$

$17^8 \equiv 17^4 * 17^4 \equiv 21 * 21 \equiv 441 \equiv 41\ \ (mod\, 100)$

$17^{16} \equiv 17^8 * 17^8 \equiv 41 * 41 \equiv 1681 \equiv 81\ \ (mod\, 100)$

$17^{32} \equiv 17^{16} * 17^{16} \equiv 81 * 81 \equiv 6561 \equiv 61\ \ (mod\, 100)$

$17^{64} \equiv 17^{32} * 17^{32} \equiv 61 * 61 \equiv 3721 \equiv 21\ \ (mod\, 100)$

$17^{128} \equiv 17^{64} * 17^{64} \equiv 21 * 21 \equiv 441 \equiv 41\ (mod\, 100)$

Pour calculer 17^{154}

$17^{154} \equiv 17^{128} * 17^{16} * 17^8 * 17^2 \equiv 41 * 81 * 41 * 89 \equiv 3321 * 3649 \equiv 21 * 49 \equiv 1029 \equiv 29\ \ (mod\, 100)$

Avantages et inconvénients de chiffrement asymétriques

La clé de chiffrement est différente de celle du déchiffrement, ce qui résous le problème de partage de clés, par contre ce type de schéma de chiffrement est très lent.

1.5 Nouvelles approches

Les contraints imposé par la politique de sécurisation, motivent les chercheurs à proposer de nouvelles solutions pour satisfaire les exigences de confidentialité de la communauté mondiale.

1.5.1 Chiffrement hybride

La cryptographie hybride utilise des algorithmes à clé publique et des algorithmes à clé privée, d'où l'adjectif hybride. Ce faisant, il combine les avantages des deux systèmes et pallie à certains inconvénients. En effet, un chiffrement hybride est rapide mais ne présente pas de faiblesse au niveau de la clé comme un chiffrement à clé publique. En effet, un algorithme symétrique oblige à conserver la clé sur le disque dur ou sur une clé USB, ce qui implique un risque qu'un pirate s'infiltre dans la mémoire de l'ordinateur ou du support USB pour avoir accès à la clé. Par analogie, c'est comme de posséder un coffre-fort et de ranger la clé dans un tiroir ouvert : n'importe quel cambrioleur peut la trouver et ouvrir le coffre sans avoir à le crocheter.

La plupart des systèmes hybrides fonctionnent de la manière suivante. Une clé aléatoire (ou pseudo-aléatoire) est générée pour l'algorithme symétrique (par exemple AES). Elle varie généralement entre 128, 256 ou 512 bits selon les algorithmes. Le destinataire génère alors une clé publique et une clé privée. La clé publique sert à chiffrer la clé aléatoire. Étant donné que cette dernière est courte, la chiffrer est rapide, alors que chiffrer le message avec un algorithme asymétrique aurait été bien plus long. Il ne reste plus qu'à envoyer le message chiffré accompagné de la clé chiffrée correspondante. Le destinataire utilise alors sa clé privée pour déchiffrer la clé aléatoire. Avec cette dernière, il retrouve le message via un déchiffrement symétrique.

1.5.2 Chiffrement quantique

La cryptographie quantique repose sur trois domaines distincts : – La cryptographie, en ce sens qu'elle permet de garantir la confidentialité d'une clé. – La physique quantique, et plus particulièrement la mécanique quantique. – La théorie de l'information, car elle fournit un "système" inconditionnellement sûr. Le problème majeur en cryptographie est le transfert de la clé entre les deux parties communicantes. Ici, on n'émet aucune hypothèse sur la sécurité du canal employé. La raison en est que la cryptographie quantique

repose sur le principe d'Heisenberg :

Certaines quantités subatomiques ne peuvent être simultanément mesurées.

La conséquence de ce principe est qu'il est impossible de mesurer ces particules sans les modifier. Il est donc possible de construire un canal de communication que nul ne peut espionner sans modifier la transmission de manière détectable. Ainsi, il est possible de transmettre une clé secrète entre deux personnes sans qu'elles disposent d'informations secrètes communes préalables. Dans le cadre du transport d'une clé, la technique qui nous occupera ici consistera en l'envoi de photons. On utilisera la technique dite de Polarisation de photons.[11]

1.5.3 Chiffrement chaotique

Pendant longtemps, le chaos à été considéré comme dangereux, ou indésirable par la communauté scientifique. Cependant, dans les années 90, des scientifiques ont réalisé que le chaos pouvait être contrôlé et ont commencé à chercher ses applications possible. Les signaux issus des systèmes chaotiques sont imprédictibles à long terme, peuvent présenter des propriétés spectrales et statistiques proches de l'aléatoire (signaux à large spectre, autocorrélation réduite), bien qu'issus de systèmes déterministes. Ces caractéristiques sont liées aux propriétés requises par les schémas de chiffrement, telle que la confusion et la diffusion de Shannon, usuellement rencontrées en cryptographie, usuellement montré que les systèmes chaotiques peuvent être synchronisés. Une des applications du chaos qui alors a alors intéressé les chercheurs est l'utilisation de systèmes chaotiques à des fins de chiffrement.

De nombreux schémas de chiffrement basés sur le chaos ont été proposés dans la littérature. En revanche, très peu de travaux ont réellement fait un lien entre les algorithmes de chiffrement standard et ceux basés sur la génération de séquences chaotiques.[12]

1.6 La cryptanalyse

La cryptanalyse est l'ensembles des procédés d'attaque d'un cryptosystème. Elle est indispensable pour l'étude de la sécurité des procédés de chiffrement utilisés en cryptographie. Son but ultime est de trouver un algorithme de déchiffrement des messages. Le plus souvent on essaye de reconstituer la clé secrète de déchiffrements. On vertu de principes des Kerckhoffs, la seule partie secrète du cryptosystème est la clé. Les attaques souvent évoquées dans la littérature spécialisée dans le domaine de la cryptologie sont :

1. **Attaque à texte chiffrée** L'attaquant ne dispose que d'un ou plusieurs messages chiffrés qu'il souhaite déchiffrer. C'est le type d'attaque le plus difficile.

2. **Attaque à texte clair connu :** Le cryptanalyse a non seulement accès aux textes Chiffrés de plusieurs messages, mais aussi aux textes clairs correspondants. La tâche est de retrouver la ou les clés qui ont été utilisées pour chiffrer ces messages ou un algorithme qui permet de déchiffrer d'autres messages chiffrés avec ces mêmes clés.

3. **Attaque à texte clair choisi :** L'opposant a accès à une machine Chiffrant : Le cryptanalyste a non seulement accès aux textes chiffrés et aux textes clairs correspondants, mais de plus il peut choisir les textes en clair. Cette attaque est plus efficace que l'attaque à texte clair connu, car le cryptanalyste peut choisir des textes en clair spécifiques qui donneront plus d'informations sur la clé.

4. **Attaque à texte chiffrée choisi** : L'opposant a accès à une machine déchiffrable : Le cryptanalyste peut choisir différents textes chiffrés à déchiffrer. Les textes déchiffrés lui sont alors fournis. Par exemple, le cryptanalyste a un dispositif qui ne peut être désassemblé et qui fait du déchiffrement automatique. Sa tâche est de retrouver la clé.

5. **Attaque par force brute** Elle consiste en l'énumération de toutes les valeurs possibles de la clé, c'est-à-dire à une exploration complète de l'espace des clés. La complexité de cette méthode apparait immédiatement : une clé de 64 bits oblige à essayer 264 combinaisons différentes. Pour une clé de 64 bits, il existe$1.844 * 10^{19}$ combinaisons différentes, sur un ordinateur essayant un milliard de clés par seconde il faudra 584 ans pour être sûr de trouver la clé.[3]

6. **Attaque par séquences connues** Ce type d'attaque consiste à supposer connue une certaine partie du message en clair (par exemple les entêtes standards dans le cas d'un message transmis par courrier électronique) et de partir de cette connaissance pour essayer de deviner la clé. Cette attaque peut réussir si l'algorithme de chiffrement laisse apparaitre les mêmes régularités que le message original.[3]

7. **Attaque par séquences forcées** Cette méthode, basée sur la précédente, consiste à faire chiffrer par la victime un bloc dont l'attaquant connait le contenu.[3]

8. **Attaque par analyse différentielle** Cette attaque utilise les faibles différences existant entre des messages successifs (par exemple des identifiant (logs) de serveur) pour essayer de deviner la clé.[3]

1.6.1 Différentes notions de sécurité d'un cryptosystème

1. **La sécurité inconditionnelle** : qui ne préjuge pas de la puissance de calcul du cryptanalyste qui peut être illimitée.

2. **La sécurité calculatoire** : La notion de sécurité calculatoire repose sur la théorie de la complexité, pour évaluer la sécurité d'un cryptosystème on utilise souvent la sécurité calculatoire, elle repose sur le critère suivant « même avec des ordinateurs faisant 10^9 opérations élémentaires par seconde un calcul qui n´nécessite 2^100 opérations ´élémentaires est hors de portée actuellement car pour l'effectuer il faut environ 4 *10^13 années !.

3. **La sécurité prouvée** : la sécurité prouvée consiste à ramener la sécurité du système a un problème réputé par sa complexité.

4. **La confidentialité parfaite** : le codage utilisé ne donne aucune information sur la clé.

1.7 Conclusion

Dans ce chapitre, nous avons présenté les briques fondamentales de la cryptographie ainsi, certains techniques et quelque fondement théoriques qui vont permettre de bien comprendre cet axe de recherche.Le chapitre présent les deux types de chiffrement symétrique et asymétrique avec des exemples des algorithmes de cryptages symétrique par flot et par bloc. Il traite aussi les différents modes de chiffrements utilisé dans le chiffrement symétrique et les notions formelle de la sécurité informatique. Nous avons évoqué aussi les différentes formes d'attaque sur les les cryptosystèmes.

Chapitre 2

Les algorithmes de cryptage :DES et RSA

2.1 Introduction

Le chiffrement à clé secréte repose sur le principe $Ke = Kd = K$. Autrement dit, Alice et Bob conviennent secrètement d'une clé secréte K qui est donc utilisée à la fois pour le chiffrement et le déchiffrement. Ils conviennent également d'un algorithme cryptographique de chiffrement et de déchiffrement. On utilise souvent l'analogie au coffre-fort pour caractériser les systèmes cryptographiques à clé secréte : seul celui qui possède la clé (Alice et Bob à priori) est capable d'ouvrir le coffre. Oscar, qui ne la possède pas, devra forcer le coffre s'il veut accéder à son contenu. Évidemment, si Oscar parvient par un moyen quelconque à obtenir cette clé, il pourra déchiffrer tous les messages échangés entre Alice et Bob.

Dans le cas des systèmes symétriques, on utilise une même clé pour le chiffrement et le déchiffrement. Le problème est dans la transmission de la clé. Il faut une clé par destinataire. Dans le cas des systèmes asymétriques, chaque personne possède deux clés distinctes (clé privée, clé publique) avec impossibilité de déduire la clé privée à partir de la clé publique. De ce fait, il est possible de distribuer librement cette dernière.

Dans ce chapitre, nous présentons deux algorithmes de chiffrement les plus utilisés. l'algorithme **Data Encryption Standard** (DES) et le second algorithme de **Rivest, Shamir et Adlman** (RSA).

2.2 Algorithme Data Encryption Standard (DES)

Publié en 1977 par le NBS (National Bureau of Standards), le DES est un algorithme de chiffrement de données recommandé pour les organisations à caractère fédéral, com-

mercial ou privé. Le DES tire son origine des travaux menés par le groupe cryptographique d'IBM dans le cadre du projet LUCIFER. Le DES a été l'objet de nombreuses implémentations, à la fois en matériel et en logiciel, depuis sa publication. Après une décennie de succès, pendant laquelle les moyens et techniques de cryptanalyse mis en œuvre pour en étudier les caractéristiques n'ont pas permis d'en découvrir des faiblesses rédhibitoires, le DES a, depuis peu, révélé des sensibilités à des attaques nouvelles et puissantes, parfois réalisées sur un simple micro-ordinateur. Aussi l'ISO (International Organization for Standardization) a-t-il récemment refusé la normalisation du DES, ce qui n'empêche pas cet algorithme d'être, de loin, aujourd'hui encore comme le moyen de chiffrement le plus sûr (et le plus largement utilisé) pour des données non militaires.[6]

2.2.1 Principe du chiffrement DES

Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits dont 8 bits (1 octet) servent de teste de parité (pour vérifier l'intégralité de la clé). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester 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 impaire de (1) dans l'octet à qui il appartient. La clé possède donc une longueur utile de 56 bites, ce qui signifie que seuls 56 bites servent réellement dans l'algorithme. L'algorithme consiste à effectuer des combinaisons 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 bites et formée de 16 blocs de 4 bites, généralement notés K1 à K16 étant donné que seule 56 bites servent effectivement à chiffrer, il peut exister $2^{56}(soit\, 7.2 * 10^{16})$ clé différentes.[6]

2.2.2 L'algorithme DES

Les grandes lignes de l'algorithme sont les suivantes :

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

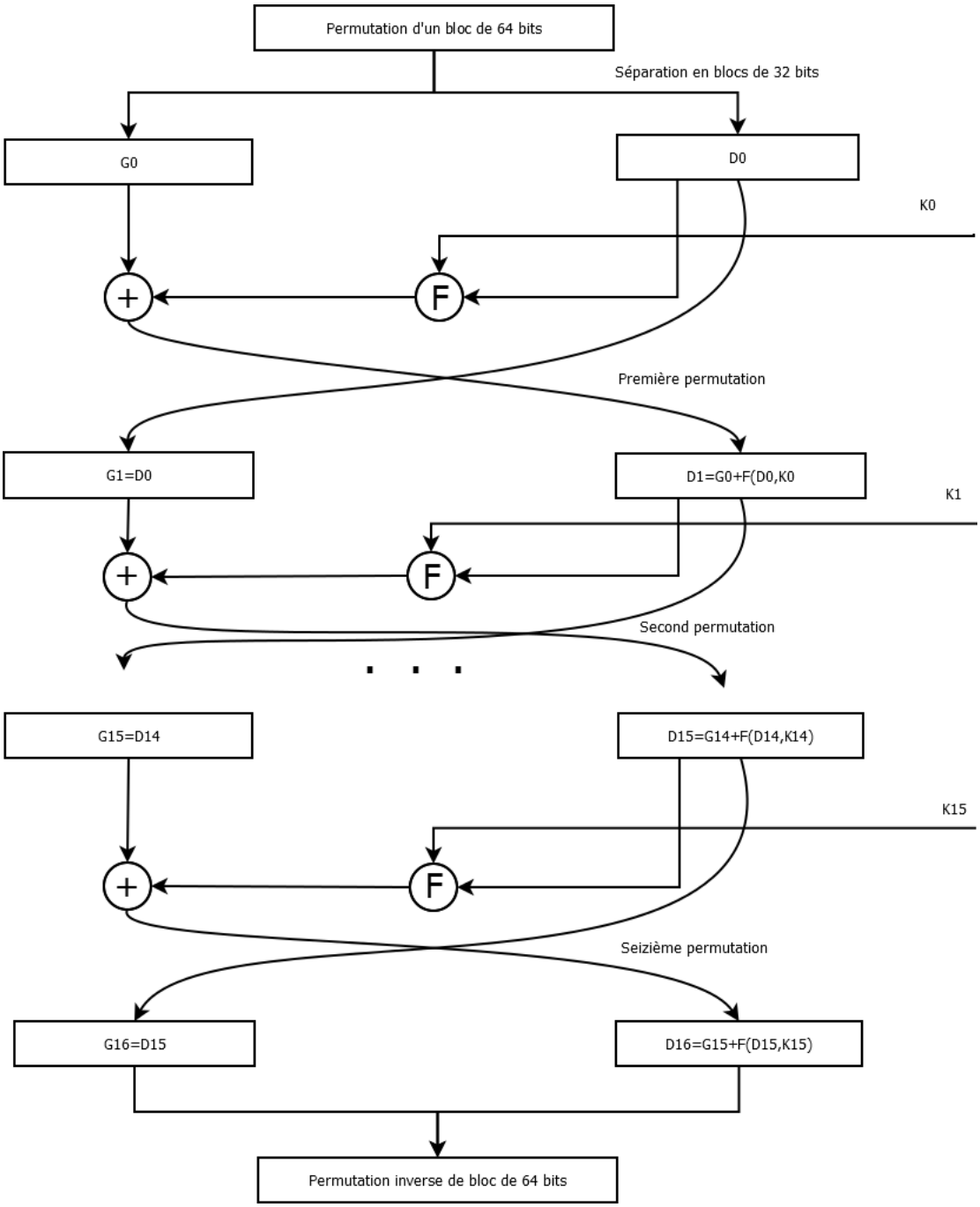

FIGURE 2.1 – Algorithme DES.

2.2.2.1 Fractionnement du texte

Le fractionnement consiste en premier lieu à convertir le message clair en binaire, puis le décomposer en blocs de 64 bits (matrices 8x8).

2.2.2.2 Permutation initiale

Dans un premier temps, chaque bit d'un bloc est soumis à la permutation initiale, pouvant être représentée par la matrice de permutation initiale (notée PI) suivante :

$$PI = \begin{matrix} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 2 \\ 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 \\ 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \\ 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{matrix} \tag{2.1}$$

Matrice de permutation initiale

Cette matrice de permutation indique, en parcourant la matrice de gauche à droite puis de haut en bas, que le 58ème bit du bloc de texte de 64 bits se retrouve en première position, le 50ème en seconde position et ainsi de suite.[6]

2.2.2.3 Scindement en blocs de 32 bits

Une fois la permutation initiale est réalisé, le bloc de 64 bits et scindé en deux blocs de 32 bits noté respectivement G et D .

$$G = \begin{matrix} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 10 \\ 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 \\ 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \end{matrix} \tag{2.2}$$

$$D = \begin{matrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{matrix} \tag{2.3}$$

La partie gauche contient les bits de position impaire et la partie droit contient les bits de position paire.[6]

2.2.2.4 Les rondes

Les blocs Gn et Dn sont soumis à un ensemble de transformation itératives appelées rondes, explicitées dans le schéma donné en Figure 2.2, et dont les détails sont donnés plus bas :

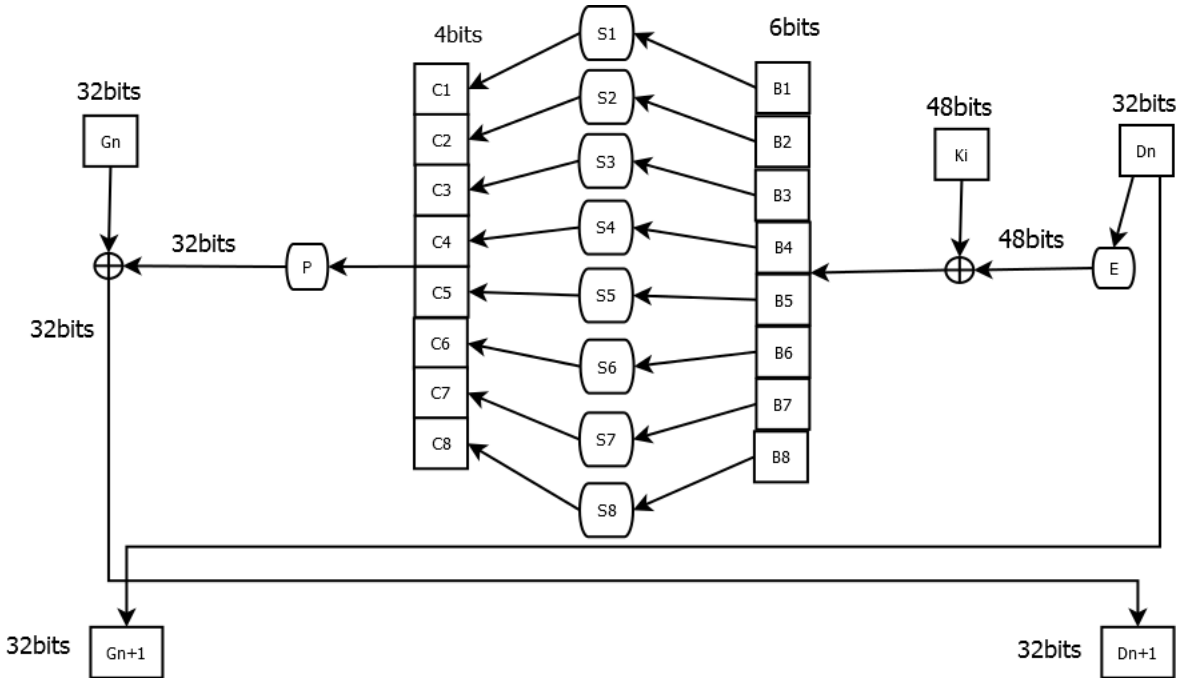

FIGURE 2.2 – Le rounde de DES.

Fonction d'expansion

Les 32 bits de blocs$D_n$sont étendu à 48 bits garce à une fonction d'expansion E

$$E = \begin{matrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{matrix} \tag{2.4}$$

Expansion de D_n

Fonction de substitution

D_n est ensuite scindé en 8 blocs de 6 bits, notéB_j. Chacun de ces blocs passe par une fonction de sélection (appelées parfois boite de substitutions ou fonctions de compression), notées S_j.[6]

La sélection de la ligne se fait sur 2 bits et la colonne sur 4 bits

$$S1 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 14 & 4 & 13 & 1 & 2 & 15 & 11 & 8 & 3 & 10 & 6 & 12 & 5 & 9 & 0 & 7 \\ 1 & 0 & 15 & 7 & 4 & 14 & 2 & 13 & 1 & 10 & 6 & 12 & 11 & 9 & 5 & 3 & 8 \\ 2 & 4 & 1 & 14 & 8 & 13 & 6 & 2 & 11 & 15 & 12 & 9 & 7 & 3 & 10 & 5 & 0 \\ 3 & 15 & 12 & 8 & 2 & 4 & 9 & 1 & 7 & 5 & 11 & 3 & 14 & 10 & 0 & 6 & 13 \end{array} \quad (2.5)$$

$$S2 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \\ 1 & 3 & 13 & 4 & 7 & 15 & 2 & 8 & 14 & 12 & 0 & 1 & 10 & 6 & 9 & 11 & 5 \\ 2 & 0 & 14 & 7 & 11 & 10 & 4 & 13 & 1 & 5 & 8 & 12 & 6 & 9 & 3 & 2 & 15 \\ 3 & 13 & 8 & 10 & 1 & 3 & 15 & 4 & 2 & 11 & 6 & 7 & 12 & 0 & 5 & 14 & 9 \end{array} \quad (2.6)$$

$$S3 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 10 & 0 & 9 & 14 & 6 & 3 & 15 & 5 & 1 & 13 & 12 & 7 & 11 & 4 & 2 & 8 \\ 1 & 13 & 7 & 0 & 9 & 3 & 4 & 6 & 10 & 2 & 8 & 5 & 14 & 12 & 11 & 15 & 1 \\ 2 & 13 & 6 & 4 & 9 & 8 & 15 & 3 & 0 & 11 & 1 & 2 & 12 & 5 & 10 & 14 & 7 \\ 3 & 1 & 10 & 13 & 0 & 6 & 9 & 8 & 7 & 1 & 15 & 14 & 3 & 11 & 5 & 2 & 12 \end{array} \quad (2.7)$$

$$S4 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 7 & 13 & 14 & 3 & 0 & 6 & 9 & 10 & 1 & 2 & 8 & 5 & 11 & 12 & 4 & 15 \\ 1 & 13 & 8 & 11 & 5 & 6 & 15 & 0 & 3 & 4 & 7 & 2 & 12 & 1 & 10 & 14 & 9 \\ 2 & 10 & 6 & 9 & 0 & 12 & 11 & 7 & 13 & 15 & 1 & 3 & 14 & 5 & 2 & 8 & 14 \\ 3 & 3 & 15 & 0 & 6 & 10 & 1 & 13 & 8 & 9 & 4 & 5 & 11 & 12 & 7 & 2 & 4 \end{array} \quad (2.8)$$

$$S5 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 2 & 12 & 4 & 1 & 7 & 10 & 11 & 6 & 8 & 5 & 3 & 15 & 13 & 0 & 14 & 9 \\ 1 & 14 & 11 & 2 & 12 & 4 & 7 & 13 & 1 & 5 & 0 & 15 & 10 & 3 & 9 & 8 & 6 \\ 2 & 4 & 2 & 1 & 11 & 10 & 13 & 7 & 8 & 15 & 9 & 12 & 5 & 6 & 3 & 0 & 14 \\ 3 & 11 & 8 & 12 & 7 & 1 & 14 & 2 & 13 & 6 & 15 & 0 & 9 & 10 & 4 & 5 & 3 \end{array} \quad (2.9)$$

$$S6 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 12 & 1 & 10 & 15 & 9 & 2 & 6 & 8 & 0 & 13 & 3 & 4 & 14 & 7 & 5 & 11 \\ 1 & 10 & 15 & 14 & 2 & 7 & 12 & 9 & 5 & 6 & 1 & 13 & 14 & 0 & 11 & 3 & 8 \\ 2 & 9 & 14 & 15 & 5 & 2 & 8 & 12 & 3 & 7 & 0 & 4 & 10 & 1 & 13 & 11 & 6 \\ 3 & 4 & 3 & 2 & 15 & 9 & 5 & 15 & 10 & 11 & 14 & 1 & 7 & 6 & 0 & 8 & 13 \end{array} \quad (2.10)$$

$$S7 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 4 & 11 & 2 & 14 & 15 & 0 & 8 & 13 & 3 & 12 & 9 & 7 & 5 & 10 & 6 & 1 \\ 1 & 13 & 0 & 11 & 7 & 4 & 9 & 1 & 10 & 14 & 3 & 5 & 12 & 2 & 15 & 8 & 6 \\ 2 & 1 & 4 & 11 & 13 & 12 & 3 & 7 & 14 & 10 & 15 & 6 & 8 & 0 & 5 & 9 & 2 \\ 3 & 6 & 11 & 13 & 8 & 1 & 4 & 10 & 7 & 9 & 5 & 0 & 15 & 14 & 2 & 3 & 12 \end{array} \quad (2.11)$$

$$S8 = \begin{array}{c|cccccccccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 13 & 2 & 8 & 4 & 6 & 15 & 11 & 1 & 10 & 9 & 3 & 14 & 5 & 0 & 12 & 7 \\ 1 & 1 & 15 & 13 & 8 & 10 & 3 & 7 & 4 & 12 & 5 & 6 & 11 & 0 & 14 & 9 & 2 \\ 2 & 7 & 11 & 4 & 1 & 9 & 12 & 14 & 2 & 0 & 6 & 10 & 13 & 15 & 3 & 5 & 8 \\ 3 & 2 & 1 & 14 & 7 & 4 & 10 & 8 & 13 & 15 & 12 & 9 & 0 & 3 & 5 & 6 & 11 \end{array} \quad (2.12)$$

Les 8 $S-Box$ du DES

chaque blocs de 6 bits est substitué en blocs de 4 bits. 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.

Exemple 1.2.1 Soit B_1 égal à 101110. Les premiers et derniers bits donnent 10, c'est-à-dire 2 en binaire. Les bits 2, 3, 4 et 5 donnent 0111, soit 7 en binaire. Le résultat de la fonction de sélection est donc la valeur situé à la ligne n°2, dans la colonne n°7. Il s'agit de la valeur 11, soit en binaire 1011.[6]

Permutation

Les blocs de 32 bits obtenus sont soumis a des permutations.[6]

$$P = \begin{matrix} 16 & 7 & 20 & 21 & 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 & 5 & 18 & 31 & 10 \\ 2 & 8 & 24 & 14 & 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 & 22 & 11 & 4 & 25 \end{matrix} \tag{2.13}$$

OU Exclusif

L'ensemble des résultats et soumis en sortie de P à un OU Exclusif avec G_n en suite affecter à D_{n+1} ainsi que D_n donne G_{n+1}.[6]

Itérations

Les étapes précédent seront appliquées 16 fois (*ronds*).[6]

Permutation initiale inverse

A la fin des ronds les deux blocs G_{16} et D_{16} sont soumis à une permutation inverse pour obtenir un texte codé sur 64 bits.[6]

$$PI^{-1} = \begin{matrix} 40 & 8 & 48 & 16 & 56 & 24 & 64 & 32 \\ 39 & 7 & 47 & 15 & 55 & 23 & 63 & 31 \\ 38 & 6 & 46 & 14 & 54 & 22 & 62 & 30 \\ 37 & 5 & 45 & 13 & 53 & 21 & 61 & 29 \\ 34 & 4 & 44 & 12 & 52 & 20 & 60 & 28 \\ 35 & 3 & 43 & 11 & 51 & 19 & 59 & 27 \\ 34 & 2 & 42 & 10 & 50 & 18 & 58 & 26 \\ 33 & 1 & 41 & 9 & 49 & 17 & 57 & 25 \end{matrix} \tag{2.14}$$

2.2.2.5 Génération des clés

La Figure2.3 montre comment à partir d'une clé de 48 bits obtenir 8 clés diversifiées de 48 bits.

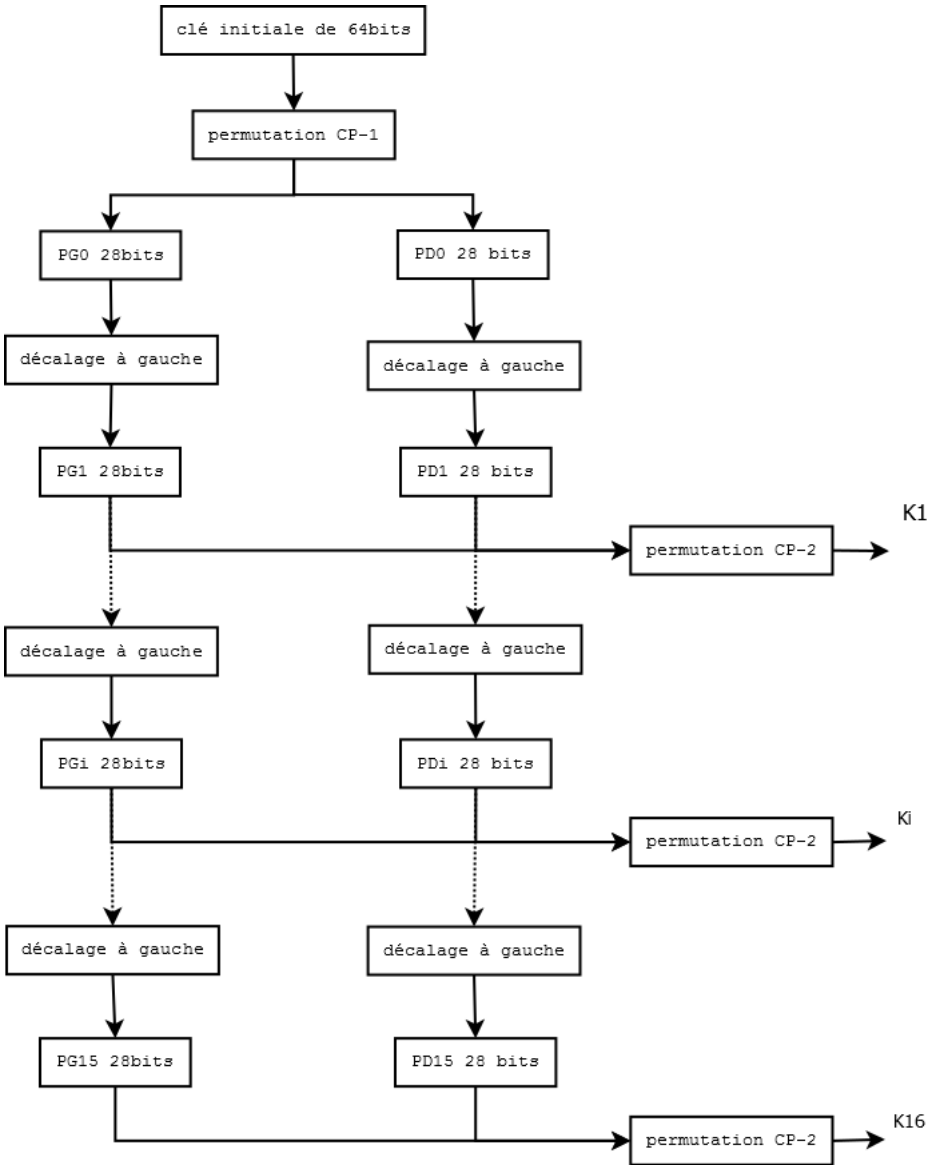

FIGURE 2.3 – Algorithme de génération des clés.

La première étape consiste à une permutation noté CP-1

$$CP\text{-}1= \begin{matrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 & 60 & 52 & 44 & 36 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{matrix} \tag{2.15}$$

Cette matrice peut s'écrire sous forme de deux matrices partie gauche et partie droit.

$$PGi = \begin{matrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ 19 & 11 & 3 & 60 & 52 & 44 & 36 \end{matrix} \tag{2.16}$$

$$PDi = \begin{matrix} 63 & 55 & 47 & 39 & 31 & 57 & 62 \\ 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 \\ 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{matrix} \tag{2.17}$$

On note G0 et D0 le résultat de cette première permutation.

Ces deux blocs subissent ensuite une rotation à gauche, de telles façons que les bits en seconde position prennent la première position, ceux en troisième position prennent la seconde, ect. Les bits en première position passent en dernière position.

Les 2 blocs de 28 bits sont ensuite regroupés en un bloc de 56 bits. Celui-ci passe par une permutation, notée CP-2, fournissant en sortie un bloc de 48 bits, représentant la clé Ki.[6]

$$CP-2 = \begin{matrix} 14 & 17 & 11 & 24 & 1 & 5 & 3 & 28 & 15 & 6 & 21 & 10 \\ 23 & 19 & 12 & 4 & 26 & 8 & 16 & 7 & 27 & 20 & 13 & 2 \\ 41 & 52 & 31 & 37 & 47 & 55 & 30 & 40 & 51 & 45 & 33 & 48 \\ 44 & 49 & 39 & 56 & 34 & 33 & 46 & 42 & 50 & 36 & 29 & 32 \end{matrix} \tag{2.18}$$

Des itérations de l'algorithme permettent de donner les 16 clés K1 à K16 utilisées dans l'algorithme du DES.

Numéro d'itération	Nombre de décalages à gauche
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 2.4 – Rotation de la clé.

2.3 Décryptage DES

Le déchiffrement du DES s'effectue en appliquant les 16 clés inversement, comme il est indiqué dans la figure suivante.

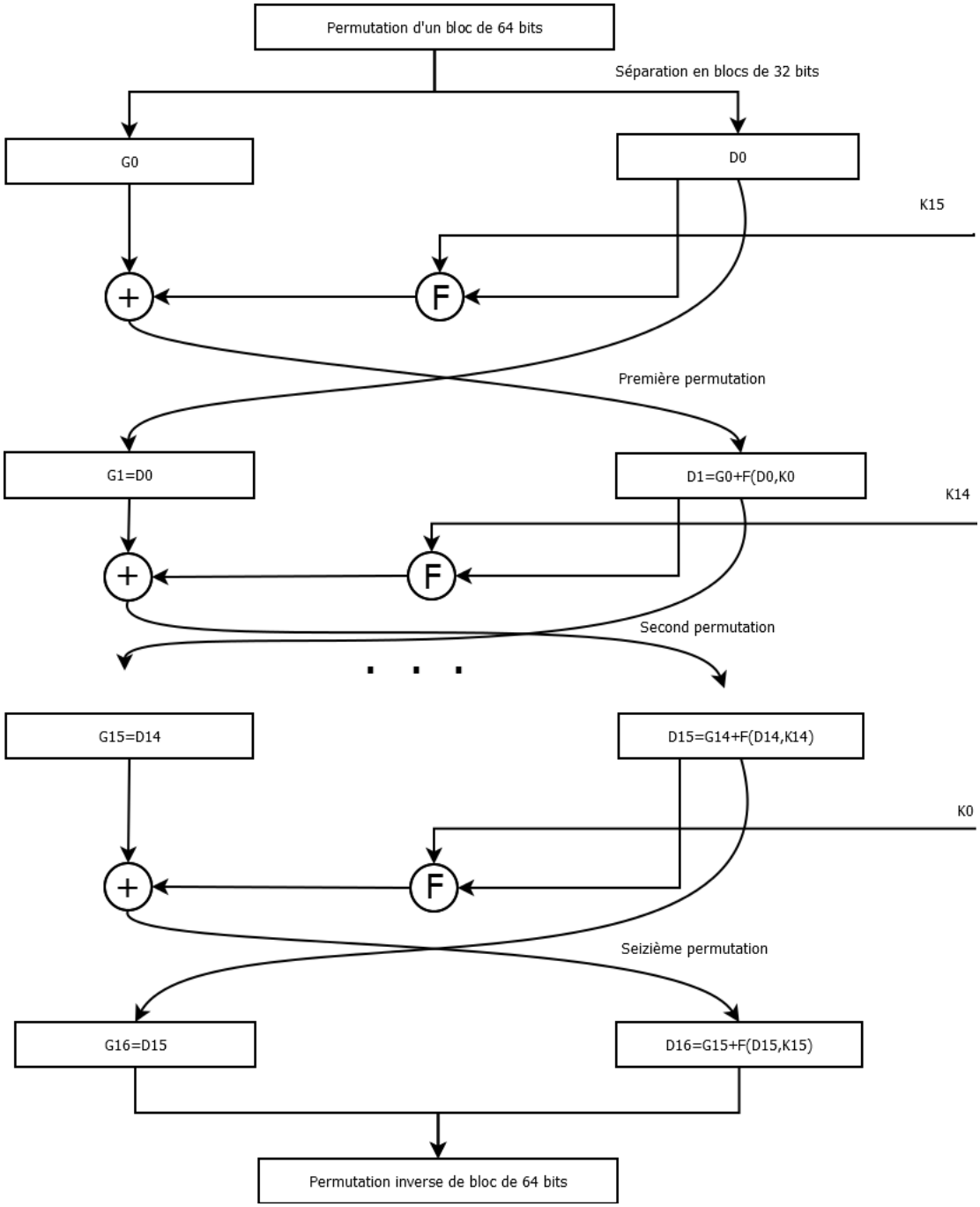

FIGURE 2.5 – Algorithme de décryptage de DES.

2.4 Six façons différentes pour casser DES

2.4.1 Recherche exhaustive

Imaginons que nous disposions d'un bloc de données chiffrées C' et que nous voulions trouver la clé secrète correspondante. Si nous disposons d'un peu d'information sur la structure ou le contenu des données en clair (texte ASCII, image JPEG, paquet de réseau ayant une structure connue,...), la méthode la plus simple est une recherche exhaustive de la clé correspondante parmi les $2^{56} \simeq 7.2 * 10^{16}$ possibilités. Le principe, très simple, est le suivant : une clé après l'autre, on essaye de déchiffrer le bloc de données, l'information sur le texte clair nous permettant de reconnaître la bonne et donc d'interrompre la recherche. Nous aurons besoin, en moyenne, de2^{55} essais avant de terminer notre attaque[8].

2.4.2 Une machine dédiée

La complexité, mesurée en terme de quantité de calcul, d'une recherche exhaustive est certes énorme, mais à présent, grâce aux progrès de la technologie des microprocesseurs depuis 1977, elle est loin d'être inaccessible. DES est un algorithme orienté hardware qui a le gros désavantage pour le cryptanalyste d'être très lent en software. Une plate-forme PC, à savoir un processeur Intel Pentium III 666MHz, peut examiner environ 2millions de clés par seconde, ce qui implique un temps de recherche moyen de 600 années pour un seul PC (avec un processeur de 3GHz on peut examiner environ 9 millions de clés/seconde donc un temps de recherche approximatif de 254 ans) . Une solution matérielle a été proposée et réalisée par l'EFF (Electronic Frontier Fundation) en 1998, dans le seul but de prouver que DES n'est pas (ou plus) du tout un algorithme sûr. Deep Crack, tel est le nom de cette machine extraordinaire, a coûté moins de 210'000$. Elle est constituée de 1536chips qui sont capables de décrypter un bloc en 16 cycles d'horloge, le tout étant cadencé à 40 MHz. Ces caractéristiques lui donnent la possibilité d'examiner 92milliards de clés par seconde, ce qui donne un temps de recherche moyen situé entre 4 et 5 jours. Vu le budget modeste de l'EFF, il n'est pas difficile de tirer des conclusions alarmistes sur la sécurité de DES vis-à-vis d'organismes gouvernementaux (tel la NSA, organe américain chargé de l'espionnage électronique) ou d'organisations criminelles.[8]

2.4.3 Un gigantesque cluster

Il n'est même pas nécessaire de disposer de gros moyens financiers pour pouvoir casser le DES. De la bonne volonté et quelques bénévoles sont suffisants. Le 19 janvier 1999, dans le cadre d'un concours sponsorisé par une des entreprises majeures en sécurité informatique, RSA Labs, a cassé une clé en moins de 23 heures. L'organisation **distributed .net** regroupe des milliers d'ordinateurs (de la machine la plus simple aux serveurs mul-

tiprocesseurs les plus puissants) sur Internet qui fournissent gracieusement leur puissance de calcul à disposition lorsqu'ils sont inactifs. Plus de 100'000 ordinateurs ont reçu un certain nombre de clés à contrôler via le réseau, ce qui a permis un taux de traitement de 250milliards de clés par seconde.[8]

2.4.4 Un compromis temps-mémoire

Nous avons déjà abordé l'idée de recherche exhaustive de clé : on a à disposition un couple texte clair - texte chiffré et on essaye les2^{56} clés possibles. Cette méthode ne demande quasiment aucune mémoire, mais en contrepartie, on devra essayer en moyenne 2^{55} clés avant de tomber sur la bonne. D'un autre côté, il serait imaginable, pour un texte clair x donné, de pré-calculer le texte chiffré $y_k = E_K(x)$ correspondant pour toutes les2^{56}clés K, et de stocker les paires (y_K, K) triées par leur première coordonnée. Plus tard, lorsque l'on obtient un texte chiffré y à partir de x, il est possible, par une simple recherche dans notre table, de retrouver la clé correspondante. Nous pouvons noter que cette recherche demande un temps constant ; par contre, nous avons un besoin énorme de mémoire (1.5 milliard de Go), ainsi que de beaucoup de temps pour construire cette table. De plus, le bénéfice ne devient effectif que si l'on doit chercher plus d'une clé à partir du texte chiffré d'un même message. Un algorithme, dit de compromis temps-mémoire, a été proposé en 1980 par **Hellman**. Il combine à la fois une demande de mémoire moindre, ainsi qu'un temps de calcul inférieur à celui d'une recherche exhaustive. Cette attaque demande environ 1000 Go de capacité de stockage et 5 jours de calculs sur un simple PC.[8]

2.4.5 Cryptanalyse différentielle

En 1990, deux chercheurs du Weitzmann Institute, Biham et Shamir, ont présenté une nouvelle attaque, la cryptanalyse différentielle. En utilisant cette méthode, les deux chercheurs ont proposé pour la première fois une façon de casser DES qui demande moins de temps de travail qu'une recherche exhaustive. Imaginons que nous disposions d'un boîtier électronique capable de chiffrer des données avec une clé inconnue câblée dans le matériel ; de plus, nous ne disposons pas du matériel nécessaire pour récupérer physiquement la clé dans le boîtier. Il nous est cependant possible de produire au moyen de textes clairs choisis les textes chiffrés correspondants avec cette clé inconnue. La meilleure attaque différentielle connue demande actuellement 2^{47} textes clairs choisis. La phase d'analyse calcule la clé à l'aide de cet énorme amas de données. Une propriété intéressante de cette attaque est qu'il est possible de la monter même si le nombre de données disponibles est petit ; la probabilité de succès augmente linéairement avec ce nombre.[8]

2.4.6 Cryptanalyse linéaire

la cryptanalyse linéaire a été proposée par Matsui, de Mitsubishi Electronics, en 1993 . Bien qu'elle ne soit que théoriquement utile dans le monde réel, c'est l'attaque la plus efficace connue à ce jour contre DES. Imaginons le scénario suivant : nous disposons d'un grand nombre de couples texte clair - texte chiffré avec une clé identique. Nous pouvons dans ce cas procéder à ce que l'on nomme une attaque à texte clair connu. Ainsi, il est possible d'exploiter une faiblesse d'une des briques composantes de DES en effectuant des statistiques sur un flot de2^{43} couples de données (soit la bagatelle de deux fois 64 To).[8]

2.5 Avantages de DES

Après le chiffrement de message par le DES, les caractéristiques de message clair (la fréquence des caractères, le nombre d'espace..ect) seront indétectables.

Le changement de la clé au cours des 16 tours appliqué provoque un changement important dans le texte chiffré, est cela nous permet d'éviter une attaque différentielle. d'un autre coté, l'application de DES est facilement implémentable au niveau matériel est donc, la possibilité d'avoir des vitesses de chiffrement importants.[3]

2.6 Application du chiffrement DES sur un texte

— Voici un message clair « chiffré chiffré » constitué de deux blocs de 64 bits sur lequel on applique le chiffrement DES avec les modes EBC et CBC.
— La clé de chiffrement sur 64 bits « 8 caractères » est « sécurité ».
— La matrice bin_texte représente le code binaire du message clair.

```
ecrivez votre message:' chiffré chiffré'

x =

 chiffré chiffré

donnez une clé de 8 caractéres:'sécurité'

bin_texte =

     0     0     1     0     0     0     0     0
     0     1     1     0     0     0     1     1
     0     1     1     0     1     0     0     0
     0     1     1     0     1     0     0     1
     0     1     1     0     0     1     1     0
     0     1     1     0     0     1     1     0
     0     1     1     1     0     0     1     0
     1     1     1     0     1     0     0     1
     0     0     1     0     0     0     0     0
     0     1     1     0     0     0     1     1
     0     1     1     0     1     0     0     0
     0     1     1     0     1     0     0     1
     0     1     1     0     0     1     1     0
     0     1     1     0     0     1     1     0
     0     1     1     1     0     0     1     0
     1     1     1     0     1     0     0     1
```

FIGURE 2.6 – Présentation d'un message claire, sa représentation binaire et une clé de chiffrement.

2.6.1 Le DES avec le mode ECB

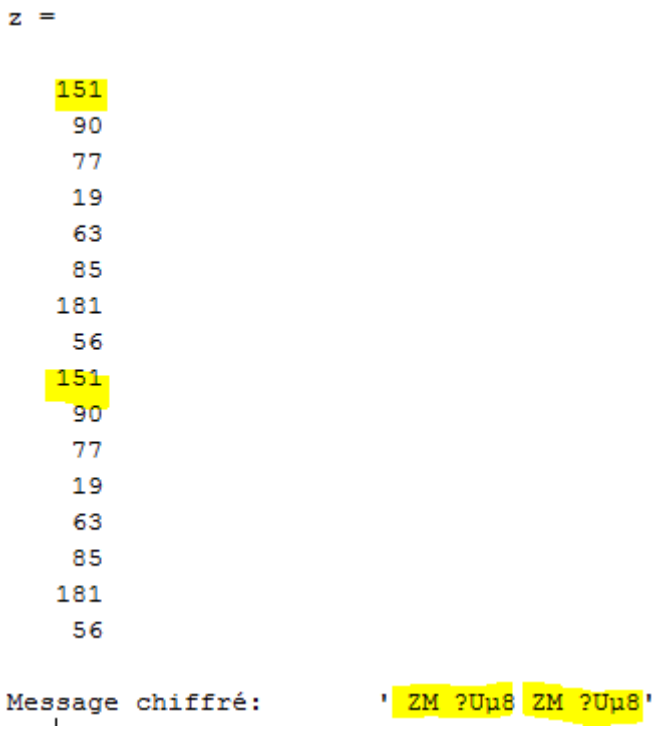

FIGURE 2.7 – Le résultats du chiffrement DES en mode ECB.

Dans 2.7 "z" est le vecteur qui contient le code ASCII du message chiffré.

Ce n'est pas si difficile de remarquer que les deux blocs sont identiques en observant seulement le message chiffré, ce qui est un atout pour celui qui veut casser le chiffrement, car s'il arrive à déchiffrer le premier bloc le deuxième aussi serai déchiffré. On peut imaginer les conséquences sur un message un peut plus long qui contient plusieurs blocs identiques.

Cet inconvénient a rendu ce mode vulnérable, et n'est jamais utilisé en pratique.

2.6.2 Le DES avec mode CBC

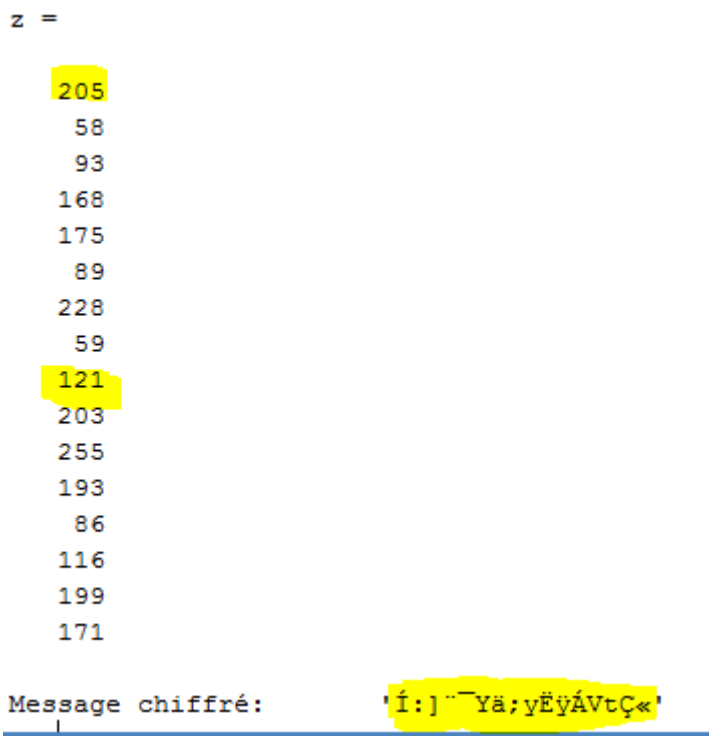

FIGURE 2.8 – Le résultats du chiffrement DES en mode CBC.

Le chiffrement en mode CBC a bien dissimulé l'information de ce message.

On remarque que ce mode assure un chiffrement différent pour plusieurs blocs identiques, Ce qui le rend le mode le plus utilisé dans le monde pratique.

L'utilisation de ce mode demande des calculs supplémentaires donc un temps de réponse un peut plus long. la différence est négligeable pour les petits textes.

On remarque que ce mode assure un chiffrement différent pour plusieurs blocs identiques. Ce qui le rend le mode le plus utilisé dans le monde pratique.

Déchiffrement

Le programme du déchiffrement des deux méthodes retrouve le message claire en lui donnant la même clé de chiffrement.

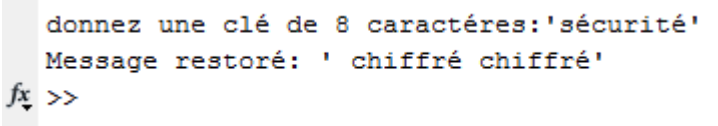

FIGURE 2.9 – Message restaurer.

2.7 Algorithme (Rivest, Shamir, Adleman) RSA

2.7.1 Chiffrement RSA

En 1977, dans a méthode for obtaining digital signature and public-Key cryptosystèmes, Ronald Rivest, Adi Shamir et Leonard Adlman ont publié leurs chiffrement RSA qui est compose d'une paire de clés, une clé publique pour le chiffrement des données et une clé privée pour les déchiffrer.

L'algorithme de chiffrement RSA repose sur des principes mathématiques telle que la congruence sur les entiers, le petit théorème de Fermat, Indication d'Euler, Algorithme d'Euler étendu et le théorème de Bachet-Bezout et d'autre, pour obtenir des fonctions à sens unique a porte dérobée ainsi le calcul de la paire de clés.

2.7.2 Preuve de RSA

La propriété a la base de la méthode RSA dérive directement de petit théorème de Fermat. Soit k un nombre entier tel que $k-1$ soit divisible par$p-1$. Il existe un entier K tel que $k = 1 + K(p-1)$.

$$X^k - X = x\left[\left(x^{p-1}\right)^k - 1\right] \tag{2.19}$$

D'après le petit théorème de Fermat, cette expression est nulle dans $\mathbb{Z}\ /p\mathbb{Z}$. ce qui signifie que $x^k - x$ est divisible par p . Si $k-1$ est divisible par $(p-1)(q-1)$ alors, d'après ce qui précède, pour tout x, $x^k - x$ est divisible par p et q donc par le produit $p.q$ puisque p et q sont premiers. On en déduit que, pour tout x, $x^k - x$ est divisible par$p * q$.[13]

2.7.3 L'arithmétique pour RSA

Pour un entier n, sachant qu'il est le produit de deux nombres premiers, il est difficile de retrouver les facteurs p et q tels que $n = p * q$. Le principe du chiffrement RSA, chiffrement à clé publique, repose sur cette difficulté. Dans cette partie nous mettons en place les outils mathématiques nécessaires pour le calcul des clés publique et privée ainsi que les procédés de chiffrement et déchiffrement RSA.

Création des clés

1. Choisir deux grand nombres premier p et q distincts.
2. Calculer $n = p * q$ (module de chiffrement).
3. Calculer $\varphi(n) = (p-1) * (q-1)$ (la valeur de l'indicatrice d'Euler).
4. Choisir e tel que $1 \prec e \prec \varphi(n)$ et $pgcd(e,\varphi(n))$=1 (exposant de chiffrement).

5. Calculer l'entier naturel d tel que $e * d\,(mod\varphi\,(n)) = 1$, $1 \prec d \prec \varphi\,(n)$ d peut ce calculer par l'algorithme d'Euclide étendu.

2.7.3.1 Génération des clés

Pour générer les clés RSA (clé privée et clé publique) Bob doit générer indépendamment deux nombres premières, ainsi, calculer le module de chiffrement

$$n = pq$$

Pour cela Bob il doit choisir un entier e tel que :

$$1 < e < \varphi(n) = (p-l)(q-1)\ et\ le\ pgcd(e,(p-l)(q-1)) = 1.$$

Un entier naturel d calculer par Bob tel que

$$1 < d < (p-1)(q-1) et\ de \equiv 1\ mod\ (p-1)(q-1).$$

Le couple (e, n) est la clé publique pendant que la clé privée est représentée par le couple (d, n).

Chiffrement de message : notons M le message claire telle que $M \prec n$

Chiffrement RSA : $E(M) = M^e\ mod\ n$.

Déchiffrement RSA : $D(M) = M^d\ mod\ n$.

Exemple 1.2.5

— $p = 5$ e $q = 17$

— $n = p * q = 85$

— $\varphi\,(n) = (p-1) * (q-1) = 64$

Choix d'un exposant et calcul de son inverse

1. choisir un exposant e telle que $pgcd(e, \varphi\,(n)) = 1$
2. calculer l'inverse d de e module $\varphi\,(n)$: $d * e \equiv 1$ mod$(\varphi\,(n))$ ce calcule ce fait par l'algorithme d'Euclid étendu

Si e=5 on a bien $pgcd(e, \varphi\,(n)) = 1$ $pgcd(5, 64) = 1$.

On applique l'algorithme d'Euclide étendu pour calculer les coefficients de Bézout correspondant a$pgcd(e, \varphi\,(n))$

On a 5*13+64 (-1)=1 donc $5 * 13 \equiv 1$ mod(64) et l'inverse de e modulo $\varphi\,(n)$ est d=13.

clé publique	clé prive
(n,e)=(85,5)	d=13

TABLE 2.1 – Les clés de chiffrement.

2.7.3.2 Chiffrement du message

Pour notre exemple, c'est Bob qui veut envoyer un message secret à Alice d'où le processus passe comme suit :

1. Bob utilise la clé public de Alice pour crypter son message .
2. Alice reçoit le message crypte et le déchiffre grâce a ça clé privée.

Message

Le message est un entier m telle que $0 \leq m \leq n$

Bob veut envoyer le message m=10.

Par l'algorithme d'exponentiation rapide Bob va calculer le message chiffre

$X \equiv m^e$ mode n

$10^2 \equiv 100 \equiv 15$ (mod 85)

$10^4 \equiv (10^2)^2 \equiv 15^2 \equiv 225 \equiv 55$ (mod 85)

$10^5 \equiv 10^4 * 10 \equiv 55 * 10 \equiv 550 \equiv 40$ (mod 85)

d'où le message chiffre est X=40.

2.7.3.3 Déchiffrement

Pour déchiffrer le message Alice elle doit utilise de ça part ça clé privée

$m \equiv X^d$ mode n

$X^d \equiv 40^{13}$ (mod 85)

$40^2 \equiv 1600 \equiv 70$ (mod 85)

$40^4 \equiv (40^2)^2 \equiv 70^2 \equiv 4900 \equiv 55$ (mod 85)

$40^8 \equiv (40^4)^2 \equiv 55^2 \equiv 3025 \equiv 50$ (mod 85)

$40^{13} \equiv 40^8 * 40^4 * 40 \equiv 50 * 55 * 40 \equiv 10$ (mod 85)

Le résultat et le message envoyé par Bob$X = 10$.

2.7.4 Inviolabilité de RSA

Dans la pratique, les nombres $n = pq$ utilisés ont plusieurs centaines de chiffres. L'inviolabilité de la méthode RSA vient de la difficulté de factoriser n pour en déduire$p$ et q . Casser ce code est une question de factorisation .

Si une fois les clés de codage RSA sont construites, il faut vérifier si notre système est fiable c.à.d si la clé privé reste secréte.

Le décodage RSA est basé sur le problème de factorisations des entiers. Dans le codage RSA qui utilise les nombres premiers avant tout factorisation, il faut vérifier si les deux nombres p et q sont premiers. Pour cela certains algorithmes classiques qui consistent à diviser le nombre par tous les nombres inférieurs, mais sont très gourmands en terme de mémoire.

Le test de Miller-Rabin est une méthode probabiliste qui cherche si un nombre est premier avec une probabilité d'erreur arbitrairement faible .

La factorisation des entiers est un problème très délicat en mathématique. De nombreux algorithmes sont proposés selon les tailles des nombres. On peut citer l'algorithme de Pollard Roh pour les petits nombres, les courbes elliptiques pour les nombres de quelque dizaines de chiffres et aussi le crible quadratique, actuellement l'un des algorithmes les plus puissant.

Enfin, signalons que l'algorithme, bien qu'assez sûr, est assez lent. Dans la pratique, il sers le plus souvent à transmettre une clé servant à déchiffrer un message codé selon une autre méthode plus rapide, typiquement les méthodes de chiffrement (DES, AES, IDEA..ect).[13]

2.7.5 Conseils d'utilisation du RSA

Pour une bonne sécurité pour le chiffrement RSA, il faut respecter certains règles telle que :

— Ne pas chiffrer des blocs court.
— $(p-1)(q-1)$ doit être grand (grand facteur premier).
— utiliser n très grand.
— Ne pas utiliser de n communs à plusieurs clés.
— Si (d, n) est compromise ne plus utiliser n.

2.8 Casser RSA

2.8.1 Algorithme NFS (Numbre Field Sieve)

Le « Numbre Field Sieve » est un algorithme le plus rapide pour la factorisation des entiers. Il consiste à déterminer des couples de nombres tells que leurs carrés soient congrus modulo m, $x^2 \equiv y^2\,[m]$ pour obtenir un produit $(x-y)*(x+y)$ multiple de .

L'étape la plus délicate pour cet algorithme est de trouver x et y, l'idée et de trouver deux nombres x et y telle que leurs carrés soient proche de m de sort que le résultat modulo m soit petit, d'où on peut trouver les petits nombres qui le dévise.[13]

Exemple 1.2.6 pour un entier $m = 7429$

on va choisir un entier proche à sa racine carrée .

$79^2 \equiv -2^2 3^3 11^1\,[7429]$

$80^2 \equiv -3^1 7^3\,[7429]$

$81^2 \equiv -2^2 7^1 31^1\,[7429]$

$82^2 \equiv -3^1 5^1 47^1\,[7429]$

$83^2 \equiv -2^3 3^3 5^1\,[7429]$

$84^2 \equiv -373\,[7429]$

$85^2 \equiv -2^2 3^1 17^1\,[7429]$

$86^2 \equiv -3^1 11\,[7429]$

$87^2 \equiv 2^2 5^1 17\,[7429]$

$88^2 \equiv 3^2 5^1 7^1\,[7429]$

$89^2 \equiv 2^2 3^1 41^1\,[7429]$

$90^2 \equiv 11^1 61^1\,[7429]$

On à deux couples dont le produit donne des petits facteurs avec seulement des puissance paires

$(79 * 86) = (2 * 3 * 11)^2\,[7429]$

$(87 * 88) = (2 * 3 * 5 * 7)^2\,[7429]$

Si on pose $x = 87 * 88$

$y = 2 * 3 * 5 * 7$

D'après ce qui précède le produit des deux nombres $87 * 88 \mp 2 * 3 * 5 * 7$ est un multiple de 7429 alors le pgcd de ces nombres et 7429 et 17 437

d'où $7429 = 17 * 437$[13].

2.8.2 Faiblesses des clés actuelles

Le développement des méthodes actuelles de décryptage pour des clé à certaine nombre de chiffres comme le crible quadratique, on note que la taille des clés décrypté récemment sont de 174 chiffres en 2003, 176 chiffres en début de 2005 et 200 en mai 2005.[13]

2.9 Application du chiffrement RSA sur un texte

Message clair

```
tapez le text à chiffrer:'Top secret'
```

FIGURE 2.10 – Message a chiffrer.

Représentation en entier de message

```
Integer representation: 84  111  112   32  115  101   99  114  101  116
```

FIGURE 2.11 – La représentation en entier de message.

Deux nombres premières choisis

```
p =

    71

q =

    59
```

FIGURE 2.12 – Deux nombres premières.

L'algorithme présente le chiffrement de la représentation en entier

```
text chiffré: 2055  2017  1613  3445   268  3996  2640  2827  3996  2588
```

FIGURE 2.13 – Le chiffrement de texte.

Texte original

```
Message Restoré :       'Top secret'
```

FIGURE 2.14 – Le message originale.

2.10 Conclusion

Dans ce chapitre nous avons présenté de manière détaillée les notions théoriques nécessaires pour réaliser et implémenter l'algorithme de chiffrement symétrique DES et l'algorithme de chiffrement asymétrique. Pour chacun des deux algorithmes nous avons fait une implémentation sous Matlab et nous avons présenté les résultats d'application sur un texte ASCII.

Chapitre 3

Application au cryptage / décryptage de la parole

3.1 Introduction

De nos jours les technologies de télécommunication interviennent sur touts les domaines, l'utilisation massive des moyens de communication vocale modernes via internet et ou via la téléphonie mobile exige la sécurisation des échanges. La sécurisation consiste à effectuer des transformations sur le signal original afin de le dissimuler et le rendre intelligible, toute en gardant la possibilité de le reconstitué. Pour satisfaire ces exigences, plusieurs méthodes de traitement des signaux sonores en particulier le signal parole ont été mis en œuvre par les scientifiques.

Ce chapitre est divisé en deux grandes parties. Dans la première partie, nous exposons les notions fondamentales du signal parole. Dans la deuxième partie nous présentons les résultats de l'application du chiffrement DES sur un signal parole.

Dans ce chapitre, l'algorithme DES est appliqué au cryptage et décryptage de la parole. En raison de la complexité du signal de la voix des difficultés ont été trouvées pour l'application du RSA. Nous présentons une approche hybride (DES-RSA) où l'algorithme DES est utilisé pour crypter le signal parole, et l'algorithme RSA est appliqué pour crypter la clé utilisé dans le DES.

3.2 Notions théoriques sur le son et la parole

3.2.1 Performances de l'oreille

On appelle son tout message naturel ou provoqué perçu par l'intermédiaire du sens de l'ouïe. Physiquement, le son s'analyse comme une variation de pression au voisinage de

l'oreille, cette onde de pression se propage de sa source jusqu'à l'oreille avec une célérité de V=340m/s environ.

Un son est caractérisé par :

— Son niveau ou intensité.

— Sa hauteur liée à la fréquence de son fondamental.

— Son timbre lié à sa composition spectrale.

L'intensité d'un son se mesure en$Watts/m^2$.

Le son le plus faible que l'oreille puisse entendre a une intensité Io de :$Io = 10^{-12}$ W/m^2 pour un signal de fréquence 1 kHz

Les sons les plus intenses que l'oreille puisse supporter ont une énergie de $100W/m^2$.

La gamme d'intensité s'étend donc sur 14 décades, ce qui est considérable. C'est pour cela qu'on utilise souvent une échelle logarithmique pour exprimer l'intensité d'un son en dB par rapport au niveau de référence Io précédent : I en dB = 10log(I/Io).[14]

Voici quelques exemples de niveaux sonores :

0 dB	seuil d'audition bourdonnement de moustique à 2 m
20 dB	intérieur d'un studio d'enregistrement
40 dB	conversation normale
60 dB	conversation vive
80 dB	rue bruyante
100 dB	marteau piqueur à 2 m
120 dB	réacteur d'avion à 10 m
130 à 140 dB	seuil de douleurs

TABLE 3.1 – Exemples de niveaux sonores.

Une oreille jeune est capable d'entendre des sons dans une gamme de fréquence très vaste qui va de 20 Hz à 20 kHz, soit 10 octaves ou 3 décades.[14]

Le niveau minimal de sensibilité (seuil d'audition) et le niveau maximal (seuil de douleur) ne sont pas constants sur toute la gamme de fréquences.

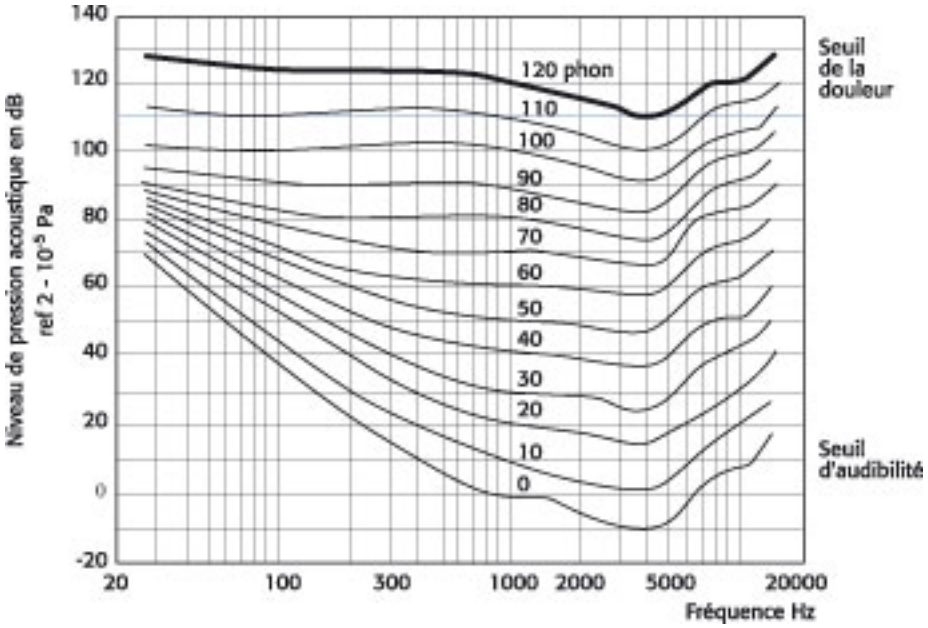

FIGURE 3.1 – Les courbes de Fletcher.

Les courbes de Fletcher montrent que l'impression de niveau sonore change avec la fréquence du signal écouté .

Cette variation de sensibilité en fonction de la fréquence explique les faits suivants :

— Le piccolo ou le triangle émergent facilement de l'orchestre.

— On entend beaucoup mieux un petit sifflet à 4 kHz qu'un gros tuyau d'orgue à 30 Hz .

— Les cordes graves d'un instrument émettent plus de puissance que les aigües, mais la sensation d'intensité est la même.

La sensibilité différentielle d'intensité liée à la variation minimale d'intensité que peut déceler l'oreille :

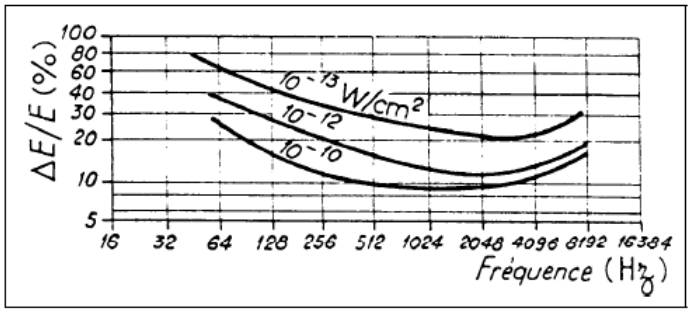

FIGURE 3.2 – Sensibilité différentielle de l'oreille.

On constate que dans la partie centrale du champ cette sensibilité différentielle est constante et vaut environ 10%.

Cela veut dire que l'oreille peut distinguer une variation d'intensité qui passe de : la valeur I1 à I2 = 0,9.I1 soit en dB : 10 log(I2/I1) = -0,5 dB

De ce chiffre découle directement un critère de qualité d'une chaîne de reproduction sonore pour laquelle les variations de la courbe de réponse devront être inférieures à ± 0,5 dB.[14]

La sensibilité différentielle de hauteur liée à la variation minimale de fréquence que peut déceler l'oreille.

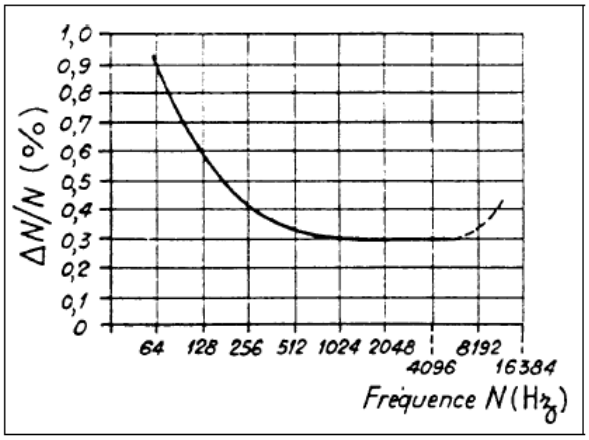

FIGURE 3.3 – Sensibilité différentielle de hauteur.

Dans la partie centrale du champ d'audition, cette sensibilité vaut 0,3 % ce qui veut dire que l'oreille peut déceler une variation de hauteur entre deux sons dont l'un est à $f1$ = 1000 Hz et l'autre à f_2 = 1003 Hz.

Cette bonne sensibilité de l'oreille aux variations de hauteur a compliqué la tâche des concepteurs de magnétophones au niveau du taux de pleurage qui devra évidemment rester en-dessous de la sensibilité de l'oreille.

Le timbre représente un son complexe formé de l'addition algébrique de plusieurs fréquences issues de la fréquence fondamentale.

Il permet d'identifier un son d'une façon unique. Deux sons peuvent avoir la même fréquence fondamentale et la même intensité, mais ne peuvent jamais avoir le même timbre. C'est grâce au timbre qu'on distingue une même note jouée au piano ou au violon, mais aussi qu'on reconnait la voix d'une personne.[15]

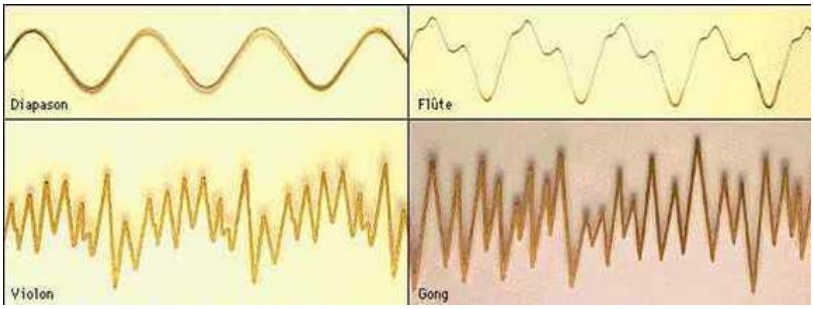

FIGURE 3.4 – Exemples de timbre sonore.

3.2.2 Son numérique

Le son numérique est obtenu à partir d'un signal analogique naturel produit par une source sonore. Ainsi, le signal analogique continu (pression acoustique) capturé est transformé en un signal numérique discret (nombre de mesures fini) à l'aide d'un capteur qui mesure les variations de pression et d'un système d'acquisition qui procède à son échantillonnage. Un système d'acquisition classique est souvent composé d'un ou de plusieurs microphones reliés à une carte son. Celle-ci envoie à un ordinateur la valeur des échantillons en vue de leur enregistrement sur un système de stockage de données (e.g. disque dur, carte mémoire, etc.). Les échantillons sont ensuite enregistrés dans un fichier dont le format varie en fonction de l'utilisation. Les formats de données brutes non compressées (e.g. WAV, AIFF, etc.) sont en général utilisés pour la manipulation des sons car ils permettent d'accéder directement aux valeurs des échantillons ce qui facilite l'analyse et les transformations appliquées sur le signal. Pour l'archivage, on privilégie les formats de données compressées sans perte (e.g. FLAC, ALAC, etc.) qui n'altèrent pas la qualité originale et qui permettent de retrouver la valeur exacte des échantillons au prix d'une étape de compression / décompression coûteuse en temps de calcul. Les formats de données compressées avec perte (e.g. MP3, OGG, AAC, etc.) sont obtenus par quantification des signaux échantillonnés et offrent un compromis entre la qualité perceptive et la taille des données. Ces formats utilisent un modèle psychoacoustique afin de réduire au maximum la taille des données permettant de représenter un signal tout en minimisant la dégradation engendrée sur la qualité sonore perçue. Ces formats offrent des taux de compression très supérieurs aux formats de compression sans perte et sont couramment utilisés pour l'archivage et les échanges sur des supports de stockage limités en espace et/ou en bande passante.[16]

3.2.2.1 Échantillonnage d'un signal audio

L'échantillonnage consiste à prélever sur un signal analogique continue $s(t)$ des échantillons représentants l'amplitude aux instants de prélèvement, on note $s\,[t]$ le signale échantillonné. Le processus d'échantillonnage consiste à enregistrer des valeurs de la fonction s(t) à des instants donnés. Dans le cas d'un échantillonnage régulier (le plus usuel), on fixe une période de temps notée T_s entre chaque mesure. L'échantillonnage consiste à associer à chaque valeur $s(nT_s)$ un Dirac localisé à l'instant $t = nT_s$. La relation entre le signal discret et le signal continu peut être exprimée par :

$$s[t] = \sum_{n=-\infty}^{+\infty} s(nT_s)\delta(t - nT_s) \tag{3.1}$$

où

$$\delta(t) = \begin{cases} 1\ si & t = 0 \\ 0 & sinon \end{cases} \tag{3.2}$$

3.2 est une fonction de Dirac discrète.

Ainsi, la fréquence d'échantillonnage $F_s = 1/T_s$ correspond au nombre d'échantillons mesurés en une seconde. Le théorème d'échantillonnage de Shannon-Nyquist initialement démontré par Whittaker définit la fréquence maximale$F_{Nyquist}$ pouvant être représentée à partir d'un signal échantillonné comme la moitié de la fréquence d'échantillonnage :

$F_{Nyquist} = F_s/2$

L'échantillonnage d'un signal périodise son spectre. En cas de sous-échantillonnage, il apparaît un phénomène de repliement du spectre ou aliasing. Le repliement du spectre engendre une distorsion du signal original qui ne peut alors plus être reconstruit sans information complémentaire.[16]

3.2.2.2 Reconstruction des signaux échantillonnés

D'après le théorème de Shannon-Whittaker , il est possible de reconstruire exactement le signal continu à partir de ses valeurs échantillonnées$\{s(nTs)\}$ $n \in Z$ si sa bande fréquence est incluse l'intervalle$[-\pi/Ts; \pi/Ts]$ en appliquant l'équation 3.3. Même lorsque cette condition n'est pas vérifiée, l'équation 3.3 permet d'obtenir une approximation du signal reconstruit. Cette reconstruction s'obtient en appliquant un filtre passe-bas sur le signal discret dont la réponse impulsionnelle s'exprime à partir d'une fonction sinus cardinal :

$$s(t) = \sum_{n=-\infty}^{+\infty} s(nT_s) \frac{sin(\frac{\pi(t-nT_s)}{T_s})}{\frac{\pi(t-nT_s)}{T_s}} = (s_{T_s} * sinc_{T_s})(t) \tag{3.3}$$

Où s_{T_s} correspond au signal s échantillonné en utilisant une période T_s. Intuitivement, ce théorème se comprend lorsque l'on observe le spectre d'amplitude d'un signal échantillonné. En effet, l'échantillonnage d'un signal périodise son spectre en ajoutant des informations dans les hautes fréquences. La reconstruction consiste donc à projeter le signal discret sur son support fréquentiel $\prod = [-F_s/2; F_s/2]$ (Bande passante définie par la fréquence de Nyquist $F_s/2$). Cette projection s'effectue en base de Fourier (domaine fréquentiel) par un produit avec la fonction porte $1_{\prod}$ qui correspond à appliquer un filtrage par la fonction sinus cardinal dans la base canonique (domaine temporel).[16] En effet :

$$F\left[\frac{sin(\frac{\pi t}{T_s})}{\frac{\pi t}{T_s}}\right] = 1_{\prod} \tag{3.4}$$

L'algorithme de Voronoï-Allebach

Dans le cas d'un signal échantillonné irrégulièrement, l'algorithme Voronoï-Allebach offre une solution pratique au problème de reconstruction des signaux. Cet algorithme repose sur le théorème de Shannon-Whittaker qui suppose le signal d'origine est limité en bande de fréquences. La reconstruction de signaux irrégulièrement échantillonnés s'obtient en appliquant un traitement itératif comprenant les étapes suivantes :

- reconstruction du signal à partir de ses échantillons en utilisant un opérateur d'approximation (e.g. interpolation linéaire, interpolation par courbe spline, interpolation de Voronoï),
- projection du signal sur le support $\prod$ en effectuant un filtrage par la fonction sincTs telle que $\prod = [-1/2T_s; +1/2T_s]$,
- projection sur $\prod$ de l'erreur d'interpolation (calculée pour les échantillons connus) puis addition avec la reconstruction précédente.

Ainsi, le signal reconstruit à l'itération i noté $s^{(i)}$ peut être formulé comme suit :

$$s^{(i)} = Vor(s - s^{(i-1)}) * sincT_s + s^{(i-1)} \tag{3.5}$$

avec s le signal irrégulièrement échantillonné (échantillons manquants mis à 0) et $s^{(0)}$ la reconstruction initiale égale au vecteur nul. Ici Vor(.) est la fonction d'interpolation de Voronoï qui affecte aux échantillons interpolés la valeur de l'échantillon connu le plus proche et pouvant être formulé comme suit :

$$Vor(s[n]) = \begin{cases} s[a]si & n < \frac{a+b}{2} \\ \frac{s[a]+s[b]}{2}si & n = \frac{a+b}{2} \\ s[b]si & n > \frac{a+b}{2} \end{cases} \tag{3.6}$$

pour$\forall n \in [a; b]$ oùa et b correspondent aux indices des échantillons connus les plus proches de l'indice n.[16]

3.2.2.3 Codage des signaux audio

L'étape de codage consiste a transformer les données a des séquences de bits qui sont généralement regroupé en 8 bit, 16 bits ou 32 bits. Elle comprend plusieurs étapes

- · La quantification qui intervient au moment de l'échantillonnage a pour but de définir un ensemble fini de représentants (dictionnaire ou quantificateur) utilisé pour enregistrer la valeur des échantillons. La quantification s'accompagne toujours d'une perte d'information liée à la taille du dictionnaire choisi. Un dictionnaire de taille plus importante permet ainsi de représenter un plus grand nombre de valeurs distinctes et s'accompagne ainsi d'un gain en précision. Cependant, il faudra un nombre de bits plus important pour que chaque élément du dictionnaire possède

un code distinct afin de rendre possible son décodage. La quantification peut aussi être appliquée sur des signaux déjà échantillonnés lorsque l'on souhaite compresser ces signaux en tolérant une perte d'information (et donc de qualité).

— · La compression ou codage de source est l'étape permettant de définir un mot (séquence de bits) associé à chaque élément ou ensemble d'éléments du dictionnaire choisi lors de la quantification. Un codage simple mais peu efficace car sans compression consiste à utiliser la représentation binaire du numéro de chaque élément unique appartenant à un dictionnaire.

— · La protection des données peut être utilisée dans certains cas lorsque le support de stockage ou le canal de transmission est sujet à des perturbations aléatoires provoquant des erreurs de décodage. Les codes correcteurs d'erreurs permettent ainsi d'éviter les erreurs de substitution (confusion entre deux éléments d'un dictionnaire) en rendant les données plus robustes par l'ajout de redondance dans le codage. Ainsi, chaque élément d'un dictionnaire possède plusieurs codes ou représentants garantissant un décodage sans erreur lorsque les altérations ne sont pas trop importantes en fonction de la capacité du code. Les disques compacts audio utilisent par exemple le codage de Reed-Solomon qui rend possible dans le meilleur des cas une reconstruction des données en cas d'effacement (lecture impossible de certaines zones en présence d'une rayure par exemple). L'utilisation des codes correcteurs a pour effet d'augmenter la taille des données et nécessite le remplacement du système de codage utilisé pour représenter certaines informations. N'entrant pas dans le cadre de notre problématique, nous avons choisi de ne pas approfondir la théorie des codes correcteurs d'erreur dans cette thèse.[16]

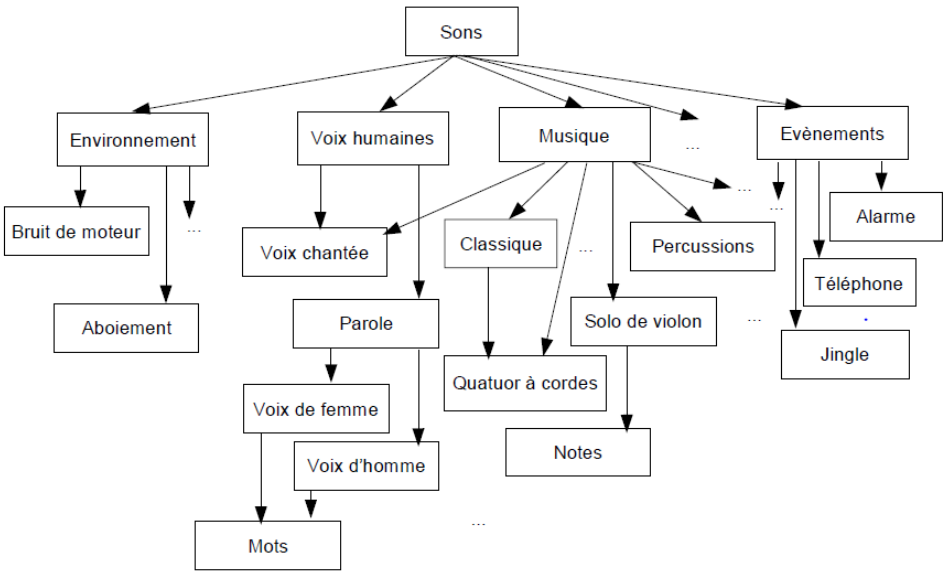

FIGURE 3.5 – Schéma de classification audio général.

3.2.3 La parole

Le traitement de la parole est aujourd'hui une composante fondamentale des sciences de l'ingénieur. Située au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques), cette discipline scientifique a connu depuis les années 60 une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications.

L'importance particulière du traitement de la parole dans ce cadre plus général s'explique par la position privilégiée de la parole comme vecteur d'information dans notre société humaine.

L'extraordinaire singularité de cette science, qui la différencie fondamentalement des autres composantes du traitement de l'information, tient sans aucun doute au rôle fascinant que joue le cerveau humain à la fois dans la production et dans la compréhension de la parole et à l'étendue des fonctions qu'il met, inconsciemment, en œuvre pour y parvenir de façon pratiquement instantanée.

les techniques modernes de traitement de la parole tendent cependant à produire des systèmes automatiques qui se substituent à l'une ou l'autre des fonctions du cerveau humain :

— Les analyseurs de parole cherchent à mettre en évidence les caractéristiques du signal vocal tel qu'il est produit, ou parfois tel qu'il est perçu (on parle alors d'analyseur perceptuel). Les analyseurs sont utilisés soit comme composant de base de systèmes de codage, de reconnaissance ou de synthèse (voir ci-dessous), soit en tant que tels pour des applications spécialisées, comme l'aide au diagnostic médical (pour les pathologies du larynx, par analyse du signal vocal) ou l'étude des langues.

— Les reconnaisseurs : L'analyse de signal nous fournis des données qui permet de décoder l'information portée par le signal vocal. Fondamentalement on a deux types de reconnaisseurs, la reconnaissance de locuteur et la reconnaissance de la parole.

La reconnaissance de locuteur a pour objectif de reconnaitre la personne qui parle. La reconnaissance de la parole a pour objectif de reconnaitre ce qui est dit.

— Les synthétiseurs ont quant à eux la fonction inverse de celle des analyseurs et des reconnaisseurs de parole : ils produisent de la parole artificielle.[17]

3.2.3.1 Analyse de parole

L'information portée par le signal de parole peut être analysée de bien des façons. On en distingue généralement plusieurs niveaux de description non exclusifs : acoustique, phonétique, phonologique, morphologique, syntaxique, sémantique, et pragmatique.

Puisque notre travail est sur le chiffrement de parole, donc on s'intéresse plus à l'analyse acoustique.[17]

Niveau acoustique

La parole apparait physiquement comme une variation de la pression de l'air causée et émise par le système articulatoire. La phonétique acoustique étudie ce signal en le transformant dans un premier temps en signal électrique grâce au transducteur approprié : le microphone (lui-même associé à un pré-amplificateur). De nos jours, le signal électrique résultant est le plus souvent numérisé. Il peut alors être soumis à un ensemble de traitements statistiques qui visent à en mettre en évidence les traits acoustiques : sa fréquence fondamentale, son énergie, et son spectre. Chaque trait acoustique est lui-même intimement lié à une grandeur perceptuelle : pitch, intensité, et timbre.[17]

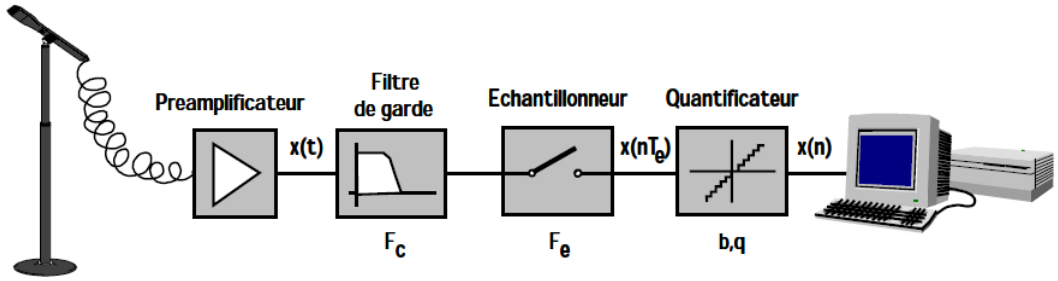

FIGURE 3.6 – Enregistrement numérique d'un signal acoustique.

Audiogramme

L'audiogramme est la représentation d'un signal audio dans le domaine temporel.

Dans le cas de la parole, le spectre peut s'étendre jusqu'à 12KHz. Il faut donc choisir une fréquence d'échantillonnageF_s= 24KHz pour satisfaire le théorème de Shannon.

Néanmoins pour les raisons économiques, on ajoute toujours un filtre de garde avec une fréquence de coupure Fc choisit en fonction de la fréquence d'échantillonnage retenue. Pour l'exemple de la téléphonie, on estime que la qualité du signal est suffisante lorsque son spectre est limité à 3400 Hz et l'on choisit la fréquence F_s= 8000Hz. Pour l'analyse, la synthèse ou la reconnaissance de la parole la fréquence choisit peut varier de 6000Hz à 16000Hz

La quantification produit une erreur de quantification qui normalement se comporte comme un bruit blanc ; le pas de quantification est donc imposé par le rapport signal à bruit à garantir. Si le pas de quantification est constant, ce rapport est fonction de l'amplitude du signal ; les signaux de faible amplitude sont dès lors mal représentés. Aussi adopte-t-onpour la transmission téléphonique une loi de quantification logarithmique et chaque échantillon est représenté sur 8 bits (256 valeurs). Par contre, la quantification du signal musical exige en principe une quantification linéaire sur 16 bits (65536 valeurs).

« .Une caractéristique essentielle qui résulte du mode de représentation est le débit binaire, exprimé en bits par seconde (b/s), nécessaire pour une transmission ou un en-

registrement du signal vocal. La transmission téléphonique classique exige un débit de 8 kHz x 8 bits = 64 kb/s ; la transmission ou l'enregistrement d'un signal audio exige en principe un débit de l'ordre de 48 kHz x 16 bits = 768 kb/s.[17]

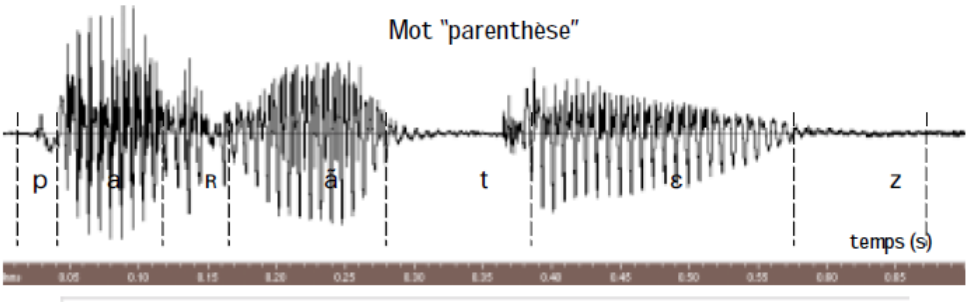

FIGURE 3.7 – Audiogramme d'un signal parole.

Transformée de Fourier à court terme

La transformée de Fourier à court terme est obtenue en extrayant de l'audiogramme une 30aine de ms de signal vocal, en pondérant ces échantillons par une fenêtre de pondération (souvent une fenêtre de Hamming) et en effectuant un transformée de Fourier sur ces échantillons.

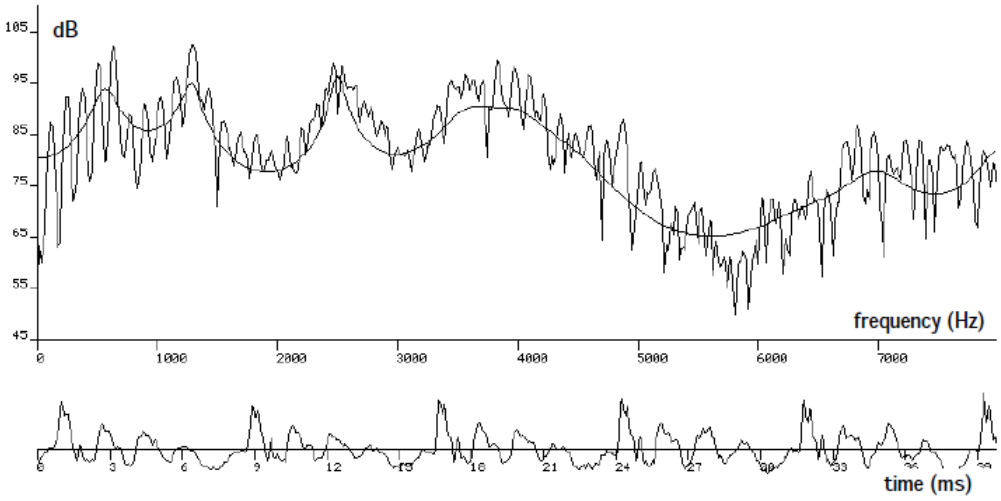

FIGURE 3.8 – Évolution temporelle (en haut) et transformée de Fourier discrète (en bas) (signaux pondérés par une fenêtre de Hamming de 30 ms).

Spectrogramme

Il est souvent intéressant de représenter l'évolution temporelle du spectre à court terme d'un signal, sous la forme d'un spectrogramme. L'amplitude du spectre y apparaît sous la forme de niveaux de gris dans un diagramme en deux dimensions temps-fréquence. On parle de spectrogramme à large bande ou à bande étroite selon la durée de la fenêtre de pondération. Les spectrogrammes à bande large sont obtenus avec des fenêtres de pondération de faible durée (typiquement 10 ms), ils mettent en évidence l'enveloppe spectrale du signal, et permettent par conséquent de visualiser l'évolution temporelle des formants. Les périodes voisées y apparaissent sous la forme de bandes verticales plus sombres. Les spectrogrammes à bande étroite sont moins utilisés. Ils mettent plutôt la structure fine du spectre en évidence : les harmoniques du signal dans les zones voisées y apparaissent sous la forme de bandes horizontales.

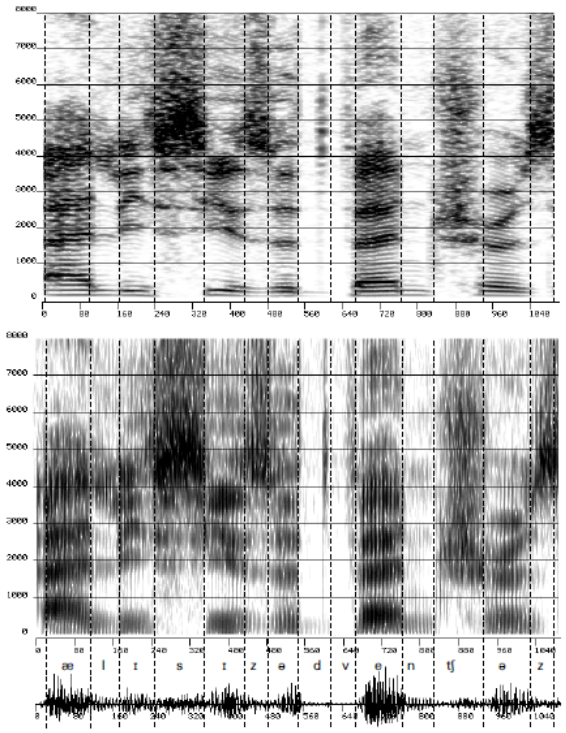

FIGURE 3.9 – Spectrogrammes à large bande (en bas), à bande étroite (en haut), et évolution temporelle de la phrase anglaise 'Alice's adventures', échantillonnée à 11.25 kHz (calcul avec fenêtres de Hamming de 10 et 30 ms respectivement).

Fréquence fondamentale

La fréquence fondamentale est l'harmonique de premier rang d'un son. Elle détermine la hauteur de son . La figure 3.10 suivante donne l 'évolution temporelle de la fréquence fondamentale de la phrase "les techniques de traitement de la parole". On constate qu'à

l'intérieur des zones voisées la fréquence fondamentale évolue lentement dans le temps. Elle s'étend approximativement de 70 à 250 Hz chez les hommes, de 150 à 400 Hz chez les femmes, et de 200 à 600 Hz chez les enfants.

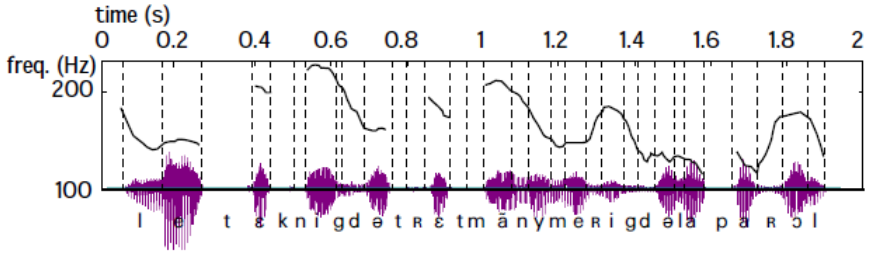

FIGURE 3.10 – Évolution de la fréquence de vibration des cordes vocales dans la phrase "les techniques de traitement numérique de la parole". La fréquence est donnée sur une échelle logarithmique.

3.2.4 Formats de fichiers audio

AIFF

Acronyme de l'anglais « Audio Interchange File Format ». Format de fichier sonore développé par Apple. Ce format n'est pas sans rappeler le format WAV de Microsoft. Les fichiers AIFF portent généralement l'extension .aif, ou .aiff. Les données sont codées en PCM big-endian sans compression. Ainsi, une piste CD Audio, codée en 16 bits, stéréo 44,1 kHz aura un bitrate de 1411,2 kbit/s. Il existe néanmoins un format compressé (AIFF ou AIFC) qui supporte une compression pouvant aller jusqu'à un rapport 1/6.[18]

ACC

Abréviation de Advanced Audio Coding, il désigne un type de compression avec pertes de données développé entre autres par la firme Fraunhofer. Il se distingue par une grande qualité d'encodage. Il est toutefois rarement compatible avec le matériel audio. Ce format pouvant contenir des GND ou DRM vous limitant dans l'utilisation d'un fichier AAC.[18]

WAV

Est l'un des tout premiers formats audio. Il est principalement utilisé pour stocker des pistes audio non compressées (PCM) comparables aux CD audio en termes de qualité. En moyenne, une minute de son WAV formaté nécessite environ 10 mégaoctets de mémoire. Les CD sont généralement numérisés en format WAV et peuvent être ensuite convertis en MP3 à l'aide d'un convertisseur audio.[18]

MP3

(MPEG Layer-3) est le format audio le plus répandu au monde. Le MP3, comme de nombreux autres formats avec perte, compresse la taille du fichier en coupant les sons non perceptibles par l'oreille humaine. À l'heure actuelle, le MP3 ne constitue pas le meilleur format existant en ce qui concerne le rapport taille du fichier et qualité du son. Cependant, étant donné qu'il s'agit du format le plus répandu et le plus compatible avec la majorité des appareils, de nombreuses personnes préfèrent sauvegarder leurs enregistrements sous ce format.[18]

FLAC

Est un format avec perte utilisé couramment. Il ne modifie pas le flux audio et le son codé avec ce format est identique à l'original. Il est le plus souvent utilisé pour la lecture audio sur les systèmes audio de pointe. Sa compatibilité de lecture sur les appareils et lecteurs est limitée. Dès lors, si souhaité, il est généralement converti en un autre format avant de le lire sur un lecteur.[18]

AAC

Est un format audio breveté possédant des capacités plus élevées (nombre de canaux, fréquences non audibles) que le MP3. Il offre en général une meilleure qualité audio tout en ayant une taille de fichier identique. AAC est actuellement l'un des algorithmes de codage avec perte ayant la meilleure qualité. Un fichier codé avec ce format peut avoir les extensions suivantes : .aac, .mp4, .m4a, .m4b, .m4p, .m4r.[18]

OGG

Est un format ouvert supportant le codage audio par divers codecs. Le Codec Vorbis est le codec le plus utilisé avec OGG. La qualité de la compression peut être comparée à celle du MP3, mais il n'est pas supporté par un aussi grand nombre de lecteurs et appareils audio.[18]

WMA

(Windows Media Audio) est un format détenu par Microsoft Corporation. Il fut à l'origine présenté pour remplacer le MP3 grâce à ses fonctionnalités de compression plus élevées. Ce fait a toutefois été discrédité par certains tests indépendants. Le format WMA supporte également la protection des données via DRM.[18]

3.3 Structure proposée

le cryptage d'un signal parole avec l'algorithme de chiffrement symétrique DES en mode CBC, commence par l'acquisition du signal puis la numérisation qui ce fait généralement par cartes son ou par logiciels. on obtiens un signal crypté par l'application du chiffrement DES sur le signal numérique (suite de valeurs). Le décryptage ce fait en appliquant l'algorithme de déchiffrement DES.

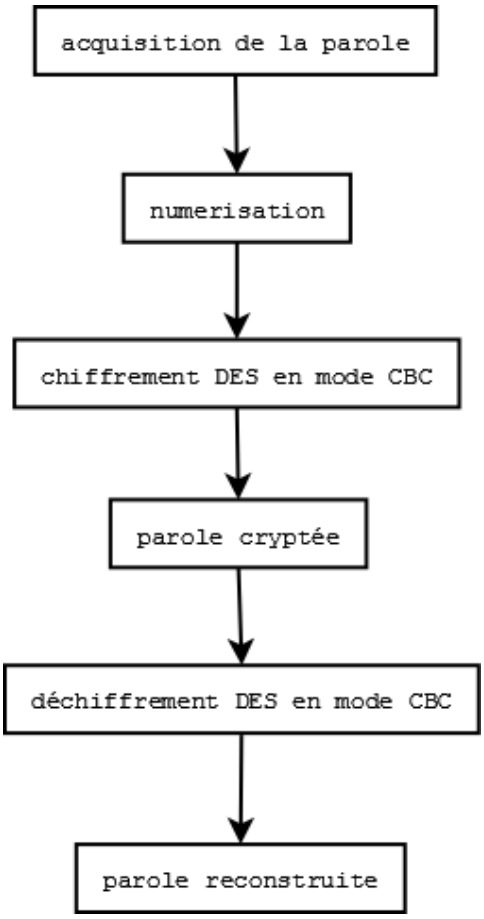

FIGURE 3.11 – Les étapes d'application du chiffrement et déchiffrement DES sur la parole.

Le schéma de la figure 3.11 présente les étapes suivis pour crypter et décrypter la parole avec l'algorithme DES.

Notre approche proposée est représentée par les deux diagrammes qui suivent :

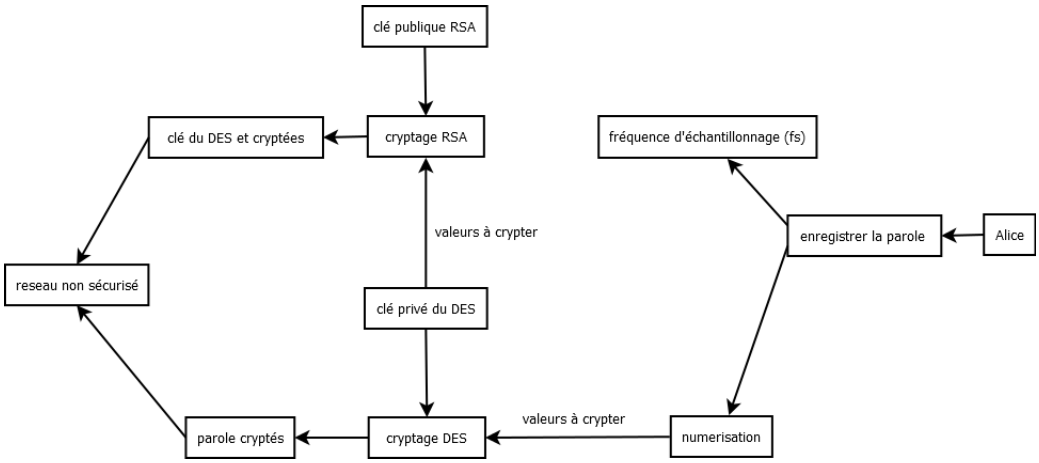

FIGURE 3.12 – Structure de cryptage niveau émetteur.

L'émetteur veut envoyer un message vocal au récepteur. Pour ce là elle va enregistrer un message parole, il va d'abord procédé à la numérisation en utilisant les méthodes citées au par-avant. Le résultat sera un ensemble d'échantillons qui vont passer par le chiffrement DES qui utilise une clé de chiffrement symétrique de 64 bits. Cette clé va être chiffrée a son tour avec le chiffrement asymétrique RSA en utilisant la clé publique générée par le récepteur. En suite le les échantillons chiffrées et la clé chiffrée serons envoyer par un réseau non sécurisé. On note que la fréquence d'échantillonnage est généralement conventionnelle (exemple : pour la téléphonie Fs=8000 Hz).

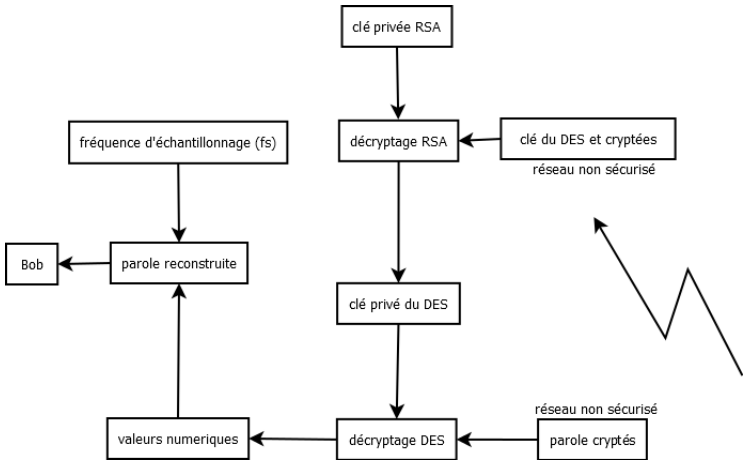

FIGURE 3.13 – Structure de décryptage niveau récepteur.

Le récepteur va récupérer la clé DES et les échantillons chiffrés sur le réseau. Il va premièrement déchiffrer la clé DES avec sa clé privé de RSA. Il va déchiffrer en deuxième lieu les échantillons avec la clé DES, puis reconstruire le message à l'aide de la fréquence d'échantillonnage.

3.3.1 Résultats de l'implémentation sur Matlab

Les résultats présentées dans les deux exemples qui suivent sont obtenus en exécutants des programmes Matlab version R2012a sur lenovo G550 qui possède les propriétés suivantes :

— processeur : Pentium(R) Dual-core CPU T4500 @ 2.30GHz.
— Mémoire installée(RAM) : 2.00 Go.
— Type du système : système d'exploitation 32 bits, (windows7)

3.3.1.1 Première exemple

Les figures ci-dessous représentent les résultats de simulation de notre programme sous Matlab.

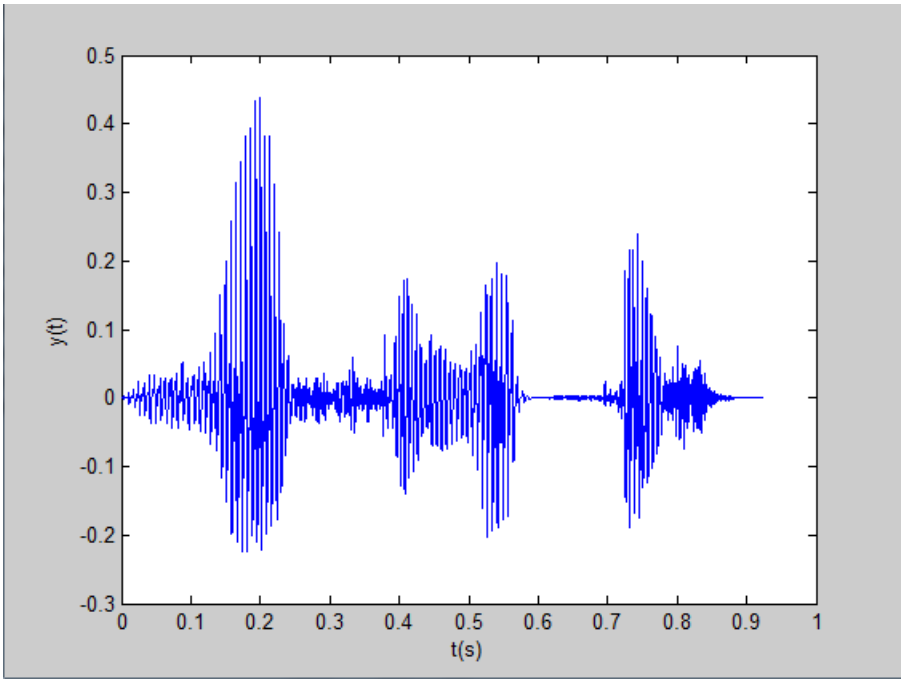

FIGURE 3.14 – Audiogramme de la parole originale y(t).

La figure 3.14 représente un signal parole original de prés de 1 seconde enregistré par téléphone puis convertis en format WAV. Cette représentation est dans le domaine temporel.

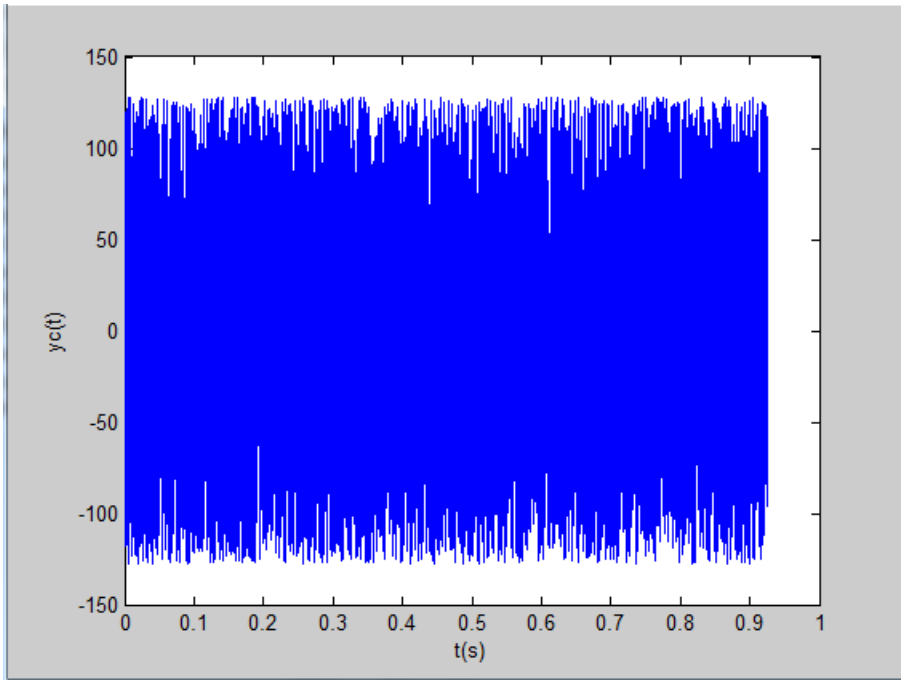

FIGURE 3.15 – Audiogramme de la parole cryptée yc(t).

La figure 3.15 donne la représentation temporelle du signal crypté par DES. On peut remarquer que la fonction de cryptage est bien accomplie du fait qu'à travers le signal crypté $yc(t)$, il est quasiment impossible d'identifier le message parole envoyé.

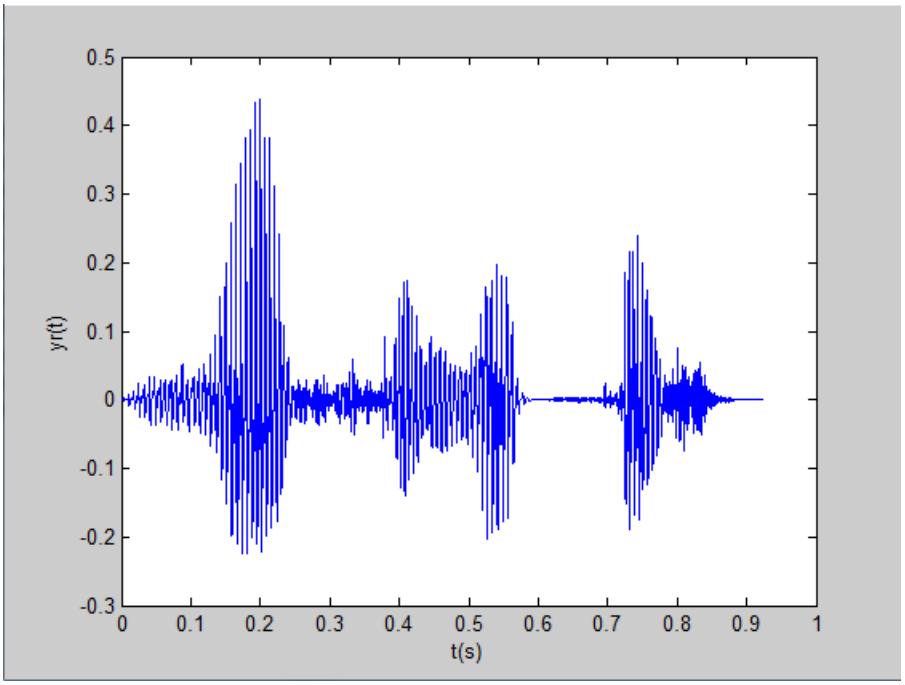

FIGURE 3.16 – Audiogramme de la parole reconstruite yr(t).

La figure 3.16 est le signal reconstruit toujours dans le domaine temporel. D'après la figure le message reconstruit correspond bien au message envoyer, cette constatation est

aussi illustrée par la figure 3.17 .

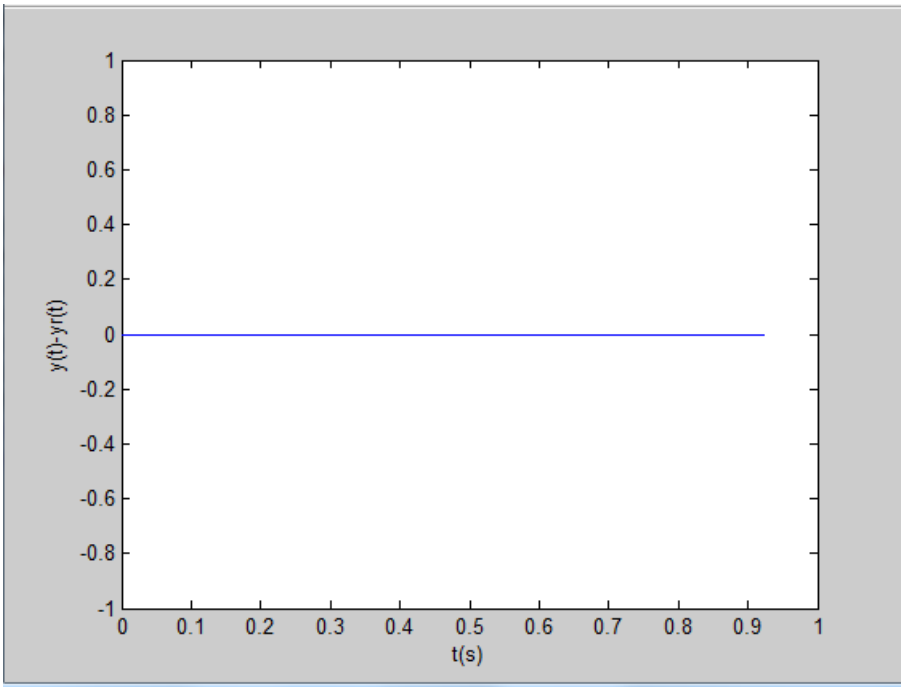

FIGURE 3.17 – L'erreur [y(t)-yr(t)].

Cette figure affiche que l'erreur entre le signal original et le signal reconstruit est nulle, donc les échantillons sont parfaitement reconstruits.

La figure 3.18 donne Le spectre fréquentiel du signal original pondéré par une fenêtre de Hamming .

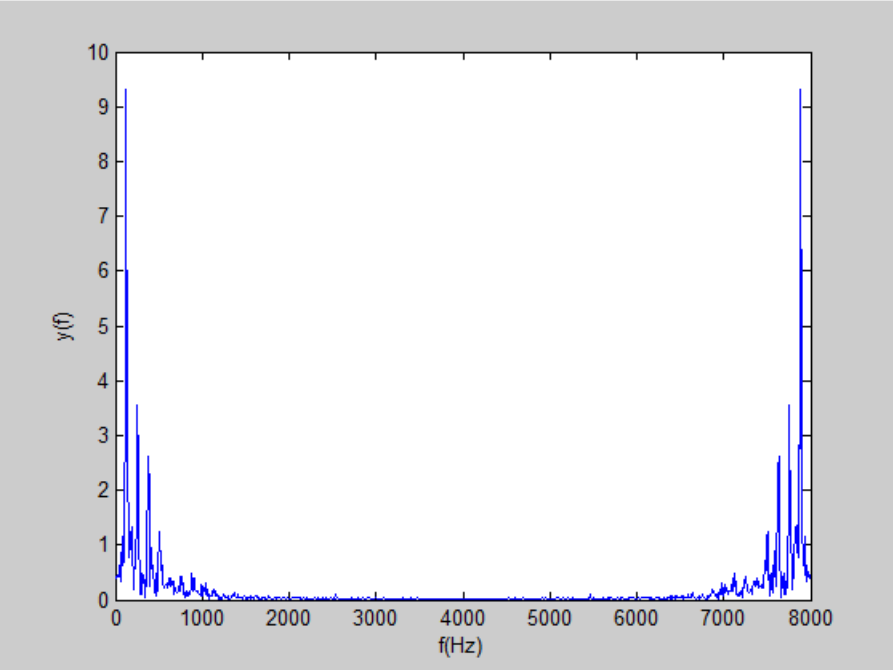

FIGURE 3.18 – La fft de la parole originale y(f).

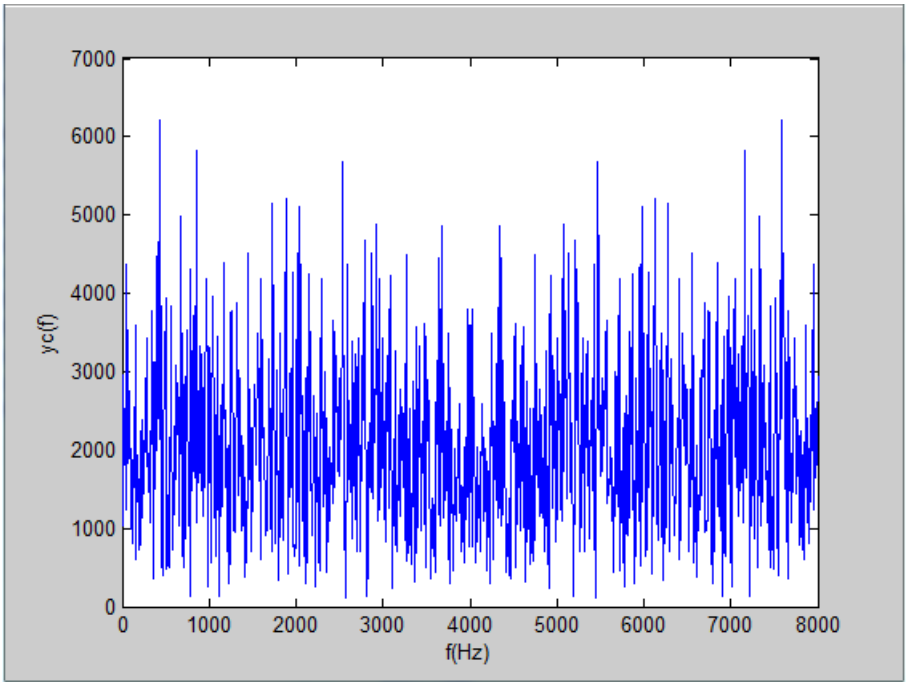

FIGURE 3.19 – La fft la parole cryptée yc(f).

La figure 3.19 montre que le signal crypté ne donne aucune information sur le signal original.

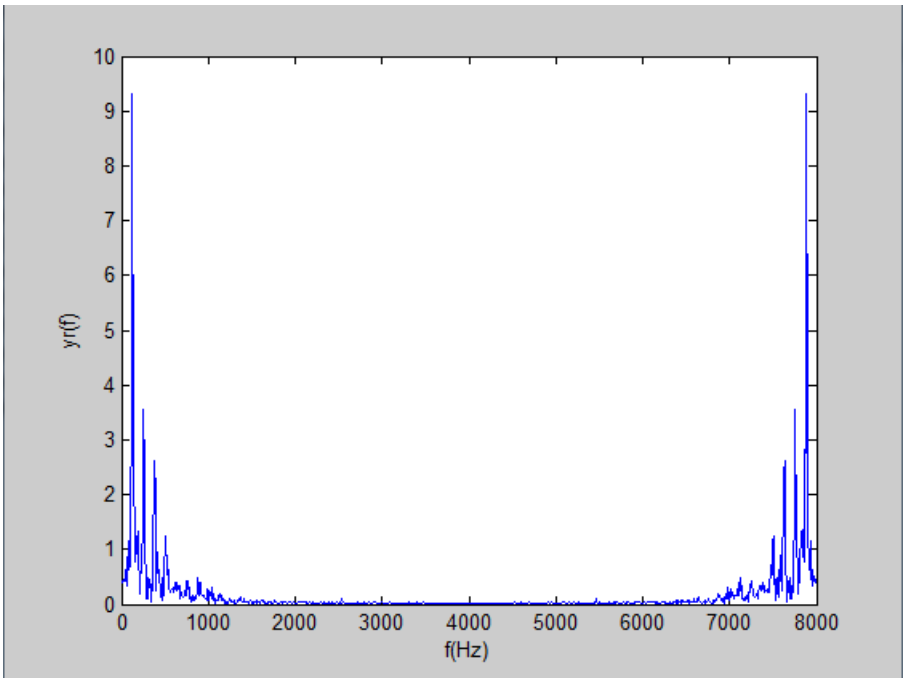

FIGURE 3.20 – La fft de la parole reconstruite yr(t).

La figure 3.20 représente le signal reconstruit, qui est de l'allure du signal original.

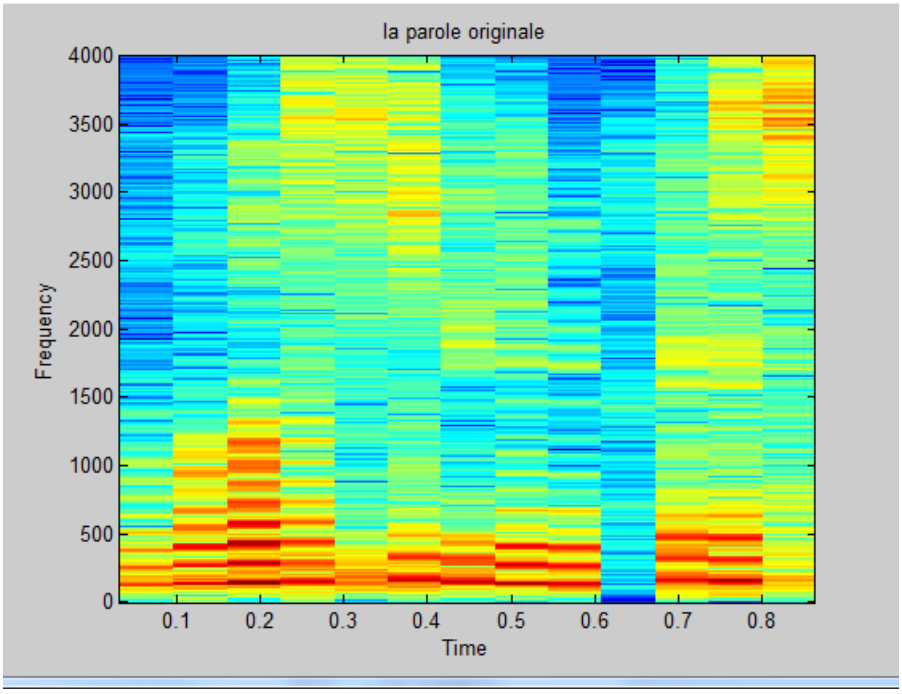

FIGURE 3.21 – Spectrogramme de signal clair.

Le spectrogramme du signal original est fréquence en fonction du temps. L'intensité est représentée par les variations de couleurs. Le rouge foncé représente les zones de fortes intensités.

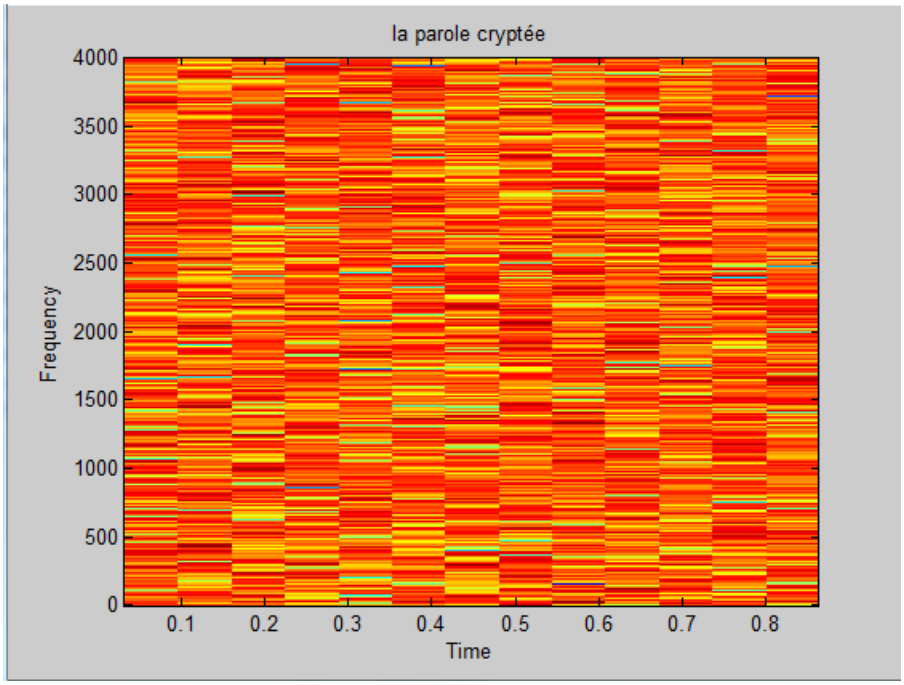

FIGURE 3.22 – Spectrogramme de signal chiffré.

Le spectrogramme du signal chiffré est très intense et on ne peut pas distinguer les informations du signal original.

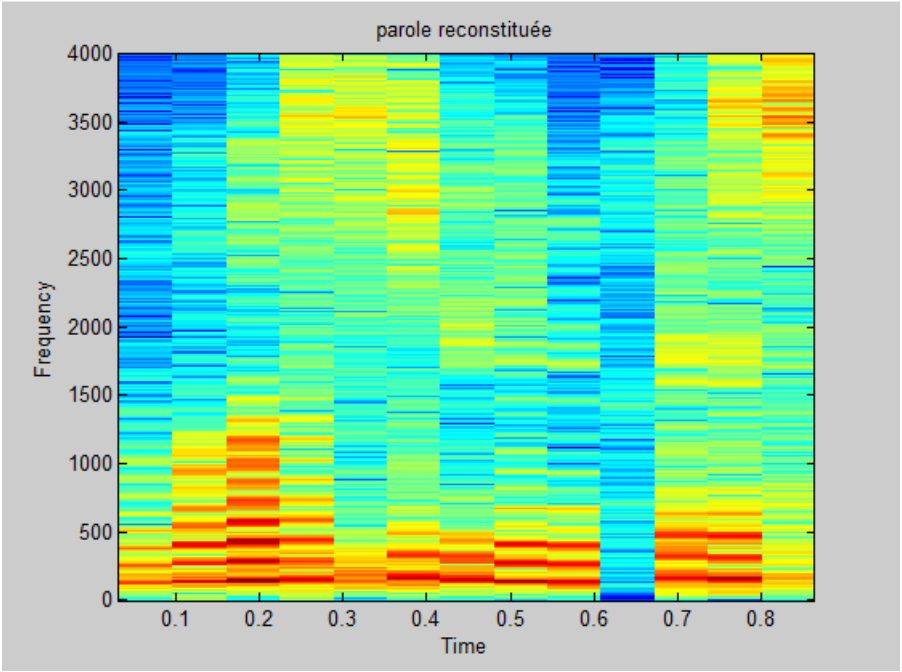

FIGURE 3.23 – Spectrogramme de signal reconstitué.

Le spectrogramme du signal reconstruit est comme celui du signal original.

Le temps pris pour cryptage et décryptage de prés d'une seconde de parole :

— Le temps d'exécution du programme du cryptage est égale à 61.3864 (s).

— Le temps d'exécution du programme de décryptage est égale à 46.8315 (s).

— Le temps d'exécution Total est égale à 108.2179(s).

3.3.1.2 Deuxième exemple

Comme deuxième exemple, nous avons choisi l'expression « bonjour » avec une voix féminine pour valoriser et affirmer les résultats de premier exemple.

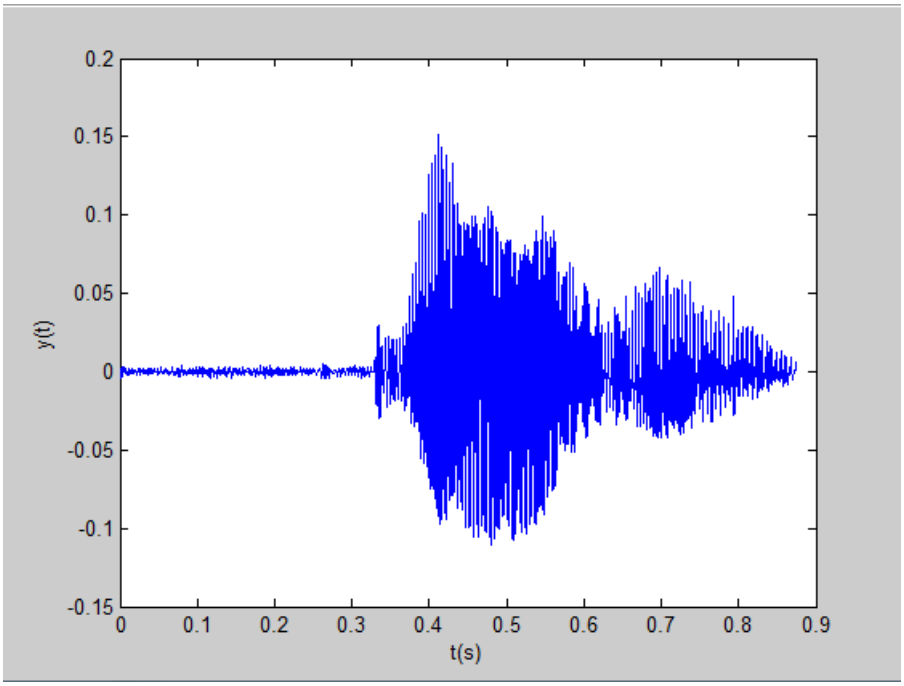

FIGURE 3.24 – Audiogramme de signal original.

La figure 3.24 présente le signal original à chiffrer avec une allure différente de celle du premier exemple.

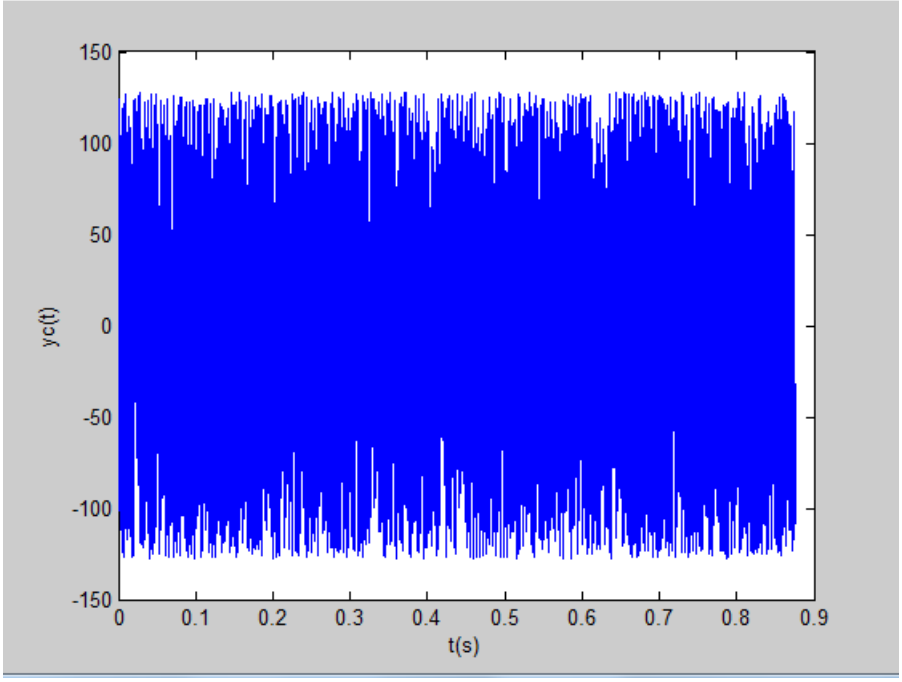

FIGURE 3.25 – Audiogramme de signal crypté.

La figure 3.25 montre le signal crypté qui cache parfaitement les caractéristiques du signal original.

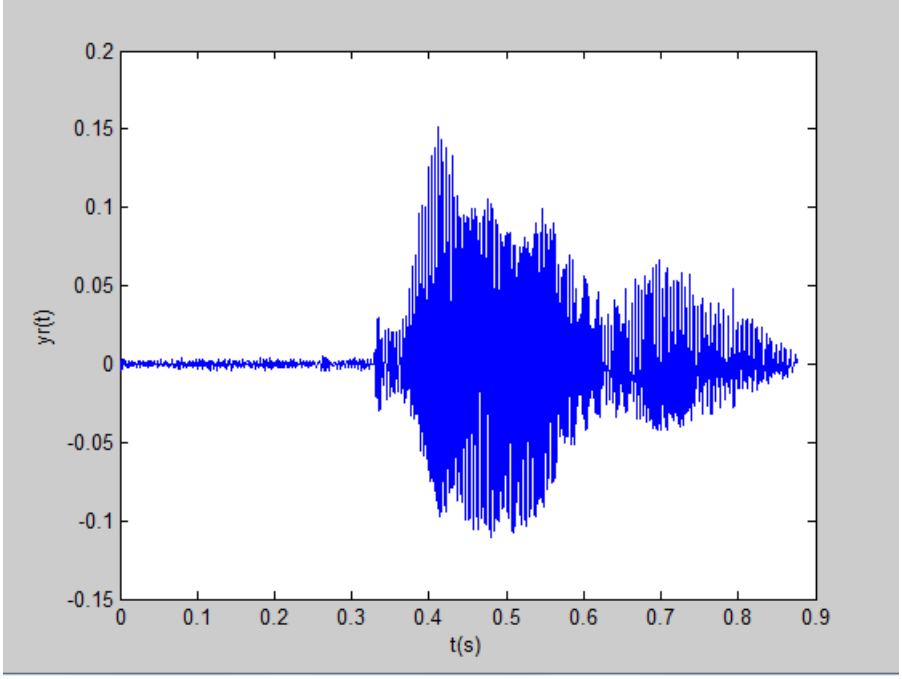

FIGURE 3.26 – Audiogramme de signal reconstruit.

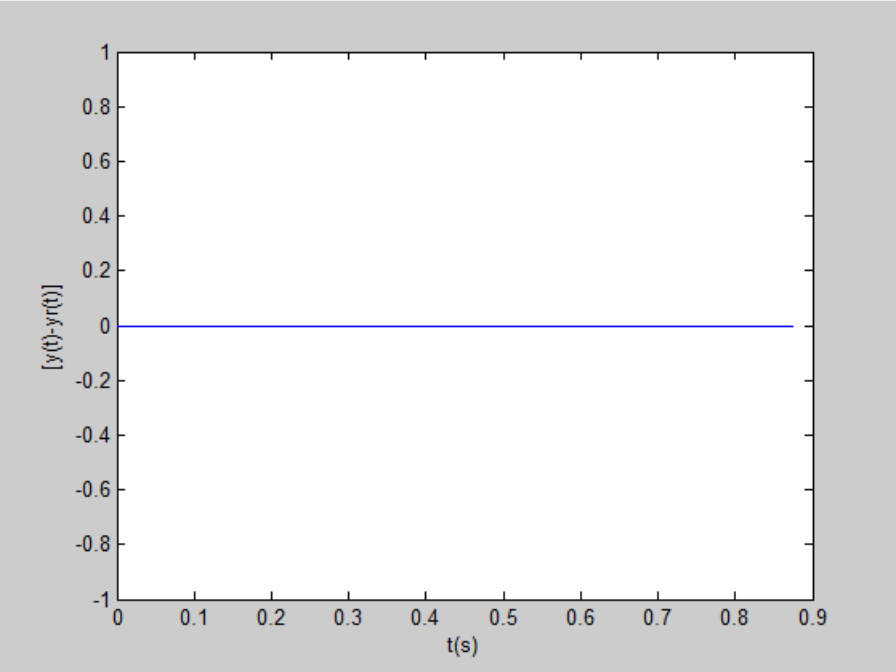

FIGURE 3.27 – L'erreur [y(t)-yr(t)].

Les figure 3.26 et 3.27 donne les résultats de la reconstruction du message original. L'erreur est quasiment nulle qui prouve que les échantillons ont tous été restitués.

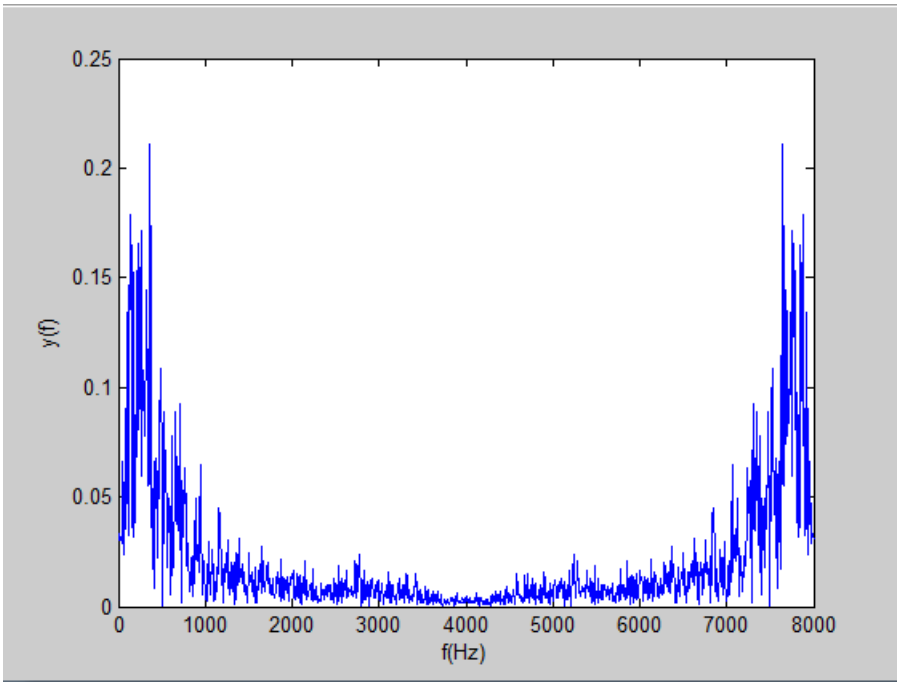

FIGURE 3.28 – La fft de signal clair.

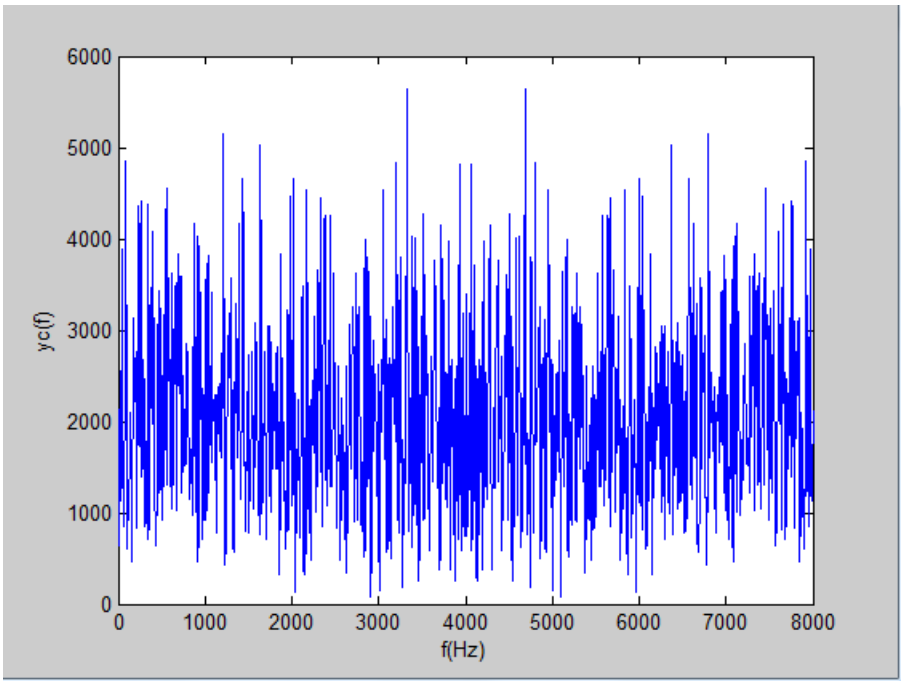

FIGURE 3.29 – La fft de signal crypté.

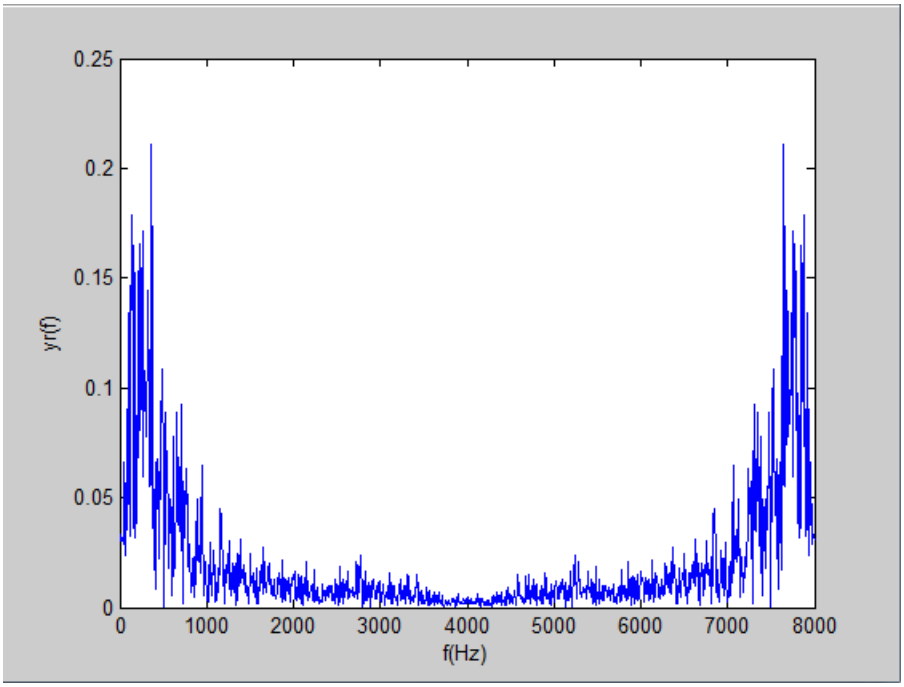

FIGURE 3.30 – La fft de signal reconstruit.

Le spectre fréquentiel de la fft pondéré sur une fenêtre de Hamming est aussi bien chiffré et ne diffuse pas d'informations sur le signal original.

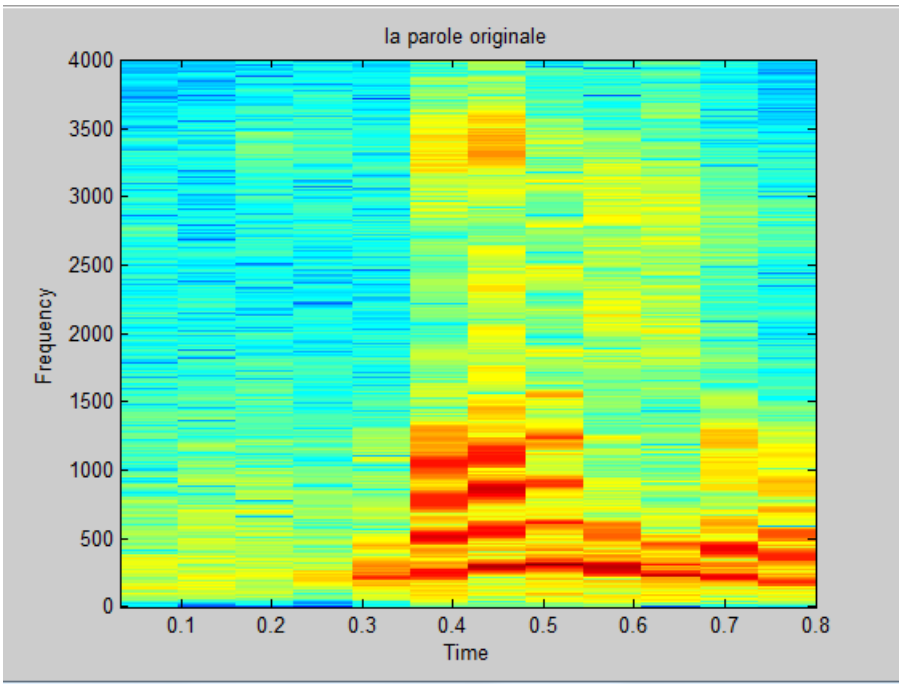

FIGURE 3.31 – Spectrogramme de signal clair.

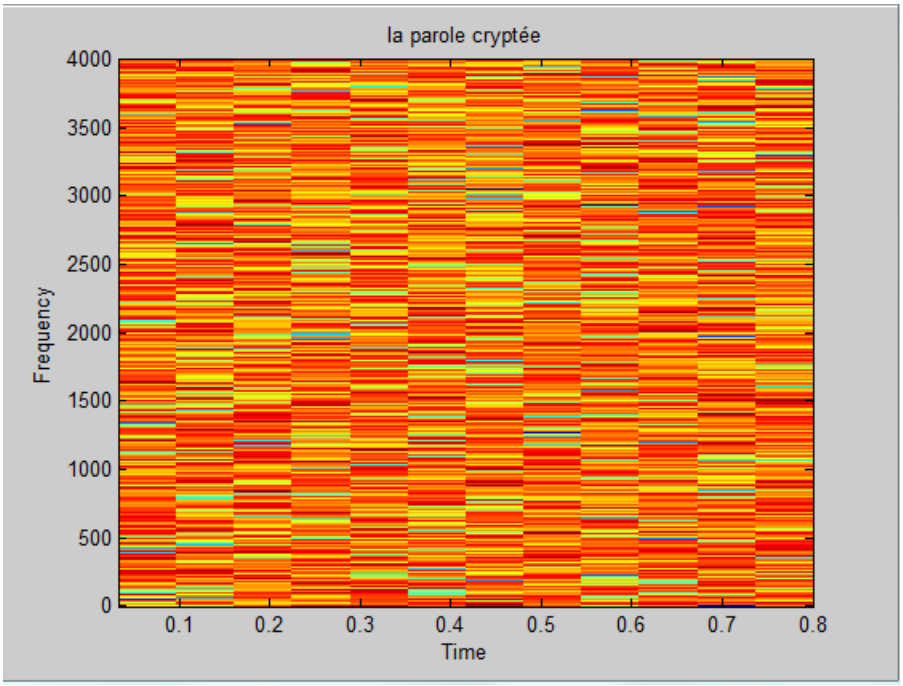

FIGURE 3.32 – Spectrogramme de signal crypté.

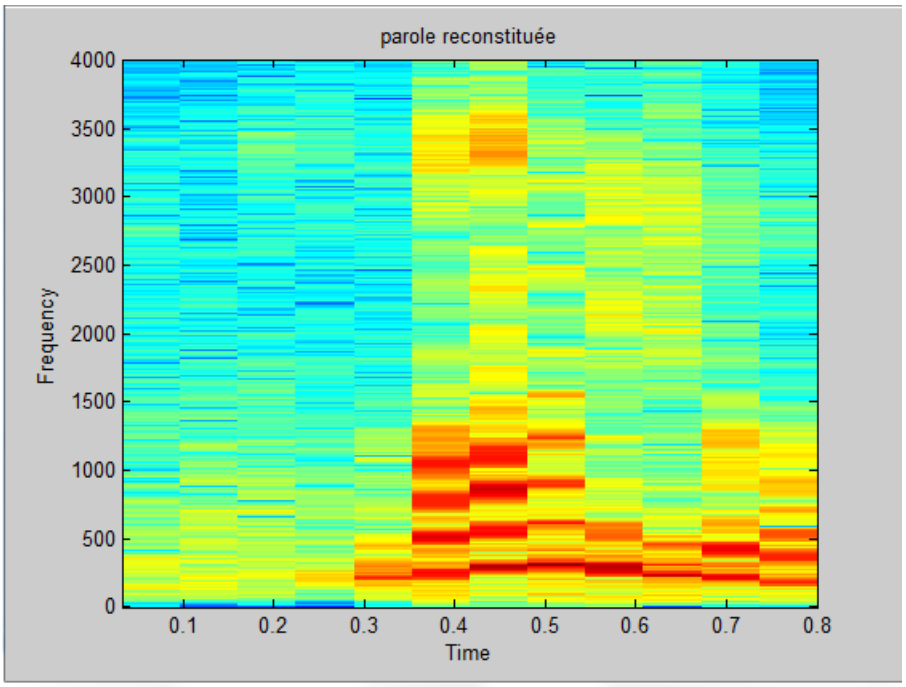

FIGURE 3.33 – Spectrogramme de signal reconstruit.

Le spectrogramme du signal chiffré cache convenablement les zones d'intensité du signal original.

Le temps pris pour cryptage et décryptage du deuxième exemple :

— Temps d'exécution du programme de cryptage =53.9919 s.

— Temps d'exécution de du programme de cryptage =43.8675 s.

— Temps d'exécution total= 97.8594 s.

Sauf que le signal reconstruit peut contenir un échantillon nul de plus que le signal original. Ce qui serait dû au bourrage du dernier bloc du message original. Cet échantillon va causer une légère augmentation de la taille mais n'affecte on aucun cas la qualité du signal reconstruit.

Notons que le temps d'exécution peut légèrement varier si le programme est exécuté sur une autre machine.

1. La vitesse d'exécution est très lente due au langage Matlab qui est un langage interprété. Le programme peut être 10 fois plus rapide avec une simulation sous C++.
2. Sous MATLAB on travail sur des fichiers audio de format wave (non compressé), Donc une très grande quantité de données. alors que en C++ on peut travailler avec des fichiers compressés mp3.

Comparaison :

Le volume d'un fichier WAV stéréo pour 1 minute échantillonné à 44kHz en 16 bit est de : 60 (secondes) * 44000 (taux d'échantillonnage) * 2 (stéréo) * 2 (16 bits = 2 octets)

= 10.56Mo.

Un morceau de musique comprimé en MP3 à 128kbps et à 44kHz a une taille de 3Mo environ (pour 3 à 4 minutes), soit environ 1Mo par minute.

Pour un fichier compressé par divers procédés et dans divers formats (MP3, OGG, ...), on donne habituellement une valeur en kbps qui est en rapport avec le taux de compression (et donc le taux de perte).

travailler avec des fichiers mp3 peut réduire le temps d'exécution des programmes 10 fois, pour illustrer cet avantage, un WAV 44Khz / 16bits / Stéréo est à 1375Kbps. Donc, un MP3 compressé à 128Kbps a un taux de compression de 11 pour 1.

3.4 Conclusion

Dans ce chapitre nous avons présenté les notions théoriques pour les traitements de signal parole, cité quelques formats de stockage et en fin nous avons présenté les résultats de chiffrement DES sur deux exemples différents de signaux parole et proposé une stratégie de chiffrement hybride DES et RSA .

Conclusion générale

L'avènement de l'informatique est a l'origine de nouveau visage de la cryptologie. La lutte sans fin entre les concepteurs des systèmes informatisés d'une part et les espions d'une autre part a donné naissance à de nombreuses applications dans le but de sécuriser les données informatiques.

Dans ce mémoire, nous avons donné une description de la cryptologie, ainsi une présentation détaillée de chiffrement a clé secréte DES et chiffrement a clé publique RSA muni d'une implémentation de DES et RSA sous Matlab pour chiffrer un texte clair.

L'objectif principal assigné à notre travail est l'implémentation de l'algorithme DES sous Matlab qui permet le cryptage et décryptage de signal parole avec la transmission des clés avec l'algorithme RSA. Le chiffrement d'une phrase " voici le teste" avec l'algorithme DES nous a permit de rendre le message intelligible.

Avec l'évolution constante des méthodes de cryptage et les systèmes informatiques, est Comme perspective du présent travail, nous pouvons citer :

- Tester d'autres méthodes de chiffrement tel que le AES, l'algorithme Russe GOST, BEA-1...
- Effectuer des comparaisons entre ces méthodes de chiffrement.
- Concevoir un système de chiffrement en temps réel.
- Hybridation pour concevoir un cryptosystème robuste, fiable.
- Approfondir dans le domaine de traitement de parole.
- Appliquer les méthodes d'automatique dans les cryptosystèmes.
- Cryptanalyse.
- Utiliser le chaos pour la génération des clés.

Bibliographie

[1] Koceila Lounici, Conception et réalisation d'un cryptosystème hybride pour la transmission sécurisé d'image, mémoire de master, UMMTO, 2016.

[2] S.Belattaf. Sécurité informatique, support de cours, UMMTO. 2016/1017.

[3] Jean-Guillaume Dumas, Jean-Louis Roch, Éric Tannier, Sébastien Varrette, Théorie des codes, Dunod, Paris, 2007.

[4] William Stallings, Cryptography and Network security, Pranciples and practice, Prentice Hall, 2006.

[5] Touiradj Ebrahimi, Franck leprévost, Bertrand warusfel, Cryptographie et sécurité des systèmes et réseaux, Lavoisier, Paris, 2006.

[6] www.comentcamarche.com.>Encyclopédie>Sécurité /législation>Cryptographie (Mai 2018).

[7] http ://www.bibmath.net/crypto/index.php, (Avril 2018)

[8] http ://dit-archives.epfl.ch/FI00/fi-sp-00/sp-00-page8.html (Avril 2018).

[9] Math.univ-lille1.fr/~bodin/fichier/ch-crypto.pdf (Mars 2018).

[10] http ://defeo.lu/in310/poly/euclide-bezout/(Mars 2018)

[11] Renaud Dumont, Cryptographie et Sécurité informatique, support de cours, Université de Liège, 2009/2010.

[12] Floriane Anstett, Les systèmes dynamiques chaotiques pour le chiffrement : synthèse et cryptanalyse, thèse de doctorat, Université Henri Poincaré - Nancy I, 2006.

[13] Hervé Lehning, cryptographie & codes secrets, Pole, Paris, 2013.

[14] Jean-Philippe Muller, le traitement numérique de signal audio, version juillet, 2001. (https ://www.robertponge.com/telechargements/ebooks/audio-numerique.pdf).

[15] https ://www.easyzic.com/dossiers/le-timbre,h27.html, (Avril 2018).

[16] Dominique Fourer. Approche informée pour l'analyse du son et de la musique, thèse de doctorat, Bordeaux I, 2013.

[17] Thierry Dutoit, Introduction au traitement automatique de la parole, cours , Faculté de polytechnique de Mons, 2000.

[18] https ://online-audio-converter.com/fr/help/audio_formats, (juin 2018)

[19] Daniel Barsky, Cryptographie Télécom, support de cours,2006/2007.