TABLE DES MATIERES

NOTATIONS

1. Minuscules latines

a	Dividende
b	Diviseur
e_k	Résultats possibles
f	Fonction
k_i	Ensemble des clés
m_i	Ensemble des messages clairs
m_w	Médium tatoué
p	Nombre premier
q	Nombre premier
q	Quotient
r	Reste
p_i	Probabilité
mod	Modulo
sig	Opération de signature privée
ver	Opération de vérification publique
ms	milliseconde
corr	corrélation
w_k	Marque

2. Majuscules latines

A	Alice
B	Bob
C	Message chiffré
D	Déchiffrement
D_{k1}	Déchiffrement avec la clé k_1
E	Eve

E	Expérience
E	Ensemble partitionné
E_{k1}	Chiffrement avec la clé k_1
E_i	Sous-ensemble
H	Entropie
I	Quantité d'information
I_c	Taille de l'image comprimée
I_o	Taille de l'image originale
K	Clé
$\mathcal{L}$	Nombre premier
M	Message clair
N	Nombre Total
O	Notation grand O
$\mathcal{O}$	Point à l'infini.
$P(x)$	polynôme
P	Probabilité
P_{ee}	Probabilité d'erreur d'extraction
S	alphabet
X	Variable aléatoire
X	Source

3. Minuscules grecques

α_i	Entier positif
β	Clé publique
δ	Signature
$\varphi(n)$	Fonction indicatrice d'Euler
γ	Message
$\mathcal{L}$	Nombre premier
ω	Oméga

4. Majuscule grecque

Π	Pi

5. Abréviations

3D	3 Dimensions
ACP	Analyse en Composante Principale
ADPCM	Adaptive Differential Pulse Code Modulation
ADSL	Asynchronous Digital Subscriber Line
AES	Advanced Encryption Standard
AES-I	AES Improved
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
BB84	Bennett et Brassard.
BPP	Bit par Pixel
BTP	Back Trraffic Protection
BRR	Bit Rate Reduction
CA	codage arithmétique
CBC	Cipher Block Chaining
CD-ROM	Compact Disk
CDMA	Collision Detection Multiple Access
CFB	Cipher FeedBack
CPU	Central Processor Unit
CS	Cryptage Selectif
DES	Data Encryption Standard
DCT	Discrete Cosinus Transform
DPCM	Differential Pulse Code Modulation
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
DWT	Digital Wavelet Transform
ECB	Electronic CodeBook

ECC	Elliptic Curve Cryptosystem
ECDLP	Elliptic Curve Discrete Logarithm Problem
EQM	Erreur Quadratique Moyenne
FFT	Fast Fourier Transform
FIPS	Federal Information Processing Standard
GIF	Graphics Interchange Format
GNU	GNU is Not Unix
GPA	Générateur Pseudo-Aléatoire
GPL	General Public Licence
GSM	Global System Mobile for communication
HTML	HyperText Markup Language
IAX	Inter-Asterisk eXchange
ICMP	Internet Control Message Protocol
IDC	Insertion de Données Cachées
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPAD	Inner Pudding Data
ISO	International Standard Organization
JPEG	Joint Photographic Expert Group
Kbps	Kilobit par seconde
KLT	Kharuen Loève Transform
LOCO	LOw COmplexity
LSB	Least Significant Bit
LSFR	Linear Feedback registers
LZW	Lempel Ziv Welch
MAC	Message Authentication Code
MATLAB	MATrix LABoratory
Mbps	Mégabit par seconde
MD5	Message Digest 5
MED	Median Edge Detection
MPEG	Moving Picture Expert Group

MSE	Mean Square Error
NBS	National Bureau of Standards
N & B	Noir & Blanc
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OFB	Output-FeedBack
OMA	Open Mobil Alliance
OPAD	Outer Pudding Data
OS	Operating System
PABX	Private Automatic Branch eXchange
PFS	Perfect Forward Secrecy
PGCD	Plus Grand Commun Diviseur
PGP	Pretty Good Privacy
PPCM	Plus Petit Commun Multiple
PSNR	Peak Signal to Noise Ratio
QoS	Quality of Service
RAM	Random Access Memory
RDRL	Registres de Décalage à Rétroaction Linéaire
RGB	Red Green Blue
RIPE	RACE Integrity Primitives Evaluation
RIPE-MD	RIPE Message Digest
RLC	Run Lenght Coding
RLE	Run Lenght Encoding
RNIS	Réseau Numérique à Intégration de Service
RSA	Rivest Shamir Adleman
RTC	Réseau Téléphonique Commuté
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RVB	Rouge Vert Bleu
SCA	Side Chanel Attack
SHA	Secure Hash Algorithme
SHS	Secure Hash Standard

SIP	Session Initiation Protocol
SVD	Singular Values Decomposition
SVH	Système Visuel Humain
TCD	Transformée en Cosinus Discret
TCP	Transport Control Protocol
TFD	Transformée de Fourier Discrète
TFR	Transformée de Fourier Rapide
TLS	Transport Layer Security
ToIP	Telephony over Internet Protocol
TVHD	Television High Definition
UDP	User Datagram Protocol
VLC	Variable Lenght Coding
VoIP	Voice over Internet Protocol
VQ	Vector Quantization
WAP	Wireless Application Protocol
WPAN	Wireless Personal Area Networks
$wPSNR$	weighted PSNR
XP	eXPerience

6. Notations spéciales

$\equiv$	Relation de congruence
$\oplus$	Opération xor
$\not\emptyset$	Blancs
$\#E$	Cardinal de E
A\B	A sachant B
$E_k(M) = C$	Fonction de chiffrement
$D_k(C) = M$	Fonction de déchiffrement
$\mathbb{F}_{p^r}$	Corps fini de cardinal p^r
$\hat{G}$	Matrice
$log_P\ (Q)$	Logarithme de Q en base P
$\|x\rangle$ et $\|y\rangle$	Polarisation linéaire orthogonale d'un photon

$\mathbb{Z}\|\boldsymbol{n}\mathbb{Z} = \mathbb{Z}_{\boldsymbol{n}}$	Ensemble quotient
♣	Début de démonstration
♦	Fin de démonstration
$\mathbb{B}$	Alphabet binaire
$\mathbb{R}$	Ensemble des nombres Réels
$F_B[m]$	Coefficient
$\widetilde{F_B}[m]$	Variable Quantifiée
$\mathcal{N}$	Loi normale
∤	Ne divise pas

INTRODUCTION ET POSITION DU PROBLEME

Cette thèse contribue à la mise en œuvre d'un système de communication et de la sécurisation de l'information. Les informations comme les images et la voix subiront donc plusieurs séries de traitements afin d'atteindre un haut niveau de confidentialité. La philosophie que nous avons adoptée dans notre recherche consiste à développer chacun de ces domaines par chapitre.

L'objectif de notre recherche dans cette thèse est :

- *De donner un panorama sur la sécurisation de l'information par l'utilisation de la cryptographie.*
- *D'étudier les traitements et codages des flux multimédias*
- *De proposer des algorithmes permettant de modéliser les éléments mathématiques utilisés par la cryptographie et de les optimiser pour mieux les implémenter.*
- *De simuler et de tester la sécurisation des informations à partir des modèles proposés et d'étudier la consommation des ressources informatiques mises en place dans un système de communication*

En effet, les réseaux numériques ont tellement évolué qu'ils sont devenus un mécanisme essentiel de communication. Ils permettent de transmettre toute sorte d'informations textuelles, sonores, images et vidéo. La croissance exponentielle du trafic des données multimédia est renforcée par l'apparition importante d'appareils photo numériques, l'utilisation des téléphones portables et d'autres moyens de communication.

Ces évolutions ne sont pas mises à l'écart de difficultés et engendrent en plus de nouveaux problèmes. Nous avons fait notre recherche afin d'apporter notre contribution sur la résolution de ces problèmes.

Dans le premier chapitre, afin de rendre une information confidentielle, nous allons voir qu'il existe plusieurs systèmes cryptographiques utilisés depuis des siècles, et bien d'autres encore que l'on peut utiliser jusqu'à ce jour. Toutes ces méthodes reposent sur des principes et algorithmes bien définis et surtout sur des formulations mathématiques plus ou moins complexes. On aborde alors cette section sur l'étude de la théorie des nombres ainsi que les principes de chiffrement symétrique, asymétrique et générateurs pseudo-aléatoires.

Dans le second chapitre, nous allons, d'une part, décrire les divers traitements que subissent les informations textes, images, voix et vidéo. Nous nous axons surtout sur l'aspect du codage source qui consiste en premier lieu à donner une autre représentation des données, il s'agit surtout d'une opération de compression. D'autre part, nous donnons une autre technique de sécurisation de l'information qui est la technique d'insertion de données cachées, c'est-à-dire d'insérer des données dans d'autres données. Toutes ces opérations ont pour but de donner une confidentialité, intégrité et aussi de permettre une authentification à une information.

En effet, les données transférées ou stockées peuvent être classées selon leurs importances. Elles peuvent être classées en tant que données confidentielles comme les mots de passe, les systèmes d'authentification, les données médicales, les diverses communications téléphoniques et informatiques. La transmission ou le stockage des données soulève donc un nombre conséquent de problèmes qui ne sont pas tous et encore résolus.

Dans ce chapitre, nous allons soulever des problèmes à partir des travaux de recherche effectués par d'autres chercheurs dans un domaine similaire. Diverses recherches académiques sur ce domaine ont déjà été aussi faites par plusieurs universités. Il y a donc :

- les problèmes relatifs à l'aspect sécurité et à l'authenticité des données pendant la transmission ou stockage, mais également à la réception.

- le problème concernant le temps de transfert et d'exécution des programmes

Le transfert et le stockage d'information, les systèmes d'authentification, les communications voix sur IP sont parmi les modèles de communication qui sont en tout temps confrontés à ces types de problèmes. Pour cela, notre contribution, afin de soulever ces problèmes, est d'utiliser des systèmes de sécurisation robuste et adaptés aux environnements de transfert et de stockage de ces informations.

Dans le troisième chapitre, nous proposons des algorithmes de calculs que nous allons adopter pour effectuer chaque processus de sécurisation et de transfert de données. Pour cela, nous nous basons sur le schéma et principe suggéré par les organismes de normalisation et ainsi nous pouvons mettre en œuvre des programmes facilement implémentable dans un environnement informatique à ressources limitées.

Nous consacrons aussi des paragraphes pour l'étude de la technologie de la Voix sur IP (VoIP pour Voice over Internet Protocol), en l'occurrence, le comportement de ce système par rapport à son utilisation par les clients et les ressources matérielles mises à disposition.

Pour la simulation, en premier lieu, nous avons utilisé le transfert sécurisé d'images, un domaine récemment apparu dans le domaine de transfert sécurisé d'information. Notre approche diffère donc des recherches antérieures sur le fait que nous avons utilisé la combinaison de trois techniques, à savoir la compression, le tatouage et la cryptographie. Le choix des méthodes utilisés se base sur un traitement sélectif des éléments composants l'image afin d'optimiser le temps d'exécution des programmes tout en assurant sa fiabilité.

Une nouvelle technique a été utilisée, dans une seconde approche, afin de sécuriser des données stockées. En effet, l'authentification est la porte qui nous mène vers une étape de la sécurité du fait qu'il est d'une nécessité de connaitre chacun des acteurs opérants dans le système. Pour cela, nous optons sur l'utilisation d'index au lieu de condensés. L'un comme l'autre utilise des fonctions à sens unique qui représentent, sous une forme très réduite, des informations représentatives des données à traiter. Ces index, protégé par des systèmes cryptographiques, seront utilisés à des fins de comparaison pour authentification.

En troisième lieu, nous apportons notre contribution sur l'étude de la sécurisation de la voix dans un réseau IP (Internet Protocol), qui deviendra, dans un futur proche, un standard universel de communication voix et vidéo dans les réseaux de Télécommunication. Comme dans tout appel téléphonique, il est d'une nécessité de chiffrer la communication pour respecter le droit et l'intimité de chaque individu. Nous faisons des sondages sur la consommation de ressources par la mise en place d'un tel système et à la sécurisation des paquets de données IP. C'est le problème majeur de ce genre de technologie qui connait actuellement différentes attaques mettant en péril tous les systèmes communicants.

Cette thèse se base donc sur la sécurisation de données qui tourne autour de l'utilisation de la cryptographie. Une discipline, qui utilise des propriétés mathématiques, et qui permet de rendre confidentielle une information afin de les rendre inintelligibles aux acteurs autres que ceux authentifiés et autorisés. Il nous a fallu pour cela utiliser et combiner d'autres méthodes comme la compression, le tatouage et l'indexation pour rendre performant l'algorithme et d'ajouter un niveau de robustesse du cryptosystème utilisé.

CHAPITRE 1
THEORIE DES NOMBRES ET CRYPTOGRAPHIE

1.1 Définitions

En cryptographie [1], [2], nous définissons les termes suivants qui sont :

- La c*onfidentialité* ou le *secret* : elle assure qu'une information secrète ne peut être accédée par des personnes non autorisées.
- Le *code* : c'est un système de symbole représentant une information mais ne référant pas à un secret.
- Le *codage* : c'est un processus utilisé en théorie de l'information, c'est l'action permettant de transformer une information en code.
- La *cryptographie* : c'est la science des codes secrets, permettant de rendre une information confidentielle à travers un canal de transmission peu sûr. Par rapport à la théorie du codage, qui consiste à protéger l'information face aux bruits non intentionnels dans le canal de transmission, la cryptographie vise à protéger l'information vis-à-vis des personnes malintentionnées.
- *Crypter* ou *chiffrer* : c'est une opération permettant de rendre confidentiel une information.
- Un *cryptosystème* : c'est un ensemble de système cryptographique permettant de crypter et décrypter une information.
- Le *texte clair* : c'est la donnée en entrée qui va être chiffrée afin de la rendre confidentielle.
- Le *texte chiffré* ou *cryptogramme* : c'est le texte clair crypter par un cryptosystème.
- Le *cryptage, chiffrement* : c'est le processus cryptographique qui permet de transformer un texte clair en cryptogramme.
- Le *déchiffrement* : c'est le processus cryptographique inverse permettant de transformer un cryptogramme en texte clair. C'est une action exécutée par des personnes autorisées.

- Le *décryptage* : c'est le processus cryptographique permettant de transformer un cryptogramme en texte clair. Cette action est faite par des personnes non-autorisées.
- La *clé de chiffrement / déchiffrement* : c'est une information secrète utilisée pendant une opération de cryptage / décryptage pour crypter / décrypter le texte clair / texte chiffré.
- La *cryptanalyse* : c'est une science permettant d'analyser la sécurité d'un cryptosystème. Souvent utilisée par les parties non-autorisées pour casser ou trouver une faille dans un cryptosystème.
- La *cryptologie* : c'est la science qui étudie la cryptographie et la cryptanalyse.

1.2 Eléments de théorie des nombres

La théorie de l'information est due à Shannon [4]. Ses liens avec la cryptographie ont été développés par Hellman selon [2], [3], [4]. Cette notion permet de mesurer la sécurité apportée par un système cryptographique et ainsi de permettre de reconnaître lesquels de ces systèmes sont plus sûrs et adéquats.

1.2.1 Entropie

L'entropie est une notion de la physique, plus précisément dans la thermodynamique. Elle consiste en une mesure de l'état de désordre d'un système d'atomes ou molécules. Elle augmente lorsque le système évolue vers un état de plus grand désordre, et elle diminue si le système évolue vers un état plus ordonné. En théorie de l'information, l'Entropie va nous permettre de mesurer la quantité d'information moyenne contenue dans un ensemble de messages et de mesurer l'incertitude [5].

Définition 1.01 :

Soit E un ensemble partitionné en n sous-ensembles $E_i, 1 \leq i \leq n$ (les messages), c'est-à-dire $E = \bigcup_{i=1}^{n} E_i$. Par définition, la quantité d'information liée à chaque message E_i est :

$$I(E_i) = log_2\left(\frac{|E|}{|E_i|}\right) = log_2\left(\frac{N}{n_i}\right) \qquad (1.01)$$

Et on définit l'entropie de la partition comme :

$$H(partition) = \sum_{i=1}^{n} \frac{n_i}{N} log_2 \left(\frac{N}{n_i}\right) \quad (1.02)$$

Que l'on peut exprimer sous la forme de probabilité avec $p_i = \frac{n_i}{N}$ la probabilité associée à l'apparition du message E_i.

$$H(partition) = -\sum_{i=1}^{n} p_i \, log_2 \, (p_i) = \sum_{i=1}^{n} p_i \, log_2 \, \left(\frac{1}{p_i}\right) \quad (1.03)$$

Du fait que les E_i forment une partition de $E, \sum_{i=1}^{n} p_i = 1$ et l'entropie correspondant à la distribution de probabilité de tous les messages possibles.

Définition 1.02 :

Soit une variable aléatoire X, on définit l'entropie par [5], [6]:

$$H(X) = -\sum_{x} P[X = x] log_2 \, \mathrm{P}[X = x] \quad (1.04)$$

L'entropie $H(X, Y)$ de deux nombres aléatoires est définie comme l'entropie $Z = (X, Y)$,

$$H(X, Y) = -\sum_{x,y} P[X = x, Y = y] log_2 \, P[X = x, Y = y] \quad (1.05)$$

Remarque :

$X \; et \; Y$ sont indépendantes si et seulement si la probabilité pout tous $x \; et \; y$

$$P[X = x, Y = y] = P[X = x] \times P[Y = y]$$

Définition 1.03 :

L'entropie conditionnelle notée $H(X \setminus Y)$ (entropie de X sachant Y) est définie par :

$$H(X \setminus Y) = H(X, Y) - H(Y)$$

C'est-à-dire :

$$H(X \setminus Y) = -\sum_{x,y} P[X = x, Y = y] log_2 P[X = x \backslash Y = y] \quad (1.06)$$

Théorème 1.01 : pour toute distribution, nous avons :

- $H(X,Y) \geq H(X)$ avec égalité si et seulement si Y peut être écrite $f(X)$
- $H(X,Y) \leq H(X) + H(Y)$ avec égalité si et seulement si X et Y sont indépendantes
- Si $P[X = x] \neq 0$ pour au moins n valeurs de x alors $H(X) \leq log_2 n$ avec égalité, si et seulement si, pour tout $X \neq 0$, $P[X = x] = \frac{1}{n}$.

1.2.2 Secret parfait

Un secret parfait veut dire que la distribution du texte clair X après connaissance du texte chiffré est égale à la distribution du texte clair : la condition de distribution, pour $X\ et\ Y$ données, est égale à la distribution originale [4], [5]. Donc, pour tout $x\ et\ y$, telle que $P[Y = y] \neq 0$, on a :

$$P[X = x \backslash Y = y] = P[X = x] \quad (1.07)$$

Théorème 1.02 :

Un secret parfait est équivalent à :

$$H(X \backslash Y) = H(X) \quad (1.08)$$

et l'indépendance statistique entre $X\ et\ Y$.

Démonstration :

♣ Nous avons : $H(X \backslash Y) = H(X,Y) - H(Y) \leq H(X)$, avec égalité si et seulement si $X\ et\ Y$ sont indépendantes. Ainsi, $H(X \backslash Y) = H(X)$ est équivalente à l'indépendance de $X\ et\ Y$.

Si nous avons un secret parfait, alors :

$$\frac{P[X = x, Y = y]}{P[Y = y]} = P[X = x \backslash Y = y] = P[X = x] \quad (1.09)$$

Pour tous $x\ et\ y$, alors $X\ et\ Y$ sont indépendantes, alors $H(X\backslash Y) = H(X)$

Maintenant, si $H(X\backslash Y) = H(X)$, et $X\ et\ Y$ sont indépendantes, nous avons :

$$P[X = x\backslash Y = y] = \frac{P[X = x, Y = y]}{P[Y = y]} = P[X = x]$$

pour tous $x\ et\ y$, alors nous avons un secret parfait. ♦

Théorème 1.03 (Shannon) : Secret parfait implique $H(K) \geq H(X)$

Démonstration :

♣ Nous démontrons d'abord la propriété intermédiaire par laquelle $H(Y) \geq H(X)$

- Premièrement, nous avons $H(Y) \geq H(Y\backslash K)$. Nous remarquons que la connaissance de K donne la même distribution pour $X\ et\ Y$, ainsi $H(Y\backslash K) = H(X\backslash K)$. Mais quand $X\ et\ K$ sont indépendantes, nous obtenons $H(Y\backslash K) = H(X)$. Ainsi nous avons $H(Y) \geq H(X)$.

- Maintenant, nous fixons X, la connaissance de K détermine Y. De plus, $K\ et\ Y$, sont indépendantes. Ainsi nous avons, $H(Y, K\backslash X) = H(K)$. Si nous avons un secret parfait, nous avons $H(Y\backslash X) \geq H(X\backslash Y) + H(Y) - H(X)$. Ainsi, nous avons $H(K) \geq H(X)$.

- D'où nous obtenons $H(K) \geq H(X)$ ♦

Corollaire 1.01 :

Si X est une suite de m-bits et si nous voulant atteindre un secret parfait pour toute distribution de X, alors la clé doit au moins être représentée avec m-bits.

Démonstration :

♣ Si nous voulons atteindre un secret parfait pour toute distribution de X, nous devons avoir $H(K) \geq H(X)$ pour toute distribution de X de la suite de m-bits. Pour une distribution uniforme nous avons $H(K) \geq m$. Maintenant si k est la longueur de la clé, nous savons que pour toute distribution de K, nous avons $H(K) \leq k$, ainsi nous obtenons $k \geq m$. ♦

Théorème 1.04 : Le chiffrement de Vernam fournit un secret parfait pour toute distribution du texte clair.

Démonstration :

♣ Soit $Y = X \oplus K$ le texte chiffré avec $X\ et\ K$ sont des suites binaires indépendantes de longueur n, et K est uniformément distribuée. Pour tout $x\ et\ y$, nous avons :

$$\begin{aligned} P[X = x, Y = y] &= P[X = x, K = x \oplus y] \\ &= P[X = x] \times P[K = x \oplus y] \\ &= P[X = x] \times 2^{-n} \end{aligned}$$

En additionnant en tout x, nous obtenons $P[Y = y] = 2^{-n}$.

Nous déduisons que $P[X = x \backslash Y = y] = P[X = x]$ pour tous $x\ et\ y$. ♦

1.3 Eléments mathématiques pour la cryptographie

Le but de cette paragraphe est de donner les notions de base d'arithmétique [10], [11], [12] [13], [14], [15], [16] [17], [18], [19], qui sont nécessaire en cryptographie.

1.3.1 Les nombres premiers.

On ne considère que l'ensemble $\mathbb{N}$ des entiers naturels

Définitions 1.04 :

Un nombre premier p est un nombre différent de 1 qui n'admet pas d'autres diviseurs que 1 et lui-même.

Un nombre qui n'est pas premier est appelé nombre composé.

Théorème 1.05 :

Tout nombre admet au moins un facteur premier.

Théorème 1.06 :

L'ensemble des nombres premiers est infini.

Théorème 1.07 :

Tout entier peut se décomposer en produit de facteurs premiers.

Selon l'exigence de la norme IEEE-1363 [2], [17], nous montrent que le problème de la factorisation en nombres premiers est un problème difficile qui, nous le verrons, a permis de mettre en place des systèmes cryptographiques presque inviolable. Le deuxième théorème s'interprète sur le plan cryptographique comme l'existence de grands nombres premiers. Ainsi en 1963, le plus grand nombre premier était $2^{8624} - 1$, soit un nombre de 30000 chiffres. Le record actuel est de l'ordre de millions de chiffres.

1.3.2 Congruence dans $\mathbb{Z}$

Définition 1.05 :

Les entiers relatifs *a* et *b* sont congru modulo n s'ils ont même reste par la division par n. Il revient au même de dire que *b*-*a* est multiple de n. On note :

$$a \equiv b[n] \; ou \; a \equiv b(mod\ n) \tag{1.10}$$

Proposition 1.01

(a) Pour tout $a \in \mathbb{Z}, a \equiv a(mod\ n)$: réflexivité

(b) Pour tous $a, b \in \mathbb{Z}, a \equiv b(mod\ n)$, alors $b \equiv a(mod\ n)$: symétrique

(c) Pour tous $a, b, c \in \mathbb{Z}$, si $a \equiv b\ (mod\ n)$, and $b \equiv c\ (mod\ n)$, alors $a \equiv c\ (mod\ n)$: transitivité

Démonstration :

♣ (a) Si $n \in \mathbb{N}$, alors $n|0 = a - a$, soit $a \equiv a(mod\ n)$

(b) Soient $n \in \mathbb{N}, a, b, c \in \mathbb{Z}, a \equiv b(mod\ n)$, si $a - b = kn$ pour $k \in \mathbb{Z}$. Par réécriture, $b - a = (-k)n$, implique $b \equiv a(mod\ n)$

(c) D'après la formule (1.10), quand $a \equiv b(mod\ n)$ et $b \equiv c(mod\ n)$, alors $n|(a - b)$ et $n|(b - c)$. Par conséquent, $n|(a - b) + (b - c) = (a - c)$, c'est-à-dire $a \equiv c\ (mod\ n)$

♦

Proposition 1.02 :

Soient $n \in \mathbb{N}$ et $a, b, c \in \mathbb{Z}$. Si $a \equiv b(mod\ n)$ et $c \equiv d(mod\ n)$, alors :

$$a + c \equiv b + d(mod\ n),\ a - c \equiv b - d(mod\ n), \text{ et } ac \equiv bd(mod\ n) \tag{1.11}$$

Démonstration :

♣ S'il existe $k, l \in \mathbb{Z}$ ainsi que $a = b + kn$ et $c = d + ln$, alors :

$$a \pm c = b + kn \pm (d + ln) = b \pm d + (k \pm l)n,$$

Soit

$$a \pm c \equiv b \pm d(mod\ n)$$

Similairement,

$$ac \equiv (b + kn)(d + ln) \equiv bd(mod\ n)$$ ♦

Proposition 1.03 :

Si $pgcd(c, n) = g$, alors :

$$ac \equiv bc(mod\ n) \tag{1.12}$$

Si et seulement si :

$$a \equiv b(mod\ n/g)$$

Démonstration :

♣ Si $ac - bc = kn$, quelques soit $k \in \mathbb{Z}$, alors $(a - b)c/g = kn/g$. Or $pgcd(c/g, n/g) = 1$. Par conséquent $(n/g)\ divise\ (a - b)$, à savoir :

$$a \equiv b(mod\ n/g)$$

Inversement, si $a \equiv b(mod\ n/g)$, alors il existe un entier $d \in \mathbb{Z}$ tels que $a = b + dn/g$, soit $ac = bc + d(c/g)n$. D'où, $ac \equiv bc(mod\ n)$ ♦

Proposition 1.04 :

Soient $a, b, c \in \mathbb{Z},\ m, n \in \mathbb{N}$ et $a \equiv b(mod\ n)$, nous avons les relations suivantes :

(a) $am \equiv bm(mod\ mn)$

(b) $a^m \equiv b^m(mod\ n)$ (1.13)

(c) Si $m\ divise\ n$, alors $a \equiv b(mod\ m)$

Démonstration :

♣ (a) Pour $a \equiv b(mod\ n),\ a - b = kn$, quelque soit k. En multipliant par m, nous obtenons $(a - b) = knm$, d'où $am - bm = (km)n$, tels que $am \equiv bm(mod\ n)$

(b) Si $n|(a - b)$, alors

$$n|(a - b)(a^{m-1} + a^{m-2}b + \cdots + b^m) = a^m - b^m$$

En d'autre terme,

$$a^m \equiv b^m(mod\ n)$$

(c) Si $a = b + kn$ quelque soit $k \in \mathbb{Z}$ et $n = lm$ quelque soit $l \in \mathbb{N}$, alors $a = b + klm$, si $a - b = (kl)m$, d'où $a \equiv b(mod\ m)$ ♦

1.3.3 Ensemble quotient $\mathbb{Z}|n\mathbb{Z} = \mathbb{Z}_n$

Proposition 1.05 :

La relation de congruence est une relation d'équivalence sur $\mathbb{Z}$.

Définitions 1.06 :

L'ensemble $\bar{x}$ des éléments congrus à x de $\mathbb{Z}$ est dit classe d'équivalence de $x\ modulo\ n$. L'ensemble des classes d'équivalence modulo n est noté $\mathbb{Z}|n\mathbb{Z} = \mathbb{Z}_n$

Définition 1.07 :

On définit deux lois internes sur $\mathbb{Z}_n$ par les deux égalités suivantes :

$$\overline{xy} = \overline{yx}\ \ et\ \ \bar{x} + \bar{y} = \overline{x + y} \quad (1.14)$$

Théorème 1.08 :

L'ensemble quotient $\mathbb{Z}_n$ muni des lois précédentes est un anneau.

1.3.3.1 Divisibilités dans $\mathbb{Z}$

Définition 1.08 :

Soient a et b deux éléments de $\mathbb{N}$ avec $a \neq 0$, « a divise b » que l'on note $a|b$ s'il existe un $k \in \mathbb{N}$ telle que : $b = ka$. On dit aussi que $b\ est\ multiple\ de\ a$.

Théorème 1.09 :

Pour $a\ et\ b$ éléments de $\mathbb{Z}$, il existe un couple unique (q, r) de $\mathbb{Z}^2$ tel que :

$$a = bq + r\ avec\ 0 \leq r < |b| \tag{1.15}$$

On dit que q est le quotient et r le reste de la division euclidienne de a par b.

a *:* la dividende.

b : le diviseur.

Démonstration :

♣ Considérons l'ensemble $S = \{a - bx\ /x \in \mathbb{Z}\ et\ a - bx \geq 0\}$.

$S \neq \emptyset$, car $a - b(-sign\ b)a^2) \geq 0$. Il existe donc un plus petit élément dans S, désignons le par $r = a - bq$. Par hypothèse $r \geq 0$. En supposant que $r \geq |b|$, on aurait obtenu un élément

$$0 \leq r - |b| = r - b\ sign\ b = a - b(q + sign\ b) \in S$$

Or $0 \leq r - |b| = a - b(q + sign\ b) < r$, d'où la contradiction et ceci n'est éliminée que si $r < |b|$ ♦

1.3.3.2 Plus petit commun multiple ($ppcm$) et plus grand commun diviseur ($pgcd$).

Définition 1.09 :

Soient a et b deux éléments de $\mathbb{Z}$.

- On dit que m est le $ppcm$ de a et b et noté $m = ppcm(a, b)$ si et seulement si :

- $a\ divise\ m$ et $b\ divise\ m$: $a|m$ et $b|m$

- Si a divise m' et b divise m' alors m divise m': $a|m'$ et $b|m' \implies m|m'$

- On dit que d est le plus grand commun diviseur de a et b et noté $d = pgcd(a, b)$ si

- $d\ divise\ a\ et\ d\ divise\ b$: $d|a\ et\ d|b$

- Si $d'\ divise\ a\ et\ d'\ divise\ b\ alors\ d'\ divise\ d$: $d'|a\ et\ d'|b \Rightarrow d'|d$

Remarque : On trouve pour ces relations d'ordre :

$$pgcd(a, b) = sup\{a, b\}\ et\ ppcm(a, b) = inf\{a, b\}.$$

Définition 1.10 :

Les éléments a et b de $\mathbb{Z}$ sont premiers entre eux, si et seulement si, ils n'ont pour diviseur commun que 1 et -1.

Théorème 1.10 :

Les propriétés suivantes sont équivalentes :

(i) a et b sont premiers entre eux.

(ii) $pgcd(a, b) = 1$.

(iii) $\exists\ u, v \in \mathbb{Z}$ tels que $au + bv = 1$

(iv) $\forall z \in \mathbb{Z}, \exists\ x, y \in \mathbb{Z}\ \ ax + by = z$

Théorème 1.11 :

$a\ et\ b$ sont premiers entre eux, si et seulement si, $\forall x \in \mathbb{Z}\ a|bx \Rightarrow a|x$

Corollaire 1.02 :

Si p est premier et divise un produit de facteur, alors il divise un des facteurs. Si p est premier et divise a^n, alors il divise a.

Théorème 1.12 :

Tout entier relatif est produit de facteurs premiers et sa décomposition en premiers est unique à l'ordre prés des facteurs. On dit que $\mathbb{Z}$ est factoriel.

Corollaire 1.03 :

D'après ce qui précède, on a :

$$|ab| = pgcd(a,b)ppcm(a,b) \tag{1.16}$$

1.3.4 Algorithme d'Euclide.

Cet algorithme a été établi et baptisé ainsi par Etienne Bézout [2], il permet de calculer le pgcd de deux entiers a et b en effectuant un nombre fini de divisions euclidiennes. Soient $a\ et\ b$ deux entiers positifs, on pose $r_0 = a$ et $r_1 = b$, et tant que $r_1 > 0$ on effectue les divisions euclidiennes successives suivantes, suivant le théorème 1.09.

$$\begin{cases} r_0 = r_1 q_1 + r_2 & 0 \le r_2 \le r_1 \\ r_1 = r_2 q_2 + r_3 & 0 \le r_3 \le r_2 \\ \qquad\qquad ... & \\ r_{k-2} = r_{k-1} q_{k-1} + r_k & 0 \le r_k \le r_{k-1} \\ r_{k-1} = r_k q_k + r_{k+1} & 0 \le r_{k+1} w \le r_k \end{cases} \tag{1.17}$$

Pour chaque $k \ge 0$, on a $pgcd(a,b) = pgcd(r_k, r_{k+1})$. La suite des restes $(r_1, r_2, r_3, \dots)$ étant une suite strictement décroissante d'entiers positifs, on obtient nécessairement un reste nul au bout d'un nombre fini de divisions.

Soit r_n le dernier reste non nul. On a $r_{n+1} = 0$, ce qui signifie que :

$$pgcd(a,b) = pgcd(r_n, r_{n+1}) = pgcd(r_n, 0) = r_n \tag{1.18}$$

D'où l'algorithme d'Euclide : On effectue les divisions euclidiennes successives décrites ci-dessus jusqu'à obtenir un reste nul, le pgcd de *a* et *b* est le dernier reste non nul.

Démonstration :

♣ La séquence $\{r_i\}$, obtenue à partir d'une division successive, est une suite strictement décroissante, qui s'arrête sur des valeurs positives n, telles que $r_{n+1} = 0$.

En utilisant la formule (1.15) et la relation $pgcd(a,b) = pgcd(a,r)$, nous avons :

$$pgcd(a,b) = pgcd(r_i, r_{i+1})$$

Pour tout $i \geq 0$, nous avons, en particulier :

$$pgcd(a,b) \; = \; pgcd(r_n, r_{n+1}) \; = \; pgcd(r_n, 0) \; = \; r_n \qquad \blacklozenge$$

1.3.5 Algorithme d'Euclide étendu.

L'algorithme d'Euclide étendu permet de déterminer $d = pgcd(a,b)$ ainsi que deux entiers u et v vérifiant.

$$d \; = \; au \; + \; bv \tag{1.19}$$

Reprenons la suite (1.19) des divisions euclidiennes et à chaque étape $k \geq 0$, calculons les entiers u_k et v_k tels que

$$r_k = au_k + bv_k \tag{1.20}$$

En particulier pour $k = 0$ on a $u_0 \; = \; 1, v_0 = 0, u_1 = \; 0$ et $v_1 \; = \; 1$.

On en déduit les relations de récurrence que pour $k \geq 0$, on a

$$\begin{cases} u_{k+1} = u_{k-1} - u_k q_k \\ v_{k+1} = v_{k-1} - v_k q_k \end{cases} \tag{1.21}$$

D'où l'algorithme d'Euclide étendu énoncé ci-dessous :

Soient deux entiers positifs a et b. Pour déterminer $d = pgcd(a,b)$, ainsi que deux entiers u et v tels que $d = au + bv$

On écrit $\begin{cases} r_0 = a \\ u_0 = 1 \\ v_0 = 0 \end{cases} \quad \begin{cases} r_1 = b \\ u_1 = 0 \\ v_1 = 1 \end{cases}$ et $\forall k \geq 1, \quad \begin{cases} r_{k+1} = r_{k-1} - r_k q_k \\ u_{k+1} = u_{k-1} - u_k q_k \\ v_{k+1} = v_{k-1} - v_k q_k \end{cases}$

Notons que l'algorithme d'Euclide étendu est similaire à l'algorithme d'Euclide, mais avec deus suites $u_k \; et \; v_k$ supplémentaires. Il existe $k \geq 0$ tel que $r_{k+1} = 0$

1.3.6 Exponentiation modulo n.

Le but est de calculer l'expression : $a^e \ mod\ n$ avec $a = 2,3,\ldots$. Quand on le calcul on constate que les résultats apparaissent de façon très aléatoirement dans l'ensemble des entiers modulo *n.*

Pour calculer de grandes puissances modulo n, on procède comme suit. On observe d'abord que les règles de calculs pour les exposants sont aussi valables modulos *n.* En particulier, on a les identités suivantes :

$$a^{2k} \equiv (a^2)^k (mod\ n)\ et\ a^{2k+1} \equiv (a^2)^k\ a\ (mod\ n) \tag{1.22}$$

et on peut choisir de remplacer a^2 par son reste $modulo\ n$, chaque fois que cela permet de simplifier le calcul.

1.3.7 Théorème de Fermat et d'Euler.

On désigne par $U(\mathbb{Z}_n)$ le groupe multiplicatif des éléments inversibles de $\mathbb{Z}_n$, l'ordre de ce groupe est égal à $\varphi(n)$, nombre des éléments inférieurs à n premiers avec n, ($\varphi(n)$ est appelée l'indicateur d'Euler). Ainsi on obtient :

Théorème 1.13 : *Théorème d'Euler*

Pour tout $\bar{a} \in U(\mathbb{Z}_n)$

$$(\bar{a})^{\varphi(n)} = \bar{1} \tag{1.23}$$

D'où quel que soit $b \in \mathbb{Z}$, premier avec n, c'est-à-dire $pgcd(b,n) = 1$, $b^{\varphi(n)} \equiv 1(mod\ n)$

Démonstration :

♣ Comme l'ordre du groupe $U(\mathbb{Z}_n)$ est égale à $\varphi(n)$, nous avons $(\bar{a})^{\varphi(n)} = \bar{1}$ pour tout $\bar{a} \in U(\mathbb{Z}_n)$, de plus si $pgcd(b,n) = 1$, $\bar{b} \in U(\mathbb{Z}_n)$ et $(\bar{b})^{\varphi(n)} = \bar{1}$, équivaut à $(\bar{b})^{\varphi(n)} \equiv 1(mod\ n)$ ♦

Théorème 1.14 : Petit théorème de Fermat

Si p est un nombre premier, alors quelque soit le nombre entier a non divisible par p

$$a^{p-1} \equiv 1(mod\ p) \Leftrightarrow a^{p-1} - 1 \equiv 0(mod\ p)$$

Et quelque soit le nombre entier $a \in \mathbb{Z}$

$$a^p \equiv a(mod\ p) \tag{1.24}$$

Démonstration :

♣ Il suffit d'appliquer le théorème 1.13 précédent sachant que $\varphi(p) = p - 1$ ♦

1.3.8 Système de congruence : Théorème des restes chinois.

Soient $n_1, n_2 \geq 2$ deux entiers premiers entre eux et $u_1 n_1 + u_2 n_2 = 1$ une équation de Bézout. Soient $a_1, a_2 \in \mathbb{Z}$ et $a \in \mathbb{Z}$ tel que $a \equiv a_1 u_2 n_2 + a_2 u_1 n_1 (mod\ n_1 n_2)$. Alors pour $x \in \mathbb{Z}$, on a l'équivalence :

$$\left.\begin{matrix} x \equiv a_1(mod\ n_1) \\ x \equiv a_2(mod\ n_2) \end{matrix}\right\} \Leftrightarrow x \equiv a(mod\ n_1 n_2) \tag{1.25}$$

Ou encore pour tout $k \in \mathbb{Z}$, $x = a + k n_1 n_2$

Démonstration :

♣ Vérifions d'abord que a est bien une solution du système de congruences. En effet, d'après l'hypothèse, nous avons, pour $k \in \mathbb{Z}$:

$$a \equiv a_1 u_2 n_2 + a_2 u_1 n_1 + k n_1 n_2$$

Donc

$$a \equiv a_1 u_2 n_2 \equiv a_1(u_2 n_2 + u_1 n_1) \equiv a_1(mod\ n_1)$$

Et de même

$$a \equiv a_2 u_1 n_1 \equiv a_2(u_1 n_1 + u_2 n_2) \equiv a_2(mod\ n_2)$$

Montrons maintenant l'équivalence. Supposant que $x \equiv a(mod\ n_1 n_2)$. Alors $x = a + k n_1 n_2$ pour un $k \in \mathbb{Z}$ et donc $x \equiv a \equiv a_1(mod\ n_1)$ et $x \equiv a \equiv a_2(mod\ n_2)$. Alors on a :

$x \equiv a(mod\ n_1)$ et $x \equiv a(mod\ n_2)$. Donc $x - a$ est divisible par n_1 et par n_2, comme n_1 et n_2 sont premiers entre eux, il s'ensuit que $x - a$ est divisible par $n_1 n_2$, c'est-à-dire que $x \equiv a(mod\ n_1$ et $n_2)$ ♦

1.3.9 Décomposition d'un entier en facteurs premiers.

Proposition 1.06 :

Soit *p* un nombre premier.

Si p divise un produit $q_1 q_2 \dots q_n$ de n entiers, il existe au mois un indice $i \in \{1,2,\dots,n\}$ tel que $p\ divise\ q_i$

Corollaire 1.04 :

Soit p un nombre premier. Si p divise un produit $p_1 p_2 \dots p_n$ de n nombres premiers, il existe un indice $i \in \{1,2,\dots,n\}$ tel que $p = p_i$.

Théorème 1.15: (*Théorème fondamental de l'arithmétique) :*

Tout entier $a > 1$ s'écrit de façon unique :

$$a = p_1^{\alpha 1} p_2^{\alpha 2} \dots p_n^{\alpha n} \tag{1.26}$$

Où $\begin{cases} \text{les entiers } p_i \text{ sont premiers et vérifient } p_1 < p_2 < \dots < p_n \\ les\ entiers\ \alpha_i\ sont\ positifs \end{cases}$

L'égalité (1.26) constitue la décomposition de l'entier a en facteurs premiers.

1.3.10 Résidu quadratique

Soit $n \geq 2$ un entier, a un entier inférieur à n. On dit que a est un résidu quadratique modulo n si $a = b^2$ pour une classe b. C'est-à-dire :

$$b^2 \equiv a\ (mod\ n) \tag{1.27}$$

1.3.11 Symbole de Legendre

Définition 1.10 : Si $c \in \mathbb{Z}\ et\ p > 2$ premier, alors :

$$\left(\frac{c}{p}\right) = \begin{cases} 0 & si\ p\ divise\ c \\ 1 & si\ c\ est\ résidu\ quadratique\ modulo\ p \\ -1 & si\ c\ est\ non\ résidu\ quadratique\ modulo\ p \end{cases} \tag{1.28}$$

Et $\left(\frac{c}{p}\right)$ est appelé symbole de Legendre de c avec respect de p.

Propriété 1.01 :

Si $p > 2$ est premier et $b, c \in \mathbb{Z}$ avec $p \nmid bc$, alors :

(1) $\left(\frac{c}{p}\right) \equiv c^{\frac{p-1}{2}} (mod\ p)$

(2) $\left(\frac{b}{p}\right)\left(\frac{c}{p}\right) = \left(\frac{bc}{p}\right)$

(3) $\left(\frac{b}{p}\right) = \left(\frac{c}{p}\right)$, à $condition\ que\ b \equiv c (mod\ p)$

Démonstration :

♣ (1) est le corollaire du critère d'Euler. Ainsi on peut l'utiliser pour déterminer (2)

(2) $\left(\frac{b}{p}\right)\left(\frac{c}{p}\right) \equiv b^{(p-1)/2} c^{(p-1)/2} \equiv (bc)^{(p-1)/2} \equiv \left(\frac{bc}{p}\right) (mod\ p)$

(3) est la conséquence immédiate de la définition du résidu quadratique ♦

1.3.12 Logarithme discret

Le calcul du nombre x qui fait en sort que $y = b^x$, pour y et b donnés, correspond à trouver le logarithme, $\log_b y$. Dans le cas des entiers modulos un nombre premier, on peut considérer la même problème ; parce que les puissances successives b^x parcourent tous les entiers $(\neq 0) modulo\ p$, lorsque x parcourt tous les entiers modulo $p - 1$.

On obtient ainsi la notion de logarithme discret. Ici, le mot discret sert à distinguer de la notion usuelle qu'on qualifie souvent de continue. Pour p un nombre premier, et b entre 2 et *p - 2*, on peut reformuler le théorème comme

$$y \equiv b^x (mod\ p) \tag{1.29}$$

Exactement lorsque

$$x \equiv \log_b(x)\ (mod\ p - 1) \tag{1.30}$$

1.4 Principaux systèmes de chiffrement

1.4.1 Historique

L'histoire de la cryptographie est déjà longue [20], [21]. On rapport son utilisation en Egypte il y a 4000 ans. Toutefois, pendant des siècles, les méthodes utilisées étaient restées souvent très primitives. D'autre part, sa mise en œuvre était limitée aux besoins de l'armée et de la diplomatie. Les méthode de chiffrement et de cryptanalyse ont connu un développement très important au cours de la seconde guerre mondiale et ont eu une profonde influence sur le cours de celle-ci.

Mais d'après [22], [23], c'est la prolifération actuelle des systèmes de communication qui a fait sortir la cryptographie du domaine militaire. De plus, elle a diversifié la demande et provoqué le développement de nouvelles techniques cryptographiques. Elle est à l'origine d'un développement rapide depuis les dernières décennies, qui ne semble pas s'essouffler aujourd'hui, bien au contraire.

Nombreux systèmes de chiffrement [3], [20], [21], [24] différents ont été imaginés pour se protéger contre la curiosité et la malveillance des ennemis depuis des siècles. On peut classer ces systèmes en trois grandes classes que nous allons représenter sur la *Figure 1.01*.

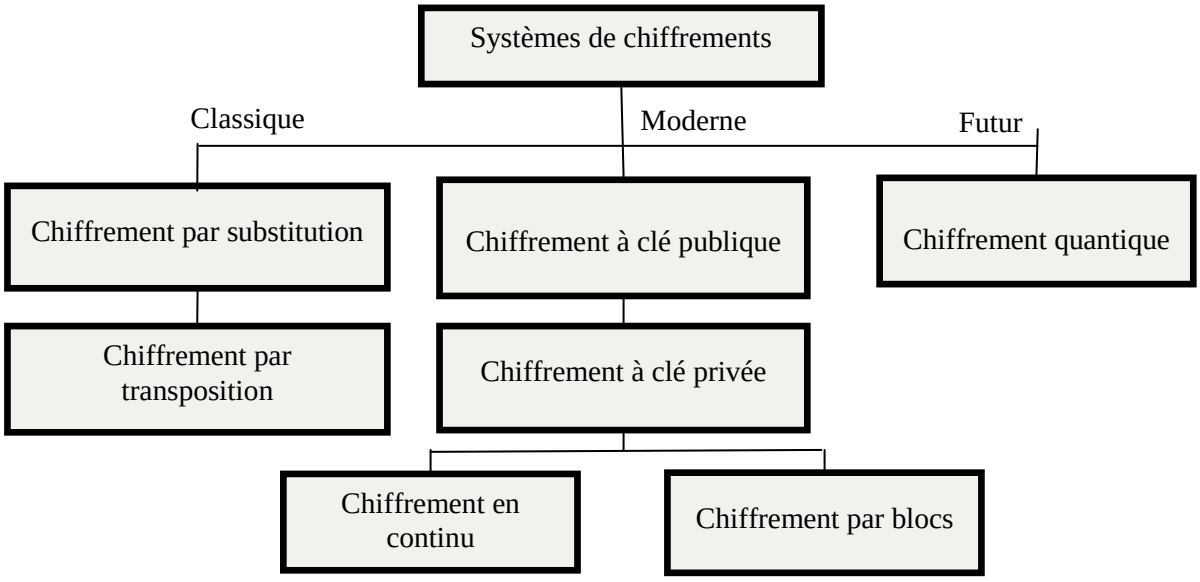


Figure 1.01: *Les principales techniques de chiffrement*

1.4.2 Systèmes classiques

Avant l'avènement des ordinateurs, l'opération de chiffrement était basée sur des caractères. L'idée était de transposer ou de remplacer les caractères d'un texte par d'autres [3], [23], [27]. Les meilleurs systèmes répètent ces deux opérations de base plusieurs fois.

1.4.2.1 Substitution

Historiquement, c'est le premier type de chiffrement utilisé. C'est un chiffrement dans lequel chaque caractère du texte en clair est remplacé par un autre caractère dans le texte chiffré.

Définitions 1.11 :

Pour tous $x, y \in \mathbb{Z}_{26}, s \in \mathbb{N}$, nous avons les schémas de chiffrement / déchiffrement suivant :

$$\left.\begin{matrix} x \longmapsto x + s \ (mod\ 26) \\ y \longmapsto y - s (mod\ 26) \end{matrix}\right\} \ chiffre\ additif$$

Pour $pgcd(s, 26) = 1$, on a $s = \{1,3,5,7,9,11,15,17,19,21,23,25\}$:

$$x \longmapsto xs\ (mod\ 26) \quad chiffre\ multiplicatif$$

Pour déchiffrer, on utilise le théorème de Bézout :

$$pgcd(s, 26) = 1 \Leftrightarrow \exists\ x, y \in \mathbb{N}: sx + 26y = 1 \Leftrightarrow x \equiv t^{-1} (mod\ 26)$$

1.4.2.2 Transposition

Un chiffrement par transposition est un chiffrement dans lequel les caractères du texte en clair demeurent inchangés mais dont les positions respectives sont modifiées.

Définition 1.12 :

Soit $\mathcal{A}_C\ et\ \mathcal{A}_M$ l'alphabet des cryptogrammes, respectivement l'alphabet des textes clairs. Soit l'application f qui, pour une clé k, modifie la position i du texte clair M en la position $k(i)$ dans le cryptogramme C.

$$c_i = f(m_i) = m_{k(i)} \quad \forall i, 0 < i < |m|$$

La substitution et la transposition peuvent être facilement cassées car elles ne cachent pas les fréquences des différents caractères du texte en clair. D'ailleurs, les procédures de chiffrement et de déchiffrement doivent être gardées secrètes.

1.4.3 Systèmes modernes

Les systèmes modernes sont plus complexes [3], [28], [29], cependant la philosophie reste la même. La différence fondamentale est qu'ils exploitent la puissance des ordinateurs modernes en manipulant directement des bits, par opposition aux anciennes méthodes qui s'opèrent sur des caractères alphabétiques. Ce n'est donc qu'un changement de taille ou de représentation.

On distingue deux classes de chiffrement à base de clés [3], [20], [22], [28], [29]:

1.4.3.1 Chiffrement à clé privé

Le chiffrement à clé privé consiste à utiliser la même clé pour le chiffrement et le déchiffrement.

Par analogie c'est le principe d'une serrure d'une porte : tous les utilisateurs autorisés ont une clé identique. On distingue le système de chiffrement en continu et chiffrement par bloc.

1.4.3.2 Chiffrement à clé publique

Les problèmes de distribution des clés sont résolus par la cryptographie à clé publique ou cryptographie asymétrique. Ce concept a été introduit par W. Diffie et M. Hellman [2] en 1975. La cryptographie à clé publique est un procédé asymétrique utilisant une paire de clés asymétrique associé : une clé publique qui crypte des données et une clé privée ou secrète correspondante pour le déchiffrement.

Ainsi, on publie la clé publique tout en conservant la clé privée secrète. Il est basé sur une méthode mathématique garantissant un cryptage facile et rapide, et un décryptage difficile. Par analogie, considérons que l'on crypte le message avec un cadenas (clé publique) que le détenteur de la clé privée peut ouvrir pour lire le cryptogramme. Il est impossible de retrouver la clé privée à partir de la clé publique.

1.4.4 Systèmes de chiffrement quantique

Les systèmes de chiffrement quantique sont des systèmes fondés sur la mécanique quantique et les propriétés particulières de la matière dans ce domaine. Ils reposent sur le principe d'incertitude d'Heisenberg [30], selon lequel la mesure d'un système quantique perturbe ce système. Une oreille indiscrète sur un canal de transmission quantique engendre des perturbations inévitables qui alertent les utilisateurs légitimes. Ce système résout ainsi les problèmes de distribution de clé.

1.4.5 Définitions des Services offertes

Les principaux services offerts par la cryptographie moderne sont les suivantes [2], [3] et [12]:

- La *confidentialité* : assure que les données concernées ne pourront être dévoilées qu'aux personnes autorisées.

- *L'intégrité* : assure que les données ne seront pas altérées (intentionnellement ou non) pendant leur transmission ou leur stockage.

- *L'authentification / l'identification* : Prouver l'origine d'une donnée ou l'identité d'une personne.

- La *signature* proprement dite (undeniability ou non répudiation) : permet à une personne de prendre part à un contrat avec impossibilité de renier ensuite ses engagements.

1.5 Principe général du chiffrement

Le principe de chiffrement est précisé dans [3], [20] et [22].

Définitions 1.13 :

- Le texte clair est noté *M*. Ce peut être une suite de bits, un fichier de texte, un enregistrement de voix numérisée, ou une image vidéo numérique, …

- Le texte en clair peut être transmis ou stocké. Le texte chiffré est noté *C*, qui a la même taille que *M*, parfois plus grand.

- La fonction de chiffrement, notée *E*, transforme M en C.

$$E(M) = C$$

- La fonction inverse, notée *D*, de déchiffrement transforme *C* en *M* :

$$D(C) = M$$

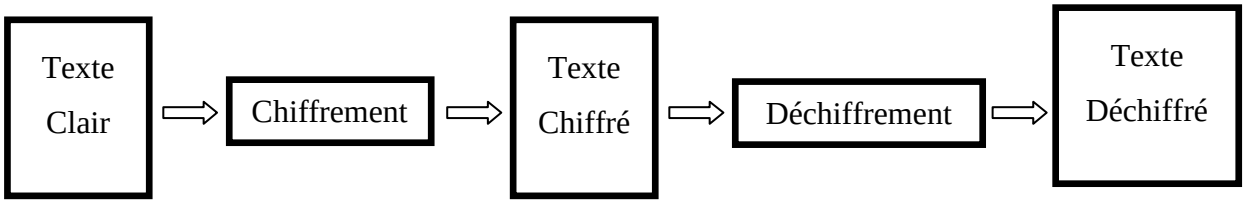


Figure 1.02: *Chiffrement et déchiffrement*

1.6 Clé de chiffrement

Une clé est une valeur qui est utilisée avec un algorithme cryptographique pour produire un texte chiffré spécifique. La cryptographie moderne utilise souvent une clé, notée *K*. Cette clé *K* peut prendre une valeur parmi un grand nombre de valeurs possibles. L'ensemble des valeurs possibles d'une clé est appelé espace des clés [22], [23], [27].

1.6.1 Chiffrement avec une clé

Dans ce type de chiffrement, les opérations de chiffrement et de déchiffrement utilisent toutes les deux la clé *K*, aussi, les fonctions s'écrivent de la même manière suivante :

$$E_k(M) = C \; et \; D_k(C) = M$$

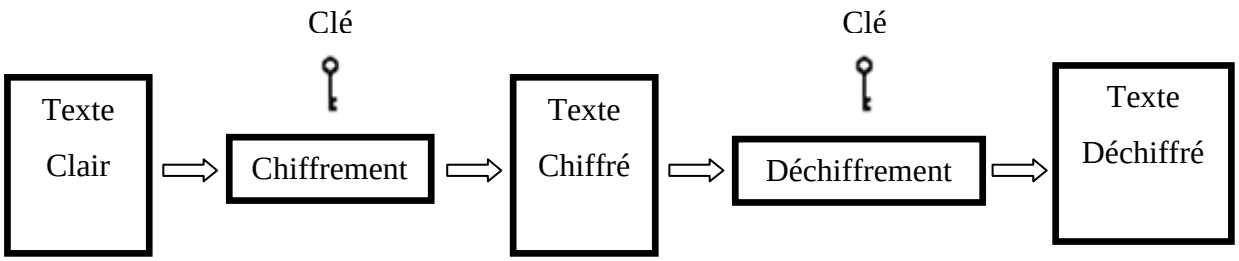


Figure 1.03: *Chiffrement et déchiffrement avec une clé*

1.6.2 Chiffrement avec deux clés

Certains algorithmes utilisent des clés différentes pour le chiffrement et le déchiffrement. Dans ce cas, la clé de chiffrement, notée k_1 est différente de la clé de déchiffrement, notée k_2.

$$E_{k1}(M) = C \; et \; D_{k2}(C) = M$$

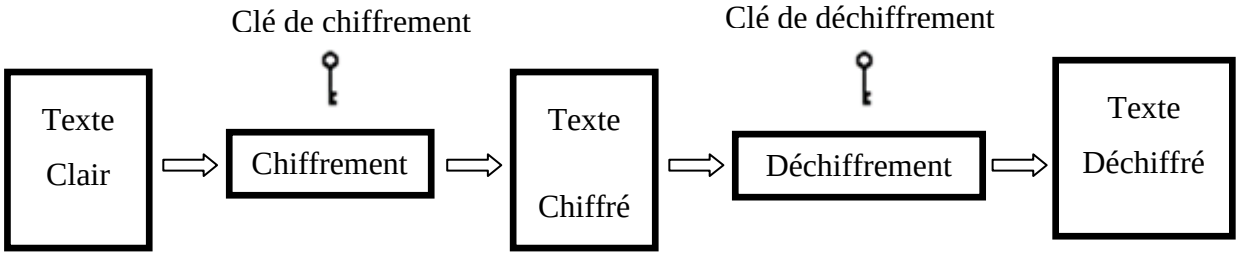


Figure 1.04: *Chiffrement et déchiffrement avec deux clés*

1.7 Algorithme cryptographique

Il y a deux types principaux d'algorithmes à base de clés [4], [20], [22] : algorithme à clé secrète et algorithme à clé publique.

1.7.1 Algorithme à clé secrète

Les algorithmes à clé secrète sont des algorithmes où la clé de chiffrement peut être calculée à partir de la clé de déchiffrement ou vice versa. Dans la plupart des cas, la clé de chiffrement et la clé de déchiffrement sont identiques. Pour de tels algorithmes, l'émetteur et le destinataire doivent se mettre d'accord sur une clé à utiliser avant d'échanger des messages [30]. Cette clé doit être gardée secrète. La sécurité d'un algorithme à clé secrète repose ainsi sur la clé.

Les algorithmes à clé secrète peuvent être classés en deux catégories. Certains opèrent sur le message en clair un bit ou un octet à la fois. Ceux-ci sont appelés algorithmes de chiffrement en continu. D'autres opèrent sur le message en clair par groupes de bits de taille supérieure à un bit. Ces groupes de bits sont appelés blocs, et les algorithmes correspondants sont appelés algorithmes de chiffrement par blocs. La taille typique des blocs est de 64 bits [27], [28].

1.7.1.1 Communications à l'aide d'un cryptosystème à clé secrète

Nous allons voir comment l'émetteur A peut envoyer un message chiffré au destinataire B [25] :

1) A et B choisissent un cryptosystème ;
2) A et B choisissent une clé ;
3) A chiffre son texte en clair à l'aide de l'algorithme choisi et avec la clé sélectionnée. Cette étape produit un texte chiffré ;
4) A envoi le texte chiffré à B ;
5) B déchiffre le texte chiffré avec le même algorithme et la même clé, et finalement lit le message.

Avec un algorithme à clé secrète, A et B peuvent réaliser l'étape 1 en public mais ils doivent réaliser l'étape 2 en secret. La clé doit être restée secrète avant, pendant et après le protocole, sinon, le message ne serait plus confidentiel.

1.7.1.2 Système de chiffrement en continu

Les systèmes de chiffrement en continu sont spécifiquement utiles pour chiffrer des communications continues. Le même bit ou octet du texte clair sera chiffré en un bit ou un octet différent à chaque chiffrement [29], [30].

a. Principe de base

La plupart des conceptions d'algorithmes de chiffrement en continu tournent autour des Registres de Décalage à Rétroaction Linéaire ou RDRL.

b. Le RDRL

Le RDRL est un circuit combinatoire simple composé de deux parties :

- Un registre à décalage : c'est une suite de bits et chaque fois qu'un bit est nécessaire, tous les bits du registre sont décalés vers la droite de un bit.

- La fonction de rétroaction : c'est une simple combinaison par *ou exclusif* de certains bits du registre. La liste de ces bits s'appelle séquence de dérivation.

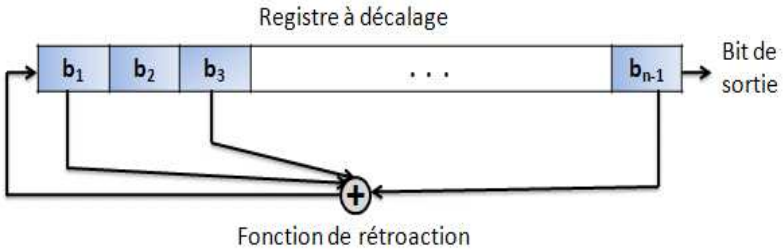


Figure 1.05: *Principe du RDRL*

c. Générateur de codons

La réalisation la plus simple d'un algorithme de chiffrement en continu est sur la *Figure 1.06.*

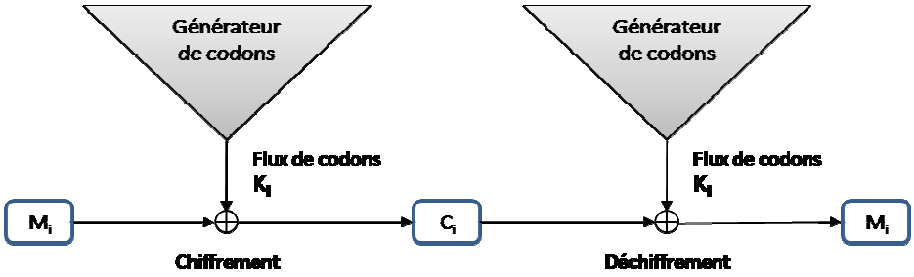


Figure 1.06: *Principe du générateur de codons*

Un générateur de codons engendre un flux de bits $k_1, k_2, \dots, k_i$. Ce flux est combiné par *ou exclusif* avec le flux de bits du texte clair $m_1, m_2, \dots, m_i$ pour produire le flux de bits du texte chiffré $c_1, c_2, \dots, c_i$.

$$c_i = m_i \oplus k_i$$

Du côté de déchiffrement, les bits du texte chiffré sont combinés par *ou exclusif* avec un flux identique de codons pour retrouver les bits du texte en clair :

$$m_i = c_i \oplus k_i$$

d. Générateur périodique

Comme le générateur de codons doit engendrer le même flux au chiffrement et au déchiffrement, il doit être déterministe. Comme il est réalisé par une machine ou matériel à états finis, la suite finira par se répéter. Ces générateurs de codons sont appelés générateurs périodiques.

Un générateur de codons doit avoir une longue période, bien plus longue que le nombre de bits dont le générateur devra produire entre deux changements de clé. Si la période est plus courte que le texte en clair, alors différentes parties du texte en clair seront chiffrées de la même manière. Si un cryptanalyste connaît une partie du texte en clair, il peut reconstituer une partie du flot de clés et l'utiliser pour en savoir plus sur le texte en clair.

Même si le cryptanalyste n'a que le texte chiffré, il peut combiner par *ou exclusif* les parties chiffrées avec le même flot de clé et obtenir les combinaisons par *ou exclusif* du texte en clair avec ce dernier.

$$\text{Message crypté : } c_i = m_i \oplus k_i;$$
$$\text{Cryptanalyste : } c_i \oplus k_i = m_i \oplus k_i.$$

C'est juste un algorithme de combinaison par *ou exclusif* avec une très longue clé.

e. RC4

Conçu par Ron Rivest, RC4 est un algorithme de chiffrement en continu, pouvant utiliser des clés de taille variables jusqu'à 2048 bits ce qui réduit bien évidemment la possibilité d'attaques.

Le flux de codons ne dépend pas du texte en clair. Il a une table-S à 8×8 bits : T[0], …, T[255]. Il y a deux compteurs *x* et *y* initialisés à zéro. Pour générer un octet aléatoire *b*, voici l'algorithme :

- $x \leftarrow x + 1\ modulo\ 256$
- $y \longleftarrow T[x] + y\ modulo\ 256$
- $échanger\ T[x]\ et\ T[y]$
- $b \longleftarrow T[x] + T[y]\ modulo\ 256.$

L'octet *b* est combiné par *ou exclusif* avec le texte en clair pour produire le texte chiffré ou bien avec le texte chiffré pour produire le texte en clair. L'initialisation est facile, on commence avec l'identité : $T[0] = 0, T[1] = 1, \ldots T[256] = 256$. Notons que la clé est devenue une suite de 256 octets en concaténant autant de copies nécessaires.

Voici l'algorithme correspondant :

- $c \longleftarrow 0$
- $Pour\ i\ de\ 0$ à 255
- $c \longleftarrow K[i] + T[i] + c\ modulo\ 256$
- $Echanger\ T[i]\ et\ T[c]$
- $Fin\ pour\ i$

f. Sécurité d'un système de chiffrement

La sécurité d'un système de chiffrement en continu dépend entièrement des détails internes du générateur de codons. Si le générateur de codons engendre un flux continu de zéros, le texte chiffré résultant sera identique au texte en clair et toute l'opération sera inutile. Le générateur de codons engendre un flux de bits qui ont l'air aléatoires, mais en fait, il est déterministe et il peut être reproduit de façon infaillible au moment du déchiffrement. Plus la sortie du générateur est aléatoire, plus il est difficile pour le cryptanalyste de le casser. Si, néanmoins, le générateur engendre le même flot de bits chaque fois qu'on l'active, le système cryptographique résultant sera facile à casser.

Si le cryptanalyste a un texte chiffré et le texte en clair correspondant, il peut combiner par *ou exclusif* le texte en clair avec le texte chiffré pour retrouver le flux de codons. Maintenant, chaque fois qu'il intercepte un message chiffré, il a les codons nécessaires pour déchiffrer le message. De plus, il peut déchiffrer et lire tout texte chiffré qu'il aurait intercepté avant.

Quand le cryptanalyste obtient une seule paire « texte en clair, texte chiffré », il peut tout lire. C'est pourquoi tous les algorithmes de chiffrement en continu utilisent des clés. La sortie du générateur de codons est une fonction de la clé. Maintenant, s'il a une paire « texte en clair, texte chiffré », il peut seulement lire les messages chiffrés avec une seule clé. Lorsqu'on change la clé, l'adversaire se retrouve à la case de départ.

1.7.1.3 Système de chiffrement par bloc

a. Principe de base

Le but des cryptanalystes est de déterminer la clé *K*, le texte en clair *M*, ou les deux. La plupart des cryptanalystes du monde réel ont quelques informations probabilistes concernant *M* avant de commencer. Ils utilisent la redondance naturelle du langage pour réduire le nombre possible de textes en clair.

b. La confusion et la diffusion

Selon Shannon, les deux techniques de base pour effacer les redondances dans un texte en clair sont la confusion et la diffusion. Les algorithmes par blocs utilisent à la fois ces deux techniques.

La confusion sert à cacher les relations entre le texte en clair, le texte chiffré et la clé. Une bonne confusion rend les relations statistiques si compliquées que même les outils de cryptanalyses puissants ne marchent pas. Le moyen le plus simple de réaliser cela est la substitution exposée dans le paragraphe précédent.

La diffusion disperse les redondances du texte en clair en les répartissant dans le texte chiffré. Un cryptanalyste qui cherche ces redondances aura plus de difficulté à les trouver. Le moyen le plus simple de provoquer cela est la transposition.

c. Réseaux de Feistel

La plupart des algorithmes de chiffrement par blocs sont des réseaux de Feistel. Le principe c'est qu'un bloc de longueur *n* bits est divisé en deux moitiés de longueur $n/2$: L (Left) et R (Right). Notons que *n* doit être pair.

Définition 1.14 : L'algorithme de chiffrement par blocs itératif où la sortie de la i-ème ronde est déterminée d'après la sortie de la ronde précédente [21], [28]:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

K_i est la sous clé utilisée dans la i-ème ronde et *f* est une fonction quelconque.

d. Table de substitution ou table-S

La solidité d'une variété de réseaux de Feistel est directement reliée à leurs tables-S. Une table-S est une simple substitution : une application de *m* bits d'entrée vers *n* bits de sortie. La table avec *m* bits en entrée et *n* bits en sortie s'appelle une table-S à $m \times n$ bits. Les tables-S constituent en général la seule étape non linéaire d'un algorithme ; elles sont ce qui lui donne sa sécurité. En général, plus elles sont grandes, meilleurs elles sont.

r \ c	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

r: rang
c: colonne

Tableau 1.01 : *Détail d'une table-S de l'algorithme DES*

e. Mode de chiffrement

Un mode cryptographique combine en général un algorithme avec une sorte de rétroaction et des opérations simples [22].

- Mode ECB (Electronic CodeBook)

Le mode du carnet de codage électronique ou ECB est la méthode la plus évidente pour utiliser un algorithme de chiffrement par blocs : un bloc de texte en clair se chiffre en un bloc de texte chiffré. Comme un même bloc de texte en clair sera toujours chiffré en un même bloc de texte chiffré, il est théoriquement possible de créer un carnet de codage de textes en clair et de textes chiffrés correspondants.

C'est le mode opératoire le plus simple à utiliser et chaque bloc est chiffré indépendamment des autres blocs.

Le défaut du mode ECB est que si un cryptanalyste obtient le texte en clair et le texte chiffré de plusieurs messages, il peut commencer à construire un carnet de codage sans connaître la clé.

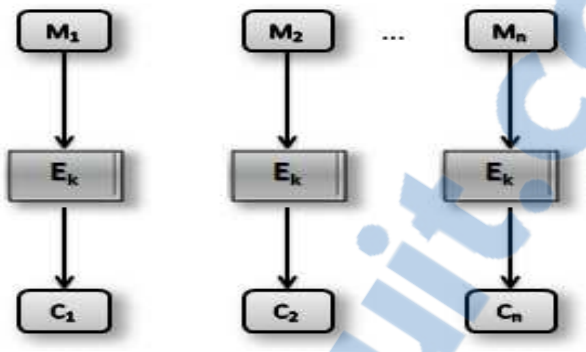


Figure 1.07: *Mode ECB*

Pourtant, si la taille du bloc est de 64 bits, le carnet de codage aura 2^{64} entrées, ce qui est assez grand pour être précalculé et stocké ; de plus, chaque clé a un carnet différent.

- Mode CBC (Cipher Block Chaining)

En mode chiffrement avec chaînage de blocs ou CBC, le texte en clair est combiné par *ou exclusif* avec le bloc chiffré précédent avant d'être chiffré.

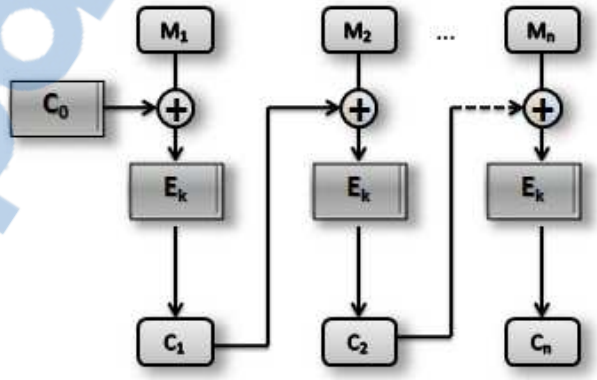


Figure 1.08: *Mode CBC*

Après qu'un bloc de texte en clair a été chiffré, le texte chiffré correspondant est aussi stocké dans un registre de rétroaction. Avant que le bloc du texte en clair suivant soit chiffré, il est combiné par *ou exclusif* avec le registre de rétroaction pour devenir la

nouvelle entrée de l'algorithme de chiffrement. Le texte chiffré résultant est à nouveau stocké dans le registre pour être combiné par *ou exclusif* avec le bloc de texte en clair suivant, et ainsi de suite jusqu'à la fin du message.

Le chiffrement de chaque bloc dépend de tous les blocs précédents. Un bloc de texte chiffré est déchiffré normalement, et aussi sauvé dans le registre de rétroaction. Une fois que le bloc suivant a été déchiffré, il est combiné par *ou exclusif* avec le contenu du registre de rétroaction. Ensuite, le bloc suivant de texte chiffré est sauvé dans le registre de rétroaction, et ainsi de suite jusqu'à la fin du message.

Le mode CBC présente deux inconvénients : d'abord, il impose de déchiffrer la totalité du fichier avant de pouvoir accéder à une partie de ce fichier ; ensuite, une erreur sur un bloc va se répercuter sur tous les blocs suivants. Ainsi, avec ce mode, on sera très attentif au contrôle d'erreurs.

- Mode CFB (Cipher FeedBack)

En mode CFB, les données peuvent être chiffrées par unités plus petites que la taille d'un bloc. On peut utiliser un mode CFB à *n* bit où *n* est inférieur ou égal à la taille du bloc. Ce mode manipule une file d'attente de la taille d'un bloc d'entrée.

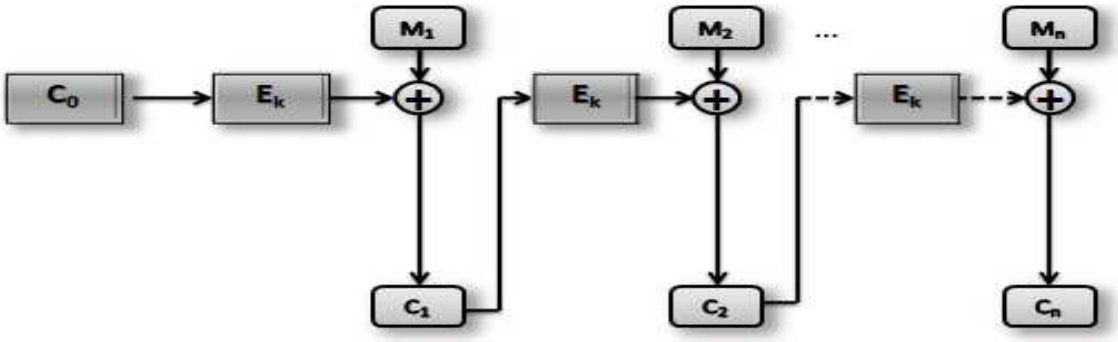


Figure 1.09: *Mode CFB*

Comme le mode CBC, une erreur dans le texte chiffré en mode CFB affecte tous les textes chiffrés suivants. En mode CFB, l'algorithme de chiffrement par blocs est utilisé comme un algorithme de chiffrement autosynchrone en continue que nous allons voir dans la suite.

- Mode OFB (Output-FeedBack)

Le mode de rétroaction de sortie ou OFB est similaire au mode CFB, sauf que le bloc de sortie précédent est mis dans la file d'attente. C'est une méthode qui consiste à utiliser un algorithme de chiffrement par blocs comme un algorithme de chiffrement synchrone en continu.

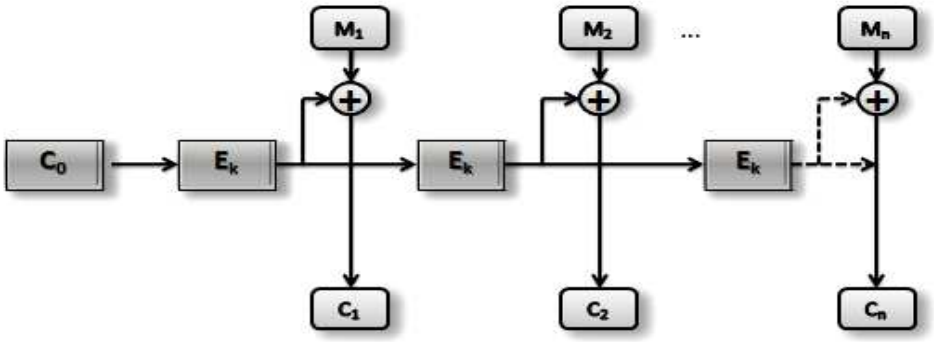


Figure 1.10: *Mode OFB*

Avec le mode OFB, il n'y a pas d'amplification d'erreur c'est-à-dire, une erreur d'un seul bit dans le texte chiffré n'occasionne qu'une erreur d'un seul bit dans le texte en clair récupéré.

- Choix du mode

Si la simplicité et la vitesse sont les critères principaux, le mode ECB est le mode le plus rapide et facile à utiliser. Pour le chiffrement de données aléatoires, l'ECB est aussi un bon mode à utiliser tandis que pour un texte en clair normal, on utilise souvent les modes CBC, CFB, ou OFB. Le mode CBC convient en général pour chiffrer des fichiers. D'ailleurs, si l'application est une réalisation logicielle, ce mode est presque toujours le meilleur choix.

Le mode CFB est généralement le mode de choix pour chiffrer des flots de caractères quand chaque caractère doit être traité individuellement. Le mode OFB est généralement utilisé pour les systèmes synchrones à grande vitesse où la propagation d'erreurs est inacceptable. Le mode OFB est le mode de choix dans un environnement où il y a beaucoup d'erreurs, car il ne les propage pas.

1.7.1.4 Le DES et son successeur.

a. Historique

Jusque dans les années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence des besoins civils, le NBS (National Bureau of Standards) lança le 15 mai 1973 un appel d'offre dans le Fédéral Register (l'équivalent du journal Officiel américain) pourra création d'un système cryptographique.

Le cahier des charges était le suivant :

- l'algorithme repose sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement.

- l'algorithme doit être facile à implémenter, logiciellement et matériellement, et doit être très rapide.

Le chiffrement doit avoir un haut niveau de sûreté, uniquement lié à la clé, et non à la confidentialité de l'algorithme [23].

Les efforts conjoint d'IBM, qui propose Lucifer (fin 1974), et de la NSA (National Security Agency) conduisent à l'élaboration du DES (Data Encryption Standard), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XX^{è} Siècle.

b. Schéma général.

Le DES utilise une clé K de 56 bits, pour chiffrer des blocs de 64 bits, les blocs chiffrés obtenus ayant aussi 64 bits. Le schéma général de DES est représente par la figure 1.11 (on a seulement représenté quelques-unes des 16 étapes).

Le bloc de texte clair subit d'abord une permutation initiale. Puis on itère 16 fois une procédure identique décrite ci-dessus, où la moitié droite est recopiée telle qu'elle à gauche, et la moitié gauche est transmise à droite en subissant au passage une modification dépendante de la clé. A la fin, on inverse les moitiés gauches et droites (ou bien, comme sur le schéma, on supprime le croisement de la dernière étape), et on applique l'inverse de la permutation initiale pour obtenir le bloc chiffré [23], [29].

Le tableau 1.02 suivant montre la permutation initiale et son inverse

Permutation initiale

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

Permutation initiale inverse

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

Tableau 1.02 : *La permutation initiale et son inverse*

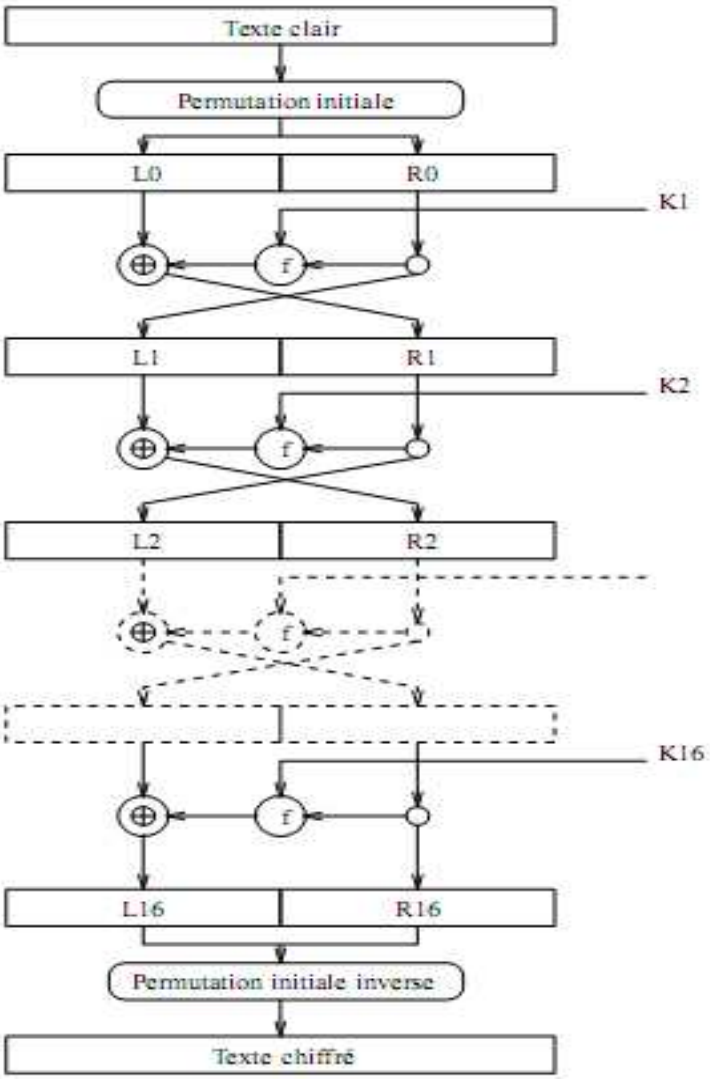


Figure 1.11: *DES : Schéma général*

La permutation initiale et son inverse sont décrits par le tableau 1.02. Les tableaux se lisent de gauche à droite et de haut en bas, le n-ième nombre est la position avant permutation du bit qui se trouve en n-ième position après permutation.

Après la permutation initiale, le message est séparé en deux moitiés de 32 bits, désignées par L_0 et R_0. A chaque itération de la procédure, on détermine deux groupes de 32 bits L_j et R_j en fonction de L_{i-1} et R_{i-1} obtenus précédemment.

Pour cela, on utilise une clé intermédiaire *K*, de 48 bits, calculée à partir de *K* et on applique les formules suivantes :

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

La fonction *f* est schématisée par la figure ci-dessus. Tout d'abord, l'argument de gauche qui possède 32 bits est expansé en 48 bits en redoublant certains bits.

Ensuite, on calcule le *ou exclusif* de cet argument expansé avec le deuxième argument (qui n'est autre que la clé K_i).

Le résultat possède 48 = 8x6 bits et est transformé en une chaîne de 32 = 8 x 4 bits en utilisant des dispositifs appelés boite-S qui calculent un bloc de 4 bits à partir d'un bloc de 6 bits.

Enfin on applique la permutation décrite par la figure 1.11 à ces 32 bit pour obtenir la valeur de *f*.

c. Les boites-S

Il y a huit boites-S différentes. On les représente par deux tableaux à 4 lignes et 16 colonnes. Les premiers et derniers bits de l'entrée déterminent une ligne du tableau, les autres bits déterminent une colonne. La valeur numérique trouvée à cet endroit indique la valeur des quatre bits de sortie.

1.7.1.5 Le triple DES

Puisque la faiblesse de DES est la faible longueur de sa clé, il est naturel de chercher à combiner plusieurs chiffrements pour obtenir un système de chiffrement global avec une clé plus longue.

La première idée qui vient à l'esprit est de composer deux chiffrements DES avec des clés différentes. Mais on peut alors monter contre ce « double-DES » une attaque à message clair dite « par le milieu » parce qu'elle s'appuie sur le message intermédiaire (inconnu) apparaissant entre les deux chiffrements DES successifs. Cette attaque consiste à construire la liste des messages intermédiaires possibles en chiffrant par DES un clair avec les 2^{56} clés possibles [21].

En déchiffrant par DES le chiffré correspondant avec des clés différentes, on obtient une autre liste de messages intermédiaires possibles et le véritable message intermédiaire est dans l'intersection des deux listes. Le coût en mémoire de cette attaque est très important mais son coût en temps n'est pas significativement plus élevé que l'attaque exhaustive sur DES.

Triple-DES consiste à composer deux chiffrements DES de même clé séparées par un déchiffrement DES avec une autre clé. Plus précisément :

$$Triple - DES_{K_1,K_2} = DES_{K_1} \circ DES_{K_2}^{-1} \circ DES_{K_1}$$

Cette façon de faire est préférée à trois chiffrements parce qu'elle généralise DES qui se trouve être le cas particulier où AT, $=K_2$. Bien sûr, le déchiffrement est :

$$Triple - DES_{K_1,K_2}^{-1} = DES_{K_1}^{-1} \circ DES_{K_2} \circ DES_{K_1}^{-1}$$

Une clé triple DES est donc composée de deux clés DES et fait 112 bits ce qui met Triple DES largement hors de portée d'une attaque exhaustive. On peut aussi concevoir une variante à trois clés DES différentes mais elle reste vulnérable à une attaque de coût en 2^{112} s'appuyant sur l'un des deux messages intermédiaires [20], [22].

1.7.1.6 L'IDEA

Développé à Jurich en Suisse par Xuejia Lai et James Massey en 1992, L'International Data Encryption Algorithm (IDEA) a été proposé pour remplacer le DES. Il utilise une clé de 128 bits, réalise un chiffrement par blocs de 64 bits, en opérant 8 rondes d'une même fonction [4]. La description de cette fonction est un peu compliquée, elle utilise seulement trois opérations [2]:

- Ou exclusif
- Addition module 2^{16}
- Multiplication modulo 2^{16}

IDEA est particulièrement adapté aux réalisations logicielles. Il est aussi efficace même sur des processeurs de 16 bits.

a. Algorithme.

Le bloc de 64 bits en entré est tout d'abord divisé en quatre blocs de 16 bits : $XI, X2, X3, X4$. La clé K est divisée en 8 blocs de 16 bits, puis décalée circulairement sur la gauche de 25 bits, et redivisée, et ainsi de suite jusqu'à obtenir 52 clés. Ces clés formeront 8 groupes de 6 clés (un groupe par ronde) : $Kl, K2, K3, K4, K5, K6$, un groupe de 4 clés pour la ronde finale : $Kl, K2, K3, K4$.

Etapes des 8 rondes :

- $Etape1 = X1 * K1$
- $Etape2 = X2 * K2$
- $Etape3 = X3 * K3$
- $Etape4 = X4 * K4$
- $Etape5 = Etape1 \oplus Etape3$
- $Etape6 = Etape2 \oplus Etape4$
- $Etape7 = Etape5 \oplus K5$

- $Etape8 = Etape6 + Etape7$
- $Etape9 = Etape8 * K6$
- $Etape\ 10 = Etape7 + Etape9$
- $Etape11 = Etape1 \oplus Etape9$ ⇒ XI de la ronde suivante
- $Etape12 = Etape3 \oplus Etape9$ ⇒ X3 de la ronde suivante
- $Etape13 = Etape2 \oplus Etape10$ ⇒ X2 de la ronde suivante
- $Etape14 = Etape4 \oplus Etape10$ ⇒ X4 de la ronde suivante

Pour finir, on applique une étape supplémentaire après la huitième ronde :

- $C1 = X1 * K1$
- $C2 = X2 + K2$
- $C3 = X3 + K3$
- $C4 = X4 * K4$

Les 4 blocs Cl, C2, C3, C4, forment alors le message chiffré.

Les 52 sous clés générées à partir de la clé de 128 bits sont produits comme suit :

- La clé de 128 bits est divisée en 8 blocs. Ces 8 blocs sont en faite les 8 premiers sous clés utilisées dans le chiffrement.
- La clé de 128 bits est ensuite cycliquement décalée de 25 positions et divisée en 8 blocs de 16 bits. Ces 8 blocs sont les 8 sous clés suivantes utilisées dans le chiffrement.
- Le cycle est répété jusqu'à ce que les 52 sous clé soient toutes générées.

b. Déchiffrement :

Le déchiffrement s'effectue essentiellement à la même manière que le chiffrement. La seule différence est que les 52 sous clés sont générées de façon inverse au chiffrement. Aussi les blocs de textes chiffrés doivent êtres traités dans l'ordre inverse de chiffrement pour inverser parfaitement le processus de chiffrement.

1.7.1.7 L'AES ou Rijndael.

L'Advanced Encryption Standard est issu d'une compétition et d'une expertise qui s'échelonne de l'appel à candidature par le NIST en 1997 jusqu'à la sélection finale du candidat Rijndael en octobre 2000. Tout comme son prédécesseur, ce standard a été l'occasion de formaliser des critères de conception réunissant l'état d'art des dernières connaissances en cryptographie [2].

Trois critères principaux ont été respectés dans sa conception :

- Résistance face à toutes les attaques connues
- Rapidités du code sur la plus grande variété de plateforme possible.
- Simplicité dans la conception.

Dans l'AES [22], [28], [29], le bloc comporte 128 bits. La clé peut comporter 128, 196 ou 256 bits, ce qui influent le nombre de tours du chiffrement. La version à 128 bits, comporte 10 itérations.

L'algorithme de cadencement de clé produit des sous clés de la même taille que la clé maîtresse.

La première itération est précédée par l'addition de la sous-clé numéro 0 qui correspond à la clé-maitre et la dernière itération ne comporte pas de *MixColums* ou fonction de brouillage de colonne. La fonction itérée se compose du quadruplet de fonctions *(SubBytes, ShiftRows, MixColums, AddRoundKey),* cette dernière fonction étant simplement l'addition bit-à-bit de la sous clé au bloc courant. On y retrouve trois étapes, conformément aux principes fondamentaux de confusion et de diffusion énoncés par Shannon.

La première étape, dite de confusion, la fonction *SubBytes,* consiste à appliquer à chacun des 16 octets de l'entrée une même permutation S. Cette fonction correspond (à une application affine près) à la fonction inverse dans le corps fini à 2^8 éléments ; elle assure la résistance de l'algorithme aux attaques différentielle et linéaire.

La phase de diffusion est composée des fonctions *ShiftRows* et *MixColums* qui représentent des opérations simples sur le corps à 2^8 éléments. Enfin on effectue un *Ou exclusif* bit-à-bit entre le résultat et la sous-clé de l'itération.

Les sous clés de 128 bits, numérotées de 0 à 10, sont dérivées de la clé secrète de la manière suivante : le sous-clé numéro 0 correspond à la clé secrète ; ensuite, la sous-clé numéro *i* (utilisée à la i-ème itération) est obtenu à partir de la sous-clé numéro $(i-1)$.

On permute de manière circulaire, par la fonction *RotWord*, les quatre derniers octets de la clé numéro $(i-1)$, puis on leur applique la fonction *SubWord* composée de 4 permutations S. Après avoir ajouté une constante (dépendant de i) au premier octet (les trois autres octets de la constante $C[i]$ du schéma sont nuls), on effectue une addition bit-à-bit entre les quatre octets ainsi obtenus et les quatre premiers octets de la sous-clé précédente. Les trois autres blocs de quatre octets de la clé numéro i sont ensuite simplement le résultat d'un *ou exclusif* entre le bloc correspondant de la sous-clé $(i-1)$ et le bloc précédent de la sous-clé i.

Toutes les opérations réalisées lors de chiffrage sont réversibles à condition d'avoir la clé. Pour déchiffrer, on procède à l'extension de la clé de la même manière qu'un chiffrage. Les additions par *ou exclusifs* lors de l'opération d'addition de la clé de la tour sont réversibles. L'opération de transformation *SubBytes* est inversée en utilisant la table-S inversée. Les décalages de l'opération de décalage *(ShiftRows)* sont inversés, c'est-à-dire vers la droite. La manipulation matricielle de l'opération de brouillage *(MixColums)* nécessite une inversion de la matrice.

1.7.1.8 Blowfish

Blowfish [29] est un algorithme qui a été conçu par Bruce Schneider. C'est un algorithme de chiffrement par blocs de 64 bits à longueur de clé variable. L'algorithme se fait en deux phases : expansion de la clé et chiffrement des données. L'expansion de la clé convertit une clé de taille inférieure à 448 bits en plusieurs tableaux de sous-clés d'une taille totale de 4168 octets.

Le chiffrement est obtenu en itérant 16 fois une fonction simple. Chaque ronde consiste en une permutation dépendante de la clé et une substitution dépendante de la clé et des données. Toutes les opérations sont des additions et des *ou exclusif* sur des nombres de 32 bits. La seule opération supplémentaire est la consultation des quatre tableaux indexés par ronde.

Blowfish utilise un grand nombre de sous-clés. Ces sous-clés doivent être précalculées avant tout chiffrement ou déchiffrement.

Le tableau-P est un tableau constitué de 18 sous-clés de 32 bits :

$P_1, P_2, \dots, P_{18}$

Il y a 4 tables-S de 32 bits ayant 256 entrées chacune :

$$S_{1,0}, S_{1,1}, \dots, S_{1,255} \qquad S_{2,0}, S_{2,1}, \dots, S_{2,255} \qquad S_{3,0}, S_{3,1}, \dots, S_{3,255} \qquad S_{4,0}, S_{4,1}, \dots, S_{4,255}$$

Blowfish est un réseau de Feistel constitué de 16 rondes. L'entrée est un élément de données *x* de 64 bits.

Pour chiffrer, procéder de la manière suivante :

- Partager *x* en deux moitiés de 32 bits : L et R ;

- Pour i variant de 1 à 16, effectuer :

- $L \longleftarrow L \oplus P_i$
- $R \leftarrow f(L) \oplus R$
- $Echanger\ L\ et\ R$
- $Echanger\ L\ et\ R\ (annuler\ le\ dernier\ échange)$
- $R \longleftarrow R \oplus P_{17}$
- $L \longleftarrow L \oplus P_{18}$
- $Remettre\ L\ et\ R\ bout$ à $bout$.

La fonction f est la suivante, en diviser L en quatre parts de 8 bits a, b, c, d, nous avons :

$$f(L) = \left(\left(S_{1,a}, S_{2,b} mod\ 2^{32}\right) \oplus S_{3,c}\right) + S_{4,d} mod 2^{32} \qquad (1.31)$$

Le déchiffrement s'effectue exactement de la même manière en inversant l'ordre des P_i.

1.7.2 Algorithme à clé publique

Les algorithmes à clé publique sont conçus de telle manière que la clé de chiffrement soit différente de la clé de déchiffrement. De plus, la clé de déchiffrement ne peut pas être calculée à partir de la clé de chiffrement.

De tels algorithmes sont appelés algorithmes « à clé publique » parce que la clé de chiffrement peut être rendue publique : n'importe qui peut l'utiliser pour chiffrer un message mais seul celui qui possède la clé de déchiffrement peut déchiffrer le message chiffré résultant.

Dans de tels systèmes, la clé de chiffrement est appelée clé publique et la clé de déchiffrement est appelée clé privée. La clé privée est aussi appelée clé secrète. Parfois, les messages seront chiffrés avec la clé privée et déchiffrés avec la clé publique ; une telle technique est utilisée pour les signatures numériques.

1.7.2.1 RSA

En 1978, l'algorithme à clé publique de Ron Rivest, A Shamir, Léonard Adleman apparaît dans [3] et [34]. C'est un système à clé publique car l'algorithme n'est pas caché, ni la clé de chiffrement.

Son fonctionnement est basé sur la difficulté de factoriser de grands entiers [26].

a. Fonctionnement

Pour chiffrer le message l'émetteur va utiliser la clé publique que le destinataire a préalablement publiée. La clé publique est un ensemble de lettre et chiffre qui vont permettre à quiconque veut lui envoyés des messages confidentiels.

Cela veut dire que tout le monde peut connaître la clé publique qui permet de crypté les messages uniquement au destinataire qui a publié la clé mais seul ce dernier pourra déchiffrer ce qui a été codé avec sa clé publique grâce à sa clé privée.

b. Génération de clé :

Choisir deux nombres entier premier p et q, de l'ordre de 100 chiffres au minimum, pour rendre la factorisation hors de porter, même avec les meilleurs ordinateurs.

Algorithme :

1) Calculer $n = p * q$

2) Calculer $m = (p - 1)(q - 1)$

3) Choisir un nombre entier *e* tel que *e > 2* et $pgcd(e, m) = 1$

4) Calculer d tel que $d * e * mod\ m = 1$

On prendra comme clé publique e et n et comme clé privée d *et* n.

c. Chiffrement

Connaissant la clé publique *e* et *n*, et le message à chiffrer M. On peut le chiffrer de la manière suivante, le message doit être remplacé par un chiffre. Ensuite on découpera le message par bloc de *x* longueurs avec *x < n.*

Le bloc B est chiffré par la formule :

$$C = B^e * mod(n)$$

C'est un bloc de message chiffré à envoyer vers le destinataire.

d. Déchiffrement

Pour le déchiffrement on va pratiquer quasiment de la même façon que le chiffrement mais avec la formule inverse :

$$b = C^d * mod(n)$$

Ce qui permettra au destinataire de trouver le message clair.

1.7.2.2 Cryptosystème d'El Gamal

a. Principe

Dans ce système [40], on suppose que les blocs de texte clair sont numérisés dans les entiers $modulo\ p$.

1) On choisit un grand nombre premier p, et un nombre $g\ (mod\ p)$, qui sont tous deux connus de tous.

2) L'utilisateur A choisit un grand nombre $a\ (mod(\ p-1))$ qui sera sa clé secrète de déchiffrement. La clé publique de A est le nombre $g^a\ (mod\ p)$

3) Pour envoyer un message m à A, l'utilisateur B choisit aléatoirement un grand nombre entier $k(mod\ p)$, et il l'envoie à A la paire :

$$(K, M), \text{où } K = (g^k mod\ p), \text{ et } M = (m\ g^{ak}\ mod\ p)$$

4) Le receveur A, qui connaît la clé secrète *a*, récupère le message *m* à partir de cette paire de la façon suivante. Il calcule d'abord :

$$(K^{-a} mod\ p) = (g^{-aK}\ mod\ p)$$

à partir du premier élément du couple reçu ; puis il multiplie M par ce résultat pour obtenir

$$\begin{aligned} Mg^{aK} &\equiv (mg^{aK})g^{-aK}(modp) \\ &\equiv mg^{aK-aK}(modp) \\ &\equiv m \end{aligned}$$

Intuitivement, le message chiffré C envoyé à A est une version masquée de *m* obtenue par la multiplication par g^{ak}. Le nombre *K*, qui accompagne le message chiffré *C*, est un indice qui permet à A de retirer le masque. Cet indice $K = (g^k\ mod\ p)$ ne peut être utilisé que par quelqu'un qui connaît la clé *a*.

Il semble que pour qu'un cryptanalyste puisse casser le cryptosystème de El Gamal, il doive retrouver la clé *a* à partir de la clé publique g^a. C'est à dire qu'il aura trouvé une solution efficace au problème du calcul du logarithme discret.

1.7.2.3 PGP

Le PGP [20] (Pretty Good Privacy) est un cryptosystème inventé par Phil Zimmermann en 1991. Il combine à la fois les meilleures fonctionnalités de la cryptographie à clé privée et de la cryptographie à clé publique. PGP est donc un système hybride.

a. Chiffrement.

Quand un utilisateur chiffre du texte clair avec PGP, PGP compresse d'abord le texte clair. La compression de données économise le temps de transmission des données et de l'espace disque et, ce qui est important, renforce la sécurité cryptographique. La plus part des techniques de cryptanalyse exploitent les redondances trouvés dans le texte clair pour craquer le texte chiffré. La compression réduit ces redondances dans le texte clair, ce qui augmente grandement la résistance à la cryptanalyse.

PGP crée ensuite une clé de session, qui est une clé secrète qui ne sert qu'une fois. Cette clé est un nombre aléatoire généré à partir des mouvements aléatoires de votre souris et des touches du clavier sur lesquelles vous tapez. Cette clé de session fonctionne avec un algorithme de chiffrement conventionnel très sûr et rapide qui chiffre le texte clair ; le résultat est le texte chiffré. Une fois que les données sont chiffrées, la clé de session est elle-même chiffrée avec la clé publique du destinataire. Cette clé de session chiffrée par la clé publique est transmise avec le texte chiffré au destinataire.

b. Déchiffrement.

Le déchiffrement fonctionne de la manière inverse. La copie de PGP du destinataire utilise la clé privée de celui-ci pour retrouver la clé de session temporaire, que PGP utilise ensuite pour déchiffrer le texte chiffré de manière conventionnelle.

La combinaison des deux méthodes de chiffrement (IDEA, RSA) associe la commodité du chiffrement à clé publique avec la vitesse du chiffrement conventionnel. Le chiffrement conventionnel est environ 1000 fois plus rapide que le chiffrement à clé publique. Le chiffrement à clé publique fournit quant à lui une solution aux problèmes de distribution de la clé et de transmission des données. Utilisées toutes les deux, la performance et la distribution de la clé sont améliorées sans aucun sacrifice sur la sécurité.

1.7.2.4 Le cryptosystème de RABIN

La cryptanalyse totale d'une instance du cryptosystème RSA revient à factoriser N. Par contre, personne n'a pu prouver que cette factorisation est nécessaire pour une cryptanalyse partielle (déchiffrer un message). La sécurité du système suivant, contre une attaque à messager claire choisie, est en revanche équivalente à la factorisation de son module. En gros, ce système consiste à remplacer l'exposant du système de chiffrement de RSA par 2.

♣ Toutefois, puisque $pgcd(2,(p-1)(q-1))$ est toujours strictement supérieur à 1, l'élévation au carrée modulo N n'est plus une fonction injective. ♦

a. Principe

1) Le destinataire Bob choisit un entier N qui soit le produit de deux grand nombres entiers distincts $p\ et\ q$, congrus à $3\ modulo\ 4$. Il choisit aussi un entier b dans l'intervalle $[0, N-1]$. Il diffuse l'entier $N\ et\ b$, tout en gardant secret $p\ et\ q$. Il lui sera utile de calculer les coefficients de Bézout u, v telles que $up + vq = 1$.

2) Pour chiffrer l'élément $m \in \mathbb{Z}_N$, Alice calcule $c = m(m+b)\ mod N$.

3) Lorsque Bob reçoit c, il déchiffre en calculant les quatre racines carrées modulo N de $c + b^2/4$. Pour cela, il peut calculer d'abord les racines $r_p\ et\ r_q\ modulo\ p\ et\ q$, ce qui est particulièrement facile puisque : $p \equiv q \equiv 3\ modulo\ 4$. Soient alors les valeurs de r_p et r_q suivantes :

$$r_p = c^{(p+1)/4} mod\ p, \qquad r_q = c^{(q+1)/4} mod\ q$$

4) Puis, il utilise le théorème des restes chinois pour obtenir les quatre racines modulo N, soit la relation :

$$r = \pm upr_q \pm vqr_p mod\ N$$

Le message initial est l'une des quatre valeurs $r - b/2$. En effet on a :

$$\left(r - \frac{b}{2}\right)\left(r - \frac{b}{2} + b\right) \equiv \left(r - \frac{b}{2}\right)\left(r + \frac{b}{2}\right) \equiv r^2 - \frac{b^2}{4} \equiv \left(c + \frac{b^2}{4}\right) - \frac{b^2}{4} \equiv c(mod N)$$

5) Déchiffrer le schéma de Rabin revient à calculer une racine carrée de $c - \frac{b^2}{4}\ mod N$

Quelqu'un qui saurait comment décrypter saurait donc calculer des racines carrées modulo N, donc pourrait facilement factoriser N

Un inconvénient évident du schéma de Rabin est que le déchiffrement est ambigu. Précisément pour $m \in \mathbb{Z}_N$, les quatre messages obtenus :

$$m, \quad -m-b, \quad \omega m + \frac{(\omega-1)b}{2}, \quad -\omega m - (\omega+1)b/2 \tag{1.32}$$

donnent le même message chiffré, ω étant une racine carrée non triviale de 1 modulo N.

Au déchiffrement, il faut pouvoir lever l'ambigüité sur les quatre messages clairs possibles.

Cela peut se faire en pratique en ajoutant de la redondance aux messages clairs (par exemple redoubler les 64 premiers bits). Avec une très grande probabilité, une seule des quatre solutions possible au déchiffrement respectera cette redondance.

Si le schéma de Rabin est sûr (si on admet que le problème de la factorisation est difficile) vis à vis d'une attaque à message clairs choisis, il est par contre vulnérable contre une attaque à message chiffrés choisis. En effet, si un attaquant choisit un message m au hasard, calcul $c = m(m+b)$ et obtient un déchiffrement m' de c, alors il y a une chance sur deux pour que m' soit différent de m et de $-m-b$. Il peut dans ce cas calculer ω et factoriser N. L'ajout de redondance rend cette attaquer inutilisable. Toutefois, on ne sait pas rigoureusement prouver que la sécurité du schéma de Rabin avec redondance est assujettie à la factorisation.

1.7.2.5 Cryptosystème basé sur les codes correcteurs

- Le cryptosystème de Mc Eliece [42]

On choisit un code correcteur sur un corps K pour lequel on connait un algorithme efficace de décodage (par exemple un code de Goppa). Ce code C est donné par une matrice génératrice G de taille $k \times n$, où $n\ et\ k$ sont la longueur et la dimension de C.

On choisit aussi une automorphisme de K^k, donnée par une matrice inversible U, et une isométrie (pour la distance de Hamming) de K^n, donnée par une matrice de permutation P, qui vont servir à masquer la nature du code initial C qui lui, restera secret. La clé publique sera constituée de la matrice $\widehat{G} = UGP$, et la clé privée des trois matrices U, G, P.

L'espace des messages clairs est K^k, et chaque message m sera chiffré en l'encodant avec la matrice $\hat{G}$ et en ajoutant une erreur aléatoire e de poids t, la capacité correctrice de C :

$$c = m\hat{G} = e \; avec \;\; \omega(e) = t \tag{1.33}$$

Le message chiffré c obtenu est un élément de K^n. Pour le déchiffrer, le destinataire lui applique la permutation obtenue par P^{-1}

$$cP^{-1} = mUG + eP^{-1} \tag{1.34}$$

et décode le mot obtenu. Il obtient mU puisque eP^{-1} est de poids t. Il ne lui reste plus qu'à appliquer la transformation déterminée par U^{-1} pour retrouver m.

Il est important que l'expéditeur du message respecte le protocole de chiffrement, en ajoutant un mot aléatoire e de poids après encodage. S'il ne le fait pas, son message clair m peut être retrouvé à partir de message chiffré, sans toutefois que la clé privée soit compromise. Il suffit pour cela qu'un attaquant sélectionne k colonnes de la matrice $\hat{G}$ formant une matrice carrée $\hat{G}_k$ inversible.

Le seul message clair m' tel que l'encodé correspondant $m\hat{G}_k$ coïncide avec c sur les k positions sélectionnées est m et il est facile de le retrouver en résolvant $m\hat{G}_k = c_k$, où c_k désigne les k symboles de c correspondant aux colonnes sélectionnées.

Si au contraire, l'expéditeur ajoute un mot aléatoire e après chaque encodage, la méthode de décryptage ci-dessus ne fonctionne que si les k positions sélectionnées sont en dehors du support de e. Ceci est hautement improbable lorsque les paramètres sont choisis convenablement.

Les paramètres suggérés à l'origine par Mc Eliece étaient $n = 1024$, $t = 50$ et $k \geq 524$. Lorsqu'on base ce cryptosystème sur un code de Goppa, un choix optimal pour la sécurité est n=1024, t=38 et k≥644. Un inconvénient de ce cryptosystème est la très grande taille de la clé publique (environ 2^{19} bits dans le cas des paramètres précédents).

Un autre inconvénient est un taux de transmission k/n nettement inférieur à 1. Ces inconvénients ont empêché ce cryptosystème d'être utilisé en pratique. D'autre part l'étude profonde de sa sécurité n'a encore bénéficié que de peu d'effort comparativement aux cryptosystèmes usuels.

- Cryptosystème de Niederreiter

Sa sécurité est équivalente à celle du précédent. Toutefois, avec de paramètres judicieux, la taille des clés est un peu raisonnable que dans le cas de Mc Eliece [19].

Le code C peut être identique à celui utilisé pour le cryptosystème de Mc Eliece mais il est ici donné par une matrice de contrôle H (de taille $(n-k) \times n$). On masque la matrice H en utilisant une matrice inversible V de taille $(n-k) \times (n-k)$ et une matrice de permutation P de taille n x n. La matrice $\widehat{H} = VHP$ obtenue est une matrice de contrôle du code $\hat{C}$. Elle est rendue publique et H est gardée secrète.

L'espace des messages clairs est cette fois ci l'ensemble des mots de longueurs n et de poids t (donc des erreurs). Le chiffré correspondant au message m est son syndrome relativement à la matrice publique :

$$c = \widehat{H}m^T \; avec \; \omega(\mathrm{m}) = \mathrm{t}$$

Pour déchiffrer, on remarque que V^{-1}c est le syndrome de Pm^T relativement à la matrice secrète :

$$V^{-1}c = HPm^T$$

L'algorithme de décodage dans C (il faut que C possède un algorithme de décodage par symbole efficace) permet de retrouver l' « erreur » Pm^T de poids t. Il ne reste plus qu'a appliquer P^{-1} pour retrouver m.

1.7.2.6 Cryptosystème utilisant les courbes elliptiques

Nous verrons que la résolution du logarithme discret dans les corps finis est accessible sur des corps de taille moyenne grâce à des logarithmes relativement récents (algorithme sous exponentiels). L'existence de ces algorithmes oblige à des modules assez grands dans les cryptosystèmes RSA et El Gamal pour préserver leur sécurité [18], [41]. Mais les problèmes du logarithme discret et de Diffie-Hellman ont des homologues naturels en termes de courbes elliptiques [43], [44], [45].

Par exemple : Soient E une courbe elliptique sur un corps fini et G un point d'ordre h (grand) sur E. Le problème du logarithme discret sur E est celui de trouver, étant donné un point A du sous groupe de E engendré par G, l'entier a vérifiant :

$$aG = A \text{ avec } 0 \leq a \leq h - 1$$

Or, on ne dispose pas actuellement de méthode sous exponentielle pour résoudre le logarithme discret sur les courbes elliptiques. Le problème du logarithme discret sur les courbes elliptiques est donc (provisoirement) plus difficile, à taille égale, que le problème du logarithme discret classique. Il est donc tentant de transcrire les cryptosystèmes basés sur le logarithme discret en termes de courbe elliptiques, pour obtenir des cryptosystèmes de sécurité équivalente avec des clés plus petites.

a. Définition d'une courbe elliptique

Soit K un corps, on appelle courbe elliptique sur K une courbe dans le plan projectif $\mathbb{P}^2(K)$, cubique et sans points singuliers, et munie d'un point distingué qui jouera un rôle particulier : élément neutre. Elle est donc définie par un polynôme irréductible homogène en trois variables à coefficient dans K [3]. Par un changement de variables homographique, on peut toujours se ramener à une équation dite de Weierstrass :

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3 \qquad (1.35)$$

Avec $a_1, a_2, a_3, a_4, a_6 \in K$.

La courbe elliptique E est l'ensemble des points $(x, y) \in K^2$ satisfaisant cette équation et d'un point imaginaire $\mathcal{O}$ appelé point à l'infini [3].

- Loi de Groupe

Les applications des courbes elliptiques en cryptographie sont principalement dues à l'existence d'une loi de groupe que nous pouvons définir sur ces dernières. En effet, l'ensemble $E \cup \mathcal{O}$ peut être équipé avec une opération d'addition qui produit un groupe abélien dont l'élément neutre est le point à l'infini $\mathcal{O}$ [3].

En cryptologie, les courbes elliptiques sont utilisées dans le corps $\mathbb{F}_p$ avec p un nombre premier strictement supérieur à 3.

Soit E une courbe elliptique définie sur $\mathbb{F}_p$. L'équation affine de Weierstrass de E peut être simplifiée, pour $K \neq 2,3$, par :

$$y^2 = x^3 + ax + b \tag{1.36}$$

La courbe définie par cette formule admet un unique point à l'infini (i.e. avec $z = 0$), de coordonnées (0 :1 :0). C'est en général ce point qui sera distingué. On appelle discriminant de cette courbe l'élément $-16(4a^3 + 27b^2)$ de K.

Le facteur entre parenthèse est le discriminant du polynôme membre de droite :

$$4a^3 + 27b^2 \not\equiv 0 (mod\ p) \tag{1.37}$$

Soient $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ deux points sur E :

- Le point $(x_1, -y_1)$ est l'opposé du point P et il est noté −P.

- Si $Q \neq P$ et $Q \neq -P$, alors le point $R = P + Q = (x_3, y_3)$ est défini par :

$$\begin{cases} x_3 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2 \\ y_3 = y_1 + \left(\frac{y_1 - y_2}{x_1 - x_2}\right)(x_3 - x_1) \end{cases} \tag{1.38}$$

- Si $P = Q$, alors le point $2P = (x_3, y_3)$ est défini par :

$$\begin{cases} x_3 = \left(\dfrac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \\ y_3 = x_1 + \left(\dfrac{3x_1^2 + a}{2y_1}\right)(x_3 - x_1) \end{cases} \tag{1.39}$$

- Si $x_1 = x_2$ mais $y_1 \neq y_2$, alors $R = \mathcal{O}$

- Si $P = Q$ et $y_1 = 0$, alors $R = \mathcal{O}$

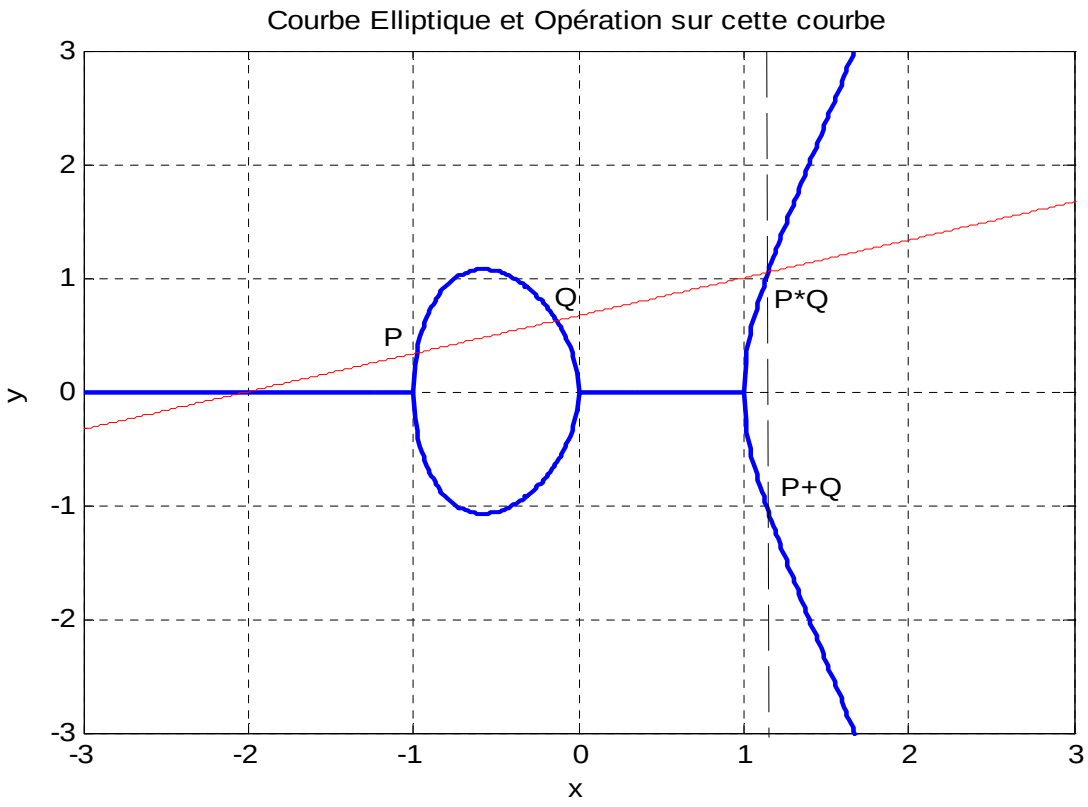


Figure 1.12: *Courbe elliptique d'équation* $y^2 = 3x^3 - 3x$

- Remarque :

Notons $\mathcal{O}$ le point distingué de E. On montre qu'il existe une loi de groupe sur les points de E telle que $\mathcal{O}$ soit l'élément neutre et que l'identité $P + Q + R = \mathcal{O}$ soit vérifiée pour tout triplet de point alignés. Lorsque la convention $\mathcal{O} = (0:1:0)$ est en vigeur, l'opposé de chaque point de E est son symétrique par rapport à l'axe des abscisses [1].

La somme de deux points P, Q finis et non opposés est donc le point de coordonnées $(x_3, -y_3)$ où x_3, y_3 sont donnés par la relation (1.38).

Bien sur, lorsque l'un des points est à l'infini, ou lorsqu' ils sont symétriques l'un de l'autre, on a les identités $P + \mathcal{O} = P$ et $P + (-P) = \mathcal{O}$.

- Calcul du nombre de points d'une courbe elliptique sur un corps fini

Soit E une courbe elliptique d'équation (1.36) sur $\mathbb{F}_p$ (avec p premier).

Le nombre de points de E est donné par l'égalité:

$$\#E = 1 + \sum_{x \in F_p} \left(1 + \left(\frac{x^3 + ax + b}{p}\right)\right)$$

Si le corps de base est un corps fini, la courbe elliptique est un groupe fini et le théorème suivant donne un renseignement très utile sur son ordre [5].

Intéressons-nous sur les courbes elliptiques E sur un corps fini $\mathbb{F}_q$ (avec $q = p^r$).

$E(\mathbb{F}_q)$ a au maximum $2q + 1$ points c'est à dire le point à l'infini plus $2q$ paires $(x, y) \in \mathbb{F}_q^2$

Théorème 1.16 : (Théorème de Hasse)

Soit une courbe elliptique sur le corps fini $\mathbb{F}_q$. Le nombre de point de E vérifie :

$$q + 1 - 2\sqrt{q} \leq \#E \leq q + 1 + 2\sqrt{q}$$

D'autre part, deux éléments suffisent pour engendrer le groupe d'une courbe elliptique et, bien souvent, il est même cyclique.

- Problème du logarithme discret

Soit G un groupe (noté additivement) cyclique fini d'ordre N engendré par un élément P.

Soit Q un élément de G. Comme G est un groupe cyclique engendré par P, il existe un unique entier k compris entre 1 et N tel que $Q = kP$. Cet entier k est appelé le logarithme discret de Q en base P et nous le noterons $log_P(Q)$ [3].

Cependant le groupe que nous utiliserons par la suite est usuellement noté additivement et nous garderons donc les notations initiales. Notons bien que kP signifie $P + P + ... + P$ k fois où le signe « + » est la loi du groupe G.

La multiplication scalaire d'un point par un entier est l'opération de base et l'opération la plus chère des protocoles basés sur les courbes elliptiques.

Pour réduire le temps de calcul, on peut utiliser des algorithmes très connus d'exponentiation pour le calcul de $[k]P$ pour un k grand. On peut également améliorer le calcul pour un k petit.

- Problème du logarithme discret sur les courbes elliptiques

Les cryptosystèmes utilisant les courbes elliptiques sont basés sur l'analogue du problème du logarithme discret sur les courbes elliptiques.

Soit E une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ et soit P un point sur la courbe E.

Le problème du logarithme discret sur les courbes elliptiques (noté par ECDLP, Elliptic Curve Discrete Logarithm Problem) consiste à trouver un nombre k étant donné le point P et le point $Q = kP$ où $kP = P + P + \cdots + P$

Actuellement, le meilleur algorithme pour résoudre le ECDLP est en temps exponentiel en la taille de la clé k, en contraste avec les algorithmes sous-exponentiels connus pour factoriser les grands entiers. Ceci permet donc aux cryptosystèmes basés sur les courbes elliptiques d'utiliser, à sécurité équivalente, des clés beaucoup plus courtes que les cryptosystèmes asymétriques classiques comme RSA ou ElGamal [6].

- Protocole d'échange de clés de Diffie-Hellman

Un des moyens pour sécuriser les données transitant entre l'émetteur et la récepteur est qu'ils établissent une clé privée entre eux. La méthode de Diffie-Hellman [7] permet justement de faire cela.

1) Alice et Bob choisissent une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$ tel que le logarithme discret soit difficile à résoudre. Ils choisissent aussi un point $P \in \mathbb{F}_q$ tel que le sous-groupe généré par P ait un ordre de grande taille. (E et P sont choisis dont l'ordre soit un grand nombre premier.)

2) Alice choisit un nombre entier secret a, calcule $P_a = aP$ et envoie P_a à Bob.

3) Bob choisit un nombre entier secret b, calcule $P_b = bP$ et envoie P_b à Alice.

4) Alice calcule $aP_b = abP$ et Bob calcule $bP_a = baP$

5) Alice et Bob utilisent une méthode quelconque connue pour extraire une clé secrète de abP. Par exemple, ils peuvent utiliser les derniers 256 bits de la première coordonnée de abP comme clé, ou ils peuvent hacher une des coordonnées de abP avec une fonction de hachage pour laquelle ils se sont mis d'accord.

- La taille des clés

Sur le groupe des points d'une courbe elliptique définie sur un corps fini, les meilleures attaques connues du logarithme discret sont des attaques génériques.

Lorsque l'on choisit une courbe, on essaie de trouver un groupe E tel que $\#E = h\,\mathcal{L}$ avec h petit et $\mathcal{L}$ premier.

Le point de base P qui servira à la multiplication $[k]P$ sera choisi dans le sous-groupe d'ordre $\mathcal{L}$.

Dans le cas où $h = 1$, $\#E = \mathcal{L}$. D'autre part, le théorème de Hasse dit que $\#E(\mathbb{F}_q) \sim q$.

Les attaques génériques (par exemple l'algorithme Pollard-Rho qui est une attaque par collision basée sur le paradoxe des anniversaires se font en $O(\sqrt{\mathcal{L}})$ [8].

Pour une sécurité de 2^{80} opérations, on voudra donc $\sqrt{\mathcal{L}} \geq 2^{80}$ soit $\sqrt{\mathcal{L}} \geq 2^{160}$, or $\#E(\mathbb{F}_q) \sim q$, pour $\#E(\mathbb{F}_q) \geq 2^{160}$, on choisira $q \geq 2^{160}$. Il nous faut donc travailler avec des corps d'ordre au moins égal à 2^{160}, c'est-à-dire travailler modulo un nombre premier d'au moins 160 bits dans le cas d'un corps premier.

Bien entendu, ce résultat est valable pour tous les niveaux de sécurité, c'est-à-dire que si l'on souhaite une sécurité de 2^n bits, il faut des clés de 2^{2n} bits.

RSA, qui n'est pas basé sur le problème du logarithme discret mais sur la factorisation en facteurs premiers, est sans doute le cryptosystème le plus utilisé mais pour une sécurité de 2^{80} opérations, il nécessite des clés de 1024 bits, c'est-à-dire des clés près de 7 fois plus longues que pour ECC.

Plus les niveaux de sécurité augmentent, plus ce rapport augmente.

Il est donc plus avantageux, en terme de taille de clés, d'utiliser ECC que RSA.

Cependant, si en termes de stockage, ECC est plus performant, on va voire après ce qu'il en est en termes de vitesse de calcul.

b. Chiffrement sur une courbe elliptique

Une transcription directe du cryptosystème d'El Gamal en termes de courbe elliptiques serait imaginable mais pose quelques problèmes techniques (grande taille du message chiffré par rapport à celle du message clair, génération de point de E en fonction du message à transmettre,…). On lui préfère donc la méthode suivante :

- Cryptosystème de Menezes/Vanstone

Soit E une courbe elliptique sur un corps fini et G un point d'ordre h (grand) sur E, rendus publics.

1) Le destinataire Bob choisit un entier $b \in [0, h-1]$ et diffuse la valeur du point $B = bG$.

2) L'expéditeur Alice peut alors chiffrer un message

$$M = (x_M, y_M) \in \mathbb{F}_q * \mathbb{F}_q$$

de la manière suivante (noter que le point M n'est pas nécessairement sur E).

Elle choisit secrètement un entier $k \in [0, h-1]$, calcul $K = kG$ et $kB = (x_{kB}, y_{kB})$.

3) Elle transmet à Bob les données (K, C) où :

$$C = (x_c, y_c) = (x_M x_{kB}, y_M y_{kB}).$$

4) A la réception, Bob peut retrouver M en calculant

$$bK = (x_{bK}, y_{bK}) \text{ et } (x_M, y_M) = (x_C x_{bK}^{-1}, y_C y_{bK}^{-1}).$$

- Algorithme de Massey-Omura

La procédure peut être implémentée de la manière suivante :

1) Alice et Bob choisissent une courbe elliptique E sur un corps fini $\mathbb{F}_q$ tel que le problème du logarithme discret soit difficile à résoudre sur $E(\mathbb{F}_q)$.

2) Alice et Bob calculent $N = \#E(\mathbb{F}_q)$.

3) Alice représente son message comme un point $M \in E(\mathbb{F}_q)$

4) Alice choisit secrètement un entier m_A premier avec N, calcule $M_1 = m_A M$ et envoie le résultat à Bob

5) Bob choisit secrètement un entier m_B premier avec N, calcule $M_2 = m_B M_1$ et envoie le résultat à Alice

6) Alice calcule $m_A^{-1} \in \mathbb{Z}/N\mathbb{Z}$ puis $M_3 = m_A^{-1} M_2$ et envoie le résultat à Bob

7) Bob calcule $m_B^{-1} \in \mathbb{Z}/N\mathbb{Z}$ puis $M_4 = m_B^{-1} M_3$. Alors M_4 est le message initial M.

Démonstration :

♣ Montrons que M_4 est le message initial M. Nous avons formellement

$$M_4 = m_B^{-1} m_A^{-1} m_B m_A M = M$$

Mais nous devons encore justifier pourquoi m_A^{-1} et m_A s'annulent.

Nous avons $m_A^{-1} m_A \equiv 1 \ (mod\ N)$ donc $m_A^{-1} m_A = 1 + kN$ pour un certain entier k. Le groupe $E(\mathbb{F}_q)$ est d'ordre N donc d'après le théorème de Lagrange, $NR = \mathcal{O}$ pour tout $R \in E(\mathbb{F}_q)$.

Ainsi :

$$m_A^{-1} m_A R = (1 + kN)R = R + k\mathcal{O} = R$$

En appliquant ceci avec $R = m_B M$, on obtient :

$$M_3 = m_A^{-1} m_B m_A = m_B M$$

De même, m_B^{-1} et m_B s'annulent, donc :

$$M_4 = m_B^{-1} M_3 = m_B^{-1} m_B M = M$$ ♦

- Algorithme de El Gamal

Soit une communication secrète entre Bob et Alice. Bob choisit une courbe elliptique E sur un corps fini $\mathbb{F}_q$ tel que le problème du logarithmme discret soit difficile à résoudre sur $E(\mathbb{F}_q)$.

Il choisit également un point $P \in E(\mathbb{F}_q)$ tel que l'ordre de P soit un grand nombre premier. Ensuite, il choisit un entier secret s et calcule :

$B = sP$. Bob rend les informations suivantes publiques : $E, \mathbb{F}_q, P, B$.

Sa clé secrète est l'entier s

Pour envoyer un message à Bob, Alice procède ainsi :

1) Utilise la clé publique (P, B) de Bob
2) Représenter le message comme un point M
3) Choisir un entier k et calculer $M_1 = kP$
4) Calculer $M_2 = M + kB$
5) Envoyer M_1 et M_2 à Bob

Bob décrypte le message en calculant

$$M = M_2 - sM_1$$

Le résultat est le bon car :

$$\clubsuit\ M_2 - sM_1 = (M + kB) - s(kP) = M + k(sP) - skP = M \qquad \blacklozenge$$

1.7.3 Choix d'algorithme

La cryptographie à clé publique et à clé secrète sont deux choses différentes ; elles résolvent des problèmes de types différents. La cryptographie à clé secrète est meilleure pour chiffrer un message car elle est infiniment plus rapide.

La cryptographie à clé publique peut faire des choses que la cryptographie à clé secrète ne permet pas ; elle est adoptée pour la gestion des clés.

En effet :

- Les algorithmes à clé publique sont lents. Les algorithmes à clé secrète sont généralement au moins 1000 fois plus rapide que les algorithmes à clé publique.

- Les cryptosystèmes à clé publique sont vulnérables aux attaques à texte clair choisi. Si $C = E(M)$ où *M* est un texte clair parmi *n* textes clairs possibles, alors, il suffit à un cryptanalyste de chiffrer les *n* messages et de comparer les textes chiffrés résultants avec *C* (la clé de chiffrement est publique). Il ne pourra pas trouver la clé de déchiffrement de cette manière, mais il pourra déterminer *M*.

1.7.4 Cryptosystèmes hybrides

Dans la plupart des applications pratiques, la cryptographie à clé publique est utilisée pour protéger et distribuer les clés de session, et ces clés de session sont utilisées dans des algorithmes à clé secrète pour protéger les messages transmis. Cela est parfois appelé un cryptosystème hybride.

Voici le protocole correspondant :

1) Le destinataire B envoi sa clé publique à l'émetteur A ;
2) A engendre une clé de session aléatoire, *k*, la chiffre avec la clé publique de B et envoie le résultat à B :

$$E_B(k)$$

3) B utilise sa clé privée pour déchiffrer le message de A et ainsi retrouver la clé de session :

$$D_B\big(E_B(k)\big) = k$$

4) A et B utilisent alors la même clé de session pour chiffrer leur conversation.

Avec ce protocole, la clé de chiffrement est créée au moment de son utilisation pour chiffrer les communications et elle est détruite dès qu'on n'en a plus besoin. Cela réduit considérablement le risque de compromettre la clé de session.

1.8 Générateurs aléatoires et pseudo-aléatoires

La cryptographie a souvent recours à des nombres aléatoires [3], [9]. Ainsi, lorsqu'une personne génère une clé secrète ou privée, elle doit faire intervenir le hasard de façon à empêcher un adversaire de deviner la clé. De même, certains protocoles cryptographiques nécessitent, pour éviter la rejouabilité par exemple, l'utilisation d'aléas imprévisibles par les opposants. Malheureusement, il est impossible de produire des suites aléatoires à l'aide uniquement d'un ordinateur : le générateur sera toujours périodique, donc prévisible.

On a donc recours à des générateurs dits pseudo-aléatoires et cryptographiquement sûrs. Un tel générateur doit présenter les caractéristiques suivantes :

1) La période de la suite doit être suffisamment grande pour que les sous suites finies utilisées avec l'algorithme ou le protocole cryptographique ne soient pas périodiques.
2) Ces sous suites doivent, sur le plan statistique, sembler aléatoires.
3) Le générateur doit être imprévisible, au sens où il doit être impossible de prédire le prochain aléa à partir des aléas précédents.

La plupart des générateurs pseudo-aléatoires sont construits en utilisant des registres à décalage *(shift registers* en anglais) et, en particulier, les registres à décalage à rétroaction linéaire *(Linear Feedback Shift Registers, LFSR).* Ces derniers présentent l'inconvénient de générer des suites linéaires, si bien que des grands nombres générés à partir de sous-suites sont fortement corrélés.

C'est pourquoi les générateurs pseudo-aléatoires sont généralement construits en combinant, à l'aide d'une fonction non linéaire, plusieurs registres à décalage de tailles différentes. Ce type de générateur est très utilisé par les algorithmes de chiffrement en continu.

Si l'on veut vraiment générer des suites aléatoires, au sens où ces suites sont de plus non reproductibles, on a généralement recours à des éléments extérieurs comme les déplacements de la souris, la vitesse de frappe, l'entrée d'un micro enregistrant le bruit atmosphérique,...

1.9 Fonctions de hachage

1.9.1 Fonctions de hachage à sens unique

Aussi appelée fonction de condensation [3], [36], [38], une fonction de hachage est une fonction qui convertit une chaîne de longueur quelconque en une chaîne de taille inférieure et généralement fixe. La chaîne résultante est appelée empreinte ou condensé de la chaîne initiale.

Une fonction à sens unique est une fonction facile à calculer mais difficile à inverser [3]. La cryptographie à clé publique repose sur l'utilisation de fonctions à sens unique à brèche secrète : pour qui connaît le secret (i.e. la clé privée), la fonction devient facile à inverser.

Une fonction de hachage à sens unique est une fonction de hachage qui est en plus une fonction à sens unique : il est aisé de calculer l'empreinte d'une chaîne donnée, mais il est difficile d'engendrer des chaînes qui ont une empreinte donnée, et donc de déduire la chaîne initiale à partir de l'empreinte. On demande généralement en plus à une telle fonction d'être sans collision**,** c'est-à-dire qu'il soit impossible de trouver deux messages ayant la même empreinte. On utilise souvent le terme fonction de hachage pour désigner une fonction de hachage à sens unique sans collision.

La plupart des fonctions de hachage à sens unique sans collision sont construites par itération d'une fonction de compression : le message *M* est décomposé en *n* blocs $m_1,...,m_n$ puis une fonction de compression est appliquée à chaque bloc et au résultat de la compression du bloc précédent ; l'empreinte *h(M)* est le résultat de la dernière compression.

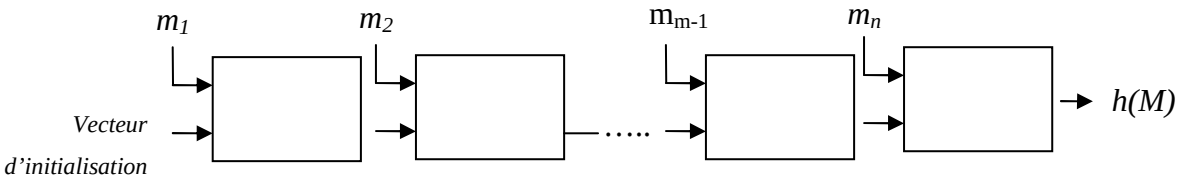


Figure 1.13: *Fonction de hachage itérative*

1.9.1.2 Fonction MD5

MD5 est une fonction de hachage cryptographique qui produit des empreintes de 128 bits. Il s'agit de la fonction de hachage utilisée pour constituer l'empreinte de la signature de PGP.

MD5 fonctionne itérativement sur des mots de 32 bits. La fonction prend en entrée une variable de chaînage de quatre mots (Valeur initiale) ainsi que le message composé d'un bloc de seize mots et engendre une sortie de quatre mots (128 bits) définissant ainsi une novelle variable de chaînage.

On enchaine ensuite le bloc de message suivant avec le résultat du hachage du premier bloc.

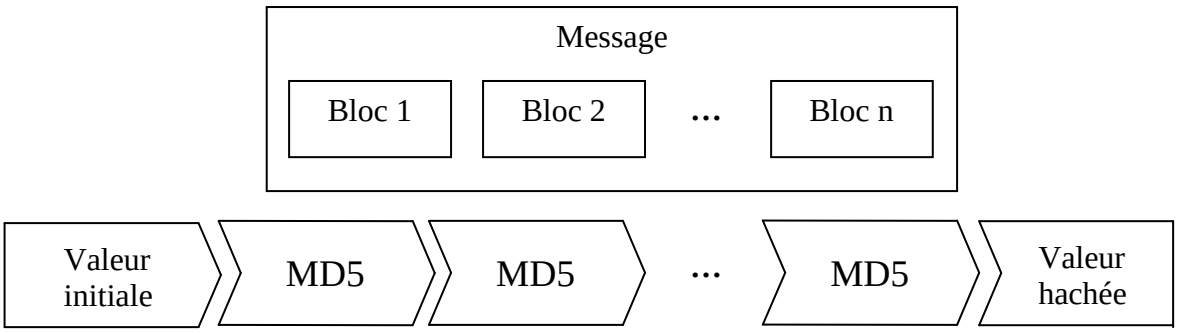


Figure 1.14: *Chainage des blocs par MD5*

Toutes les opérations utilisées par MD5 sont définies sur des mots de 32 bits. La transformation consiste en quatre étapes successives et chaque étape est constituée de seize sous-étapes. Dans chacune des sous-étapes un mot des variables de chaînage est modifié comme suit : on ajoute un mot du message et le résultat du calcul d'une fonction non linéaire dépendant des trois autres mots de la variable de chaînage ; ce résultat subit ensuite une permutation circulaire d'un certain nombre de bits.

1.9.1.3 SHA (Secure Hash Algorithm**)**

SHA [3] est la fonction de hachage utilisée par SHS *(Secure Hash Standard)*, la norme du gouvernement Américain pour le hachage. SHA-1 est une amélioration de SHA publiée en 1994. SHA-1 produit une empreinte de 160 bits à partir d'un message de longueur maximale 2^{64} bits.

SHA utilise les différentes représentations suivantes : la représentation de base est la représentation hexadécimale qui code des mots binaires de longueur 4 ; ces entiers hexadécimaux sont ensuite regroupés par mot de longueur 8 pour donner un mot de longueur 32 bits qui représente le codage binaire d'un entier compris entre 0 et 2^{32}-1.

Le codage d'un entier $0 \leq z \leq 2^{64}$ est tel que $z = 2^{32}x + y$ pour deux mots de longueur 8, $x\ et\ y$. Enfin SHA travaille sur des blocs de 512 bits représentés comme 16 mots de 32 bits, donc effectue une opération de « complément » du message initial.

1.9.1.4 RIPE-MD

Développée dans le cadre du projet **RIPE** *(RACE Integrity Primitives Evaluation)* de la communauté Européenne, RIPE-MD fournit une empreinte de 128 bits. RIPE-MD-160 est une version renforcée de RIPE-MD qui fournit une empreinte de 160 bits.

1.9.2 Intégrité et authentification de l'origine des données

Parmi les problèmes auxquels s'attaque la cryptographie, on trouve l'authentification de l'origine des données et l'intégrité [38], [39]: lorsque l'on communique avec une autre personne au travers d'un canal peu sûr, on aimerait que le destinataire puisse s'assurer que le message émane bien de l'auteur auquel il est attribué et qu'il n'a pas été altéré pendant le transfert.

Les fonctions de hachage à sens unique interviennent dans la résolution de ces problèmes. Si l'on dispose d'un canal sûr (mais plus coûteux) en parallèle du canal de communication normal, on peut communiquer l'empreinte des messages par l'intermédiaire de ce canal. On assure ainsi l'intégrité des données transférées. Sans canal sûr, le problème se complique : si l'on transfère l'empreinte sur un canal de communication non sûr, un intercepteur peut modifier les données puis recalculer l'empreinte. Il convient donc de trouver une méthode pour s'assurer que seul l'expéditeur est capable de calculer l'empreinte. Pour cela, on peut utiliser, par exemple, une fonction de hachage à sens unique qui fonctionne de plus avec une clé secrète ou privée. On remarquera qu'on fournit également l'authentification de l'origine des données.

Inversement, si on désire fournir l'authentification de l'origine des données et que l'on utilise pour cela un moyen qui ne garantit pas l'intégrité des données authentifiées, un intrus peut modifier le message et donc faire accepter comme authentifiées des données qu'il a choisies. C'est pourquoi intégrité et authentification de l'origine des données sont généralement fournies conjointement par un même mécanisme. On utilise parfois le terme *d'authenticité* pour désigner l'intégrité jointe à l'authentification des données.

1.9.3 Signature numérique

La norme ISO 7498-2 [37] définit la signature numérique comme des "données ajoutées à une unité de données, ou transformation cryptographique d'une unité de données, permettant à un destinataire de prouver la source et l'intégrité de l'unité de données et protégeant contre la contrefaçon (par le destinataire, par exemple).

La mention "protégeant contre la contrefaçon" implique que seul l'expéditeur doit être capable de générer la signature. Les ouvrages [37], [39] montrent qu'une signature numérique fournit les services d'authentification de l'origine des données, d'intégrité des données et de non-répudiation.

Ce dernier point la différencie des *codes d'authentification de message*, et a pour conséquence que la plupart des algorithmes de signature utilisent la cryptographie à clé publique.

Sur le plan conceptuel, la façon la plus simple de signer un message consiste à chiffrer celui-ci à l'aide d'une clé privée : seul le possesseur de cette clé est capable de générer la signature, mais toute personne ayant accès à la clé publique correspondante peut la vérifier. Dans la pratique, cette méthode est peu utilisée du fait de sa lenteur.

Une autre méthode utilisée pour signer consiste à calculer une empreinte du message à signer et à ne chiffrer que cette empreinte. Le calcul d'une empreinte par application d'une fonction de hachage étant rapide et la quantité de données à chiffrer étant fortement réduite, cette solution est bien plus rapide que la précédente.

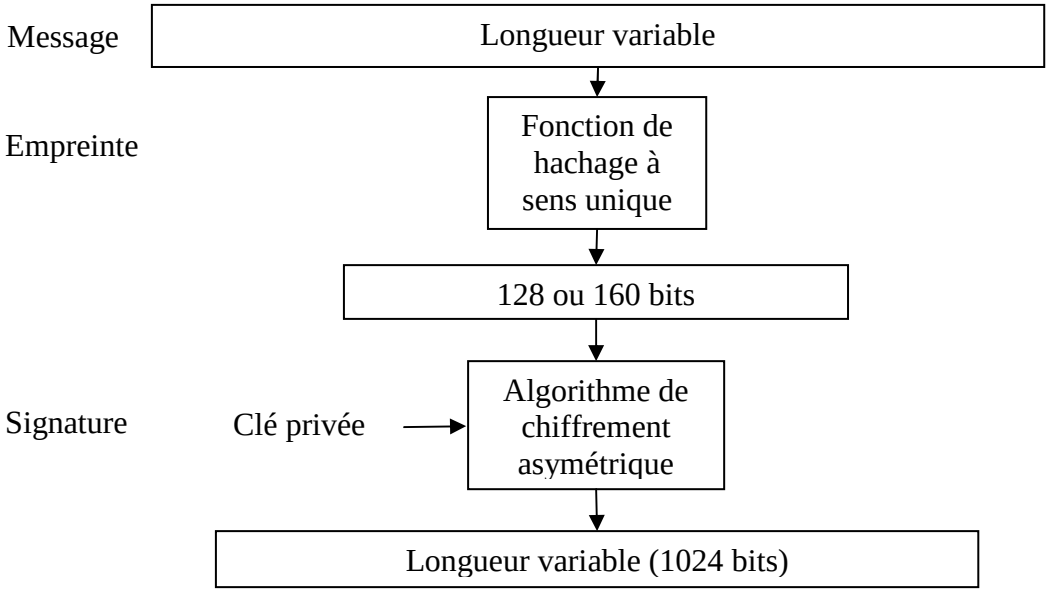


Figure 1.15: *Signature*

1.9.3.1 Mécanisme général de signature

Un procédé de signature est composé :

- d'un algorithme privé de signature noté sig qui, à un texte clair M et pour une clé fixée K, retourne une signature S.

$$sig_K(M) = S$$

- d'un prédicat public de vérification noté ver qui, à une clé fixée K et pour tout couple clair/signature (M, S), va vérifier la validité de la signature S pour le message clair M.

$$ver_K(M, S) = vrai \Leftrightarrow S = sig_K(M)$$

1.9.3.2 Signature par RSA

Bob désire envoyer un message M signé à Alice. Ils disposent pour cela de leurs systèmes RSA respectifs :

	Privée	Publiques
Alice	d_A	n_A, e_A
Bob	d_B	n_B, e_B

Le procédé de signature est alors :

$$sig_K(M) = M^{d_B}(mod\ n_B) = S \tag{1.40}$$

Celui de vérification

$$ver_K(M, S) = vrai \Leftrightarrow S^{e_B}(mod\ n_B) = M \tag{1.41}$$

1.9.3.3 Signature par El Gamal

Soit p un nombre premier pour lequel le problème du logarithme discret est difficile dans $\mathbb{Z}_p^\times$ et soit α une racine primitive de $\mathbb{Z}_p^\times$. Le message $M \in \mathbb{Z}_p^\times$ et sa signature est un couple $(\gamma, \delta) \in \mathbb{Z}_p^\times \times \mathbb{Z}_{p-1}$. L'ensemble des clés est $K = \{(p, \alpha, a, \beta) : \beta = \alpha^a (mod\ p)\}$

Privée	Publiques
a	p, α, β

On choisit $k \in \mathbb{Z}_{p-1}^\times$ aléatoire et secret qui vérifie $pgcd(k, p-1) = 1$ et on définit une signature comme :

$$sig_K(M, k) = (\gamma, \delta)$$

Avec

$$\gamma = \alpha^k (mod\ p) \quad \delta = (M - a\gamma)k^{-1} (mod\ (p-1))$$

Pour $M, \gamma \in \mathbb{Z}_p^\times\ et\ \delta \in \mathbb{Z}_{p-1}$, on définit le prédicat de vérification par :

$$ver_K(M, \gamma, \delta) = vrai \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^M (mod\ p)$$

Si la signature est construite correctement, la vérification authentifie la signature car :

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{a\gamma}\alpha^{k\delta}(mod\ p)$$
$$\equiv \alpha^{M}(mod\ p)$$

On utilise le théorème de Fermat-Euler et le fait que :

$$a\gamma + k\delta \equiv M\big(mod(p-1)\big) car\ \delta = (M - a\gamma)k^{-1}(mod(p-1))$$

Bob calcule la signature en utilisant sa clé privée a et la valeur aléatoire k (utilisée une seule fois pour la signature de M). La vérification s'effectue à l'aide de la clé publique.

1.9.3.4 Digital Signature Standard (DSS)

DSS est une variante du procédé d'El Gamal qui date de 1994. Aux Etats Unis, DSS est aux signatures ce que DES (ou AES) est au chiffrement.

Soit p un nombre premier de 512 bits, q un facteur premier de 160 bits de $p-1$ et soit α une racine q^e primitive de l'unité $modulo\ p$ tel que le problème du logarithme discret dans le sous-groupe engendré par α est difficile. Le message $M \in \mathbb{Z}_p^{\times}$ et sa signature est constituée du couple $(e_1, e_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$. L'ensemble des clés est :

$$K = \{(p, q, \alpha, a, \beta : \beta = \alpha^a (mod\ p)\}$$

Privée	Publiques
a	p, q, α, β

On choisit $1 \leq k \leq q-1$ aléatoire et on définit une signature comme :

$$sig_K(M, k) = (\gamma, \delta)$$

Pour

$$\gamma = \alpha^k (mod\ p)(mod\ q) \quad \delta = (M - a\gamma)k^{-1}(mod\ q)$$

Pour $M \in \mathbb{Z}_p^{\times}$ et $\gamma, \delta \in \mathbb{Z}_q$, nous avons :

$$e_1 = M\delta^{-1}(mod\ q)$$
$$e_2 = \gamma^{\delta^{-1}}(mod\ q)$$

Le prédicat public de vérification devient :

$$ver_K(M,\gamma,\delta) = vrai \Leftrightarrow \left(\alpha^{e_1}\beta^{e_2}(mod\ p)\right)(mod\ q) = \gamma = \left(\alpha^k\ (mod\ p)\right)(mod\ p)$$

Notons que $\delta \not\equiv 0(mod\ q) car\ \delta^{-1}(mod\ q)$ est nécessaire à la vérification des signatures. Si Bob obtient $\delta \equiv 0(mod\ q)$ dans le procédé de signature, il doit la rejeter et construire une nouvelle signature avec une nouvelle valeur de k.

Pour justifier le bon fonctionnement du procédé de vérification, il est nécessaire d'introduire le lemme suivant.

Lemme 1.01 :

Si p est un entier premier, q un facteur premier de $p-1$, g un entier non nul plus petit que $p\ et\ \alpha = g^{(p-1)/q}(mod\ p)$, alors $\alpha^q(mod\ p) = 1$. De plus, si

$$m(mod\ q) = n(mod\ q)\ alors\ \ \alpha^m(mod\ p) = \alpha^n(mod\ p)$$

Démonstration :

♣ $$\alpha^q(mod\ p) = (g^{(p-1)/q}(mod\ p))^q(mod\ p) = g^{p-1}(mod\ p) = 1$$

Si $m \equiv n\ (mod\ q)$, pour un entier λ, $m = n + \lambda q$; alors, comme

$$\alpha^q mod\ p = 1, \alpha^m \equiv \alpha^n\alpha^{\lambda q} \equiv \alpha^n(mod\ p)$$ ♦

Nous pouvons maintenant justifier le bon fonctionnement de la vérification.

Théorème 1.16 :

Pour un triplet $M,\gamma,\delta)$ valide, on a bien:

$$\left(\alpha^{e_1}\beta^{e_2}(mod\ p)\right)\ (mod\ q) = \gamma = \left(\alpha^k(mod\ p)\right)(mod\ q)$$

Pour $e_1 = M\delta^{-1}(mod\ q)\ et\ e_2 = \gamma\delta^{-1}(mod\ q)$

Démonstration :

♣ Puisque $k\delta = M + a\gamma(mod\ q)$ et en utilisant le lemme 1.1, on déduit que

$$\left(\alpha^{(M+a\gamma)\delta^{-1}}(mod\ p)\right)(mod\ q) = \left(\alpha^k(mod\ p)\right)(mod\ q)$$ ♦

1.9.4 Scellement

Tout comme la signature numérique, le scellement fournit les services d'authentification de l'origine des données et d'intégrité des données**,** mais il ne fournit pas le non répudiation. Ceci permet l'utilisation de la cryptographie à clé secrète pour la génération du sceau ou *code d'authentification de message.*

Un code d'authentification de message *(*MAC : *Message Authentication Code*) est le résultat d'une fonction de hachage à sens unique dépendant d'une clé secrète : l'empreinte dépend à la fois de l'entrée et de la clé. On peut construire un MAC à partir d'une fonction de hachage ou d'un algorithme de chiffrement par blocs. Il existe aussi des fonctions spécialement conçues pour faire un MAC. Une façon courante de générer un code d'authentification de message consiste à appliquer un algorithme de chiffrement symétrique en mode CBC au message ; le MAC est le dernier bloc du cryptogramme.

Un moyen simple de transformer une fonction de hachage à sens unique en un MAC consiste à chiffrer l'empreinte avec un algorithme à clé secrète. Une autre méthode consiste à appliquer la fonction de hachage non pas simplement aux données à protéger, mais à un ensemble dépendant à la fois des données et d'un secret.

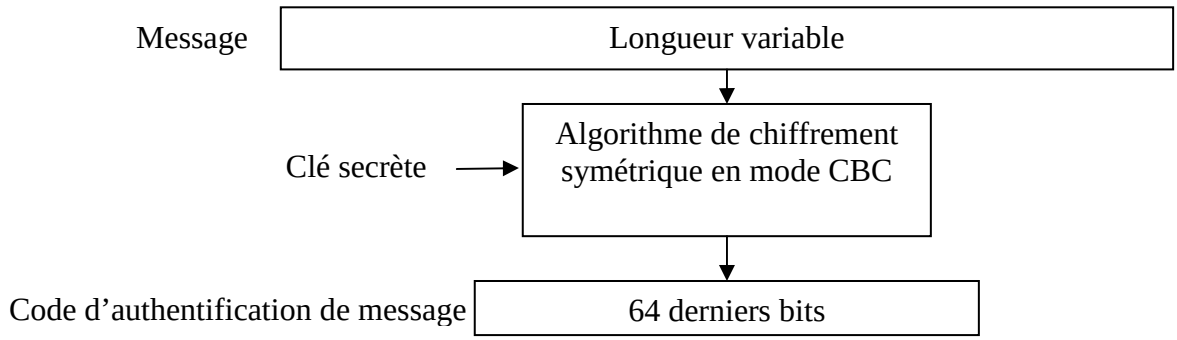


Figure 1.16: *Scellement à l'aide d'un algorithme de chiffrement symétrique*

1.9.4.1 Keyed-Hash

Un exemple simple de cette façon de procéder est de prendre pour MAC des valeurs du type H(secret, message), H(message, secret) ou H(secret, message, secret). Ces méthodes, présentées en 1992 par Gène Tsudik [38] s'appellent respectivement méthode du préfixe secret, du suffixe secret et de l'enveloppe secrète. Elles ont une sécurité limitée.

1.9.4.2 HMAC

Une méthode de calcul de MAC à base de fonction de hachage plus élaborée et plus sûre est HMAC. La méthode HMAC peut être utilisée avec n'importe quelle fonction de hachage itérative telle que MD5, SHA-1 ou encore RIPE-MD.

Soit H une telle fonction, K le secret et M le message à protéger. H travaille sur des blocs de longueur *b* octets (64 en général) et génère une empreinte de longueur l octets (16 pour MD5, 20 pour SHA et RIPE-MD-160). Il est conseillé d'utiliser un secret de taille au moins égale à l octets.

On décrit deux chaînes, *ipad (inner pudding data)* et *opad (outer pudding data)*, de la façon suivante :

$$ipad = \ l'octet\ 0 \times 36\ répété\ b\ fois,$$
$$opad\ = \ l'octet\ 0 \times 5C\ répété\ b\ fois.$$

Le MAC se calcule alors suivant la formule suivante :

$$HMAC_K(M)\ = \ H\ (\ K\ opad, H(K\ \ ipad, M)\ ).$$

Une pratique courante avec les fonctions de calcul de MAC est de tronquer la sortie pour ne garder comme MAC qu'un nombre réduit de bits. Avec HMAC on peut ainsi choisir de ne retenir que les *t* bits de gauche, où *t* doit être supérieur à $l/2$ et 80. On désigne alors sous la forme HMAC-H-t l'utilisation de HMAC avec la fonction H, tronque à *t* bits (par exemple, HMAC-SHA1-96).

1.10 Protocoles cryptographiques

Tout comme les protocoles de communication, les protocoles cryptographiques sont une série d'étapes prédéfinies [3], [12], basées sur un langage commun, qui permettent à plusieurs participants (généralement deux) d'accomplir une tâche. Dans le cas des protocoles cryptographiques, les tâches en question sont bien sûr liées à la cryptographie : ce peut être une authentification, un échange de clé,....

Une particularité des protocoles cryptographiques est que les tiers en présence ne se font généralement pas confiance et que le protocole a donc pour but d'empêcher l'espionnage et la tricherie.

1.10.1 Protocoles à apport nul de connaissance

Une situation fréquente est celle où un tiers doit prouver à un autre la connaissance d'un secret sans rien révéler sur ce secret [4], [16]**.** Les protocoles qui permettent de répondre à ce problème sont appelés protocoles à apport nul de connaissance ou preuves à divulgation nulle *(zero-knowledge).*

Les protocoles à apport nul de connaissance prennent la forme de protocoles interactifs au cours desquels Bob, le vérificateur, pose à Alice, une série de questions. Si Alice connaît le secret, elle saura répondre correctement à toutes les questions ; sinon elle aura 50% de chances de répondre correctement à chaque question. La probabilité qu'Alice connaisse effectivement le secret après *n* réponses correctes est donc de $1 - \frac{1}{2^n}$. Après 10 questions, cette probabilité sera donc de 0,999.

Soit *n* un module aléatoire, produit de deux grands nombres premiers. La clé publique d'Alice est v, un résidu quadratique modulo *n* (i.e. $x^2 = \boldsymbol{v}\ mod\ n$ admet une solution et $v^{-1} mod\ n$ existe). La clé privée correspondante est le plus petit s tel que $s \equiv \sqrt{v^{-1}}\ mod\ n$

♣ Le déroulement d'une ronde du protocole est le suivant :

1) Alice choisit un nombre aléatoire $r \leq n$, puis calcule $x = \boldsymbol{r^2}\ mod\ n$ et envoie *x* à Bob.

2) Bob envoie un bit aléatoire *b = 0* ou 1 à Alice.

3) Alice envoie $y = r\ s^b\ mod\ n$ à Bob

   - Si $b = 0$, $y = r$, donc Bob vérifie que $y^2 = x\ mod\ n$; cela prouve qu'Alice connaît $r = \sqrt{x}$
   - Si $b = 1$, $y = r\ s$, donc Bob vérifie que $y^2\ v \equiv x\ mod\ n$ cela prouve qu'Alice connaît *s.*

Si Alice ne connaît pas *s*, pour tromper Bob lorsque $b = 1$, elle doit lui envoyer $x = v^{-1}$; mais cela implique que *r* = s, donc Alice ne saura pas répondre si *b = 0.* ♦

Inversement, si Alice envoie effectivement une valeur x générée par $x = r^2 \bmod n$, elle saura répondre lorsque *b = 0 (y = r)*, mais elle ne pourra pas répondre lorsque *b = 1*, car il lui faudrait connaître *s*. Dans chaque cas, Alice a donc une chance sur deux de savoir répondre à la question.
Le principal problème des protocoles à apport nul de connaissance est le nombre important d'échanges nécessaires, qui les rend peu utilisés en pratique.

1.10.2 La notion de tiers de confiance

Beaucoup de protocoles cryptographiques, notamment ceux visant à sécuriser des environnements distribués, ont recours à la notion de *tiers de confiance* [39].
Dans un tel cadre, pour établir une communication sécurisée, Alice et Bob se font aider de Norbert le notaire, qui est une personne en qui ils ont confiance**.** Norbert va jouer un rôle dans la sécurisation des échanges entre Alice et Bob en participant à la mise en œuvre de mécanismes de sécurité, notamment en intervenant dans les protocoles d'authentification et d'échange de clé.

1.10.3 Echange de clé

Les deux méthodes les plus courantes d'établissement de clé sont le transport de clé et la génération de clé. Un exemple de transport de clé est l'utilisation d'un algorithme à clé publique pour chiffrer une clé de session générée aléatoirement. Un exemple de génération de clé est le protocole Diffie-Hellman [39] qui permet de générer un secret partagé à partir d'informations publiques. Le seul protocole décrit ici est le principe de génération de clé de Diffie-Hellman et les façons de l'étendre pour contrer les attaques de l'intercepteur. Diffie-Hellman est à la base de nombreux protocoles d'échange de clé.

1.10.3.1 Diffie-Hellman

Inventé en 1976 par Diffie et Hellman [39], ce protocole permet à deux tiers de générer un secret partagé sans avoir aucune information préalable l'un sur l'autre**.** Il est basé sur la cryptologie à clé publique, car il fait intervenir des valeurs publiques et des valeurs privées. Sa sécurité dépend de la difficulté de calculer des logarithmes discrets sur un corps fini. Le secret généré à l'aide de cet algorithme peut ensuite être utilisé pour dériver une ou plusieurs clés (clé secrète,...).

Voici le déroulement de l'algorithme :

1) Alice et Bob se mettent d'accord sur un grand entier *n* tel que *(n-1)/2* soit premier et sur un entier *g* primitif par rapport à *n.* Ces deux entiers sont publics.

2) Alice choisit de manière aléatoire un grand nombre entier *a,* qu'elle garde secret, et calcule sa valeur publique, $A = g^a mod\ n$. Bob fait de même et génère b et $B = g^b\ mod\ n$.

3) Alice envoie *A* à Bob ; Bob envoie *B* à Alice.

4) Alice calcule $K_{AB} = B^a\ mod\ n$; Bob calcule $K_{BA} = A^b mod\ n$.

   $K_{AB} = K_{BA} = g^{ab}\ mod\ n$ est le secret partagé par Alice et Bob.

Une personne qui écoute la communication connaît *g, n, A* et *B,* ce qui ne lui permet pas de calculer $g^{ab}\ mod\ n$ *:* il lui faudrait pour cela calculer l'algorithme de *A* ou *B* pour retrouver *a* ou *b.*

En revanche, Diffie-Hellman est vulnérable à l'attaque active dite attaque de l'intercepteur.

Une façon de contourner le problème de l'attaque de l'intercepteur sur Diffie-Hellman est d'authentifier les valeurs publiques utilisées pour la génération du secret partagé. On parle alors de Diffie-Hellman authentifié**.**

L'authentification peut se faire à deux niveaux :

- En utilisant des valeurs publiques authentifiées, à l'aide de certificats par exemple. Cette méthode est notamment à la base du protocole SKIP.

- En authentifiant les valeurs publiques après les avoir échangées, en les signant par exemple. Cette méthode est utilisée entre autres par le protocole Photuris.

L'inconvénient, dans les deux cas, est que l'on perd un gros avantage de Diffie-Hellman, qui est la possibilité de générer un secret partagé sans aucune information préalable sur l'interlocuteur.

1.10.3.2 Echange de clé et authentification mutuelle

Pour établir une communication sécurisée, on procède généralement, en premier lieu, à une authentification à des fins de contrôle d'accès.

Puis, un échange de clé permet l'utilisation d'un mécanisme de sécurisation des échanges : l'authentification est ainsi étendue à la suite de la communication. L'exemple de Diffie-Hellman montre comme il est important que l'échange de clé soit authentifié. On parle alors de protocole d'authentification mutuelle avec échange de clé.

1.10.3.3 Propriétés des protocoles d'échange de clé

Diffie, Van Oorschot et Wiener [39] expliquent, la notion de protocole d'authentification mutuelle avec échange de clé sûr.

Un protocole est dit sûr si les deux conditions ci-après sont valables dans chaque instance du protocole où l'un des deux tiers, par exemple Alice, exécute le protocole honnêtement et accepte l'identité de l'autre tiers.

Les conditions sont :

- Au moment où Alice accepte l'identité de Bob, les enregistrements des messages échangés par les deux tiers se correspondent (i.e. les messages n'ont pas été altérés en route).

- Il est matériellement impossible pour toute personne autre que les tiers en présence de retrouver la clé échangée.

La propriété évoquée ci-dessus représente le minimum requis pour tout protocole.

Cependant, d'autres propriétés des protocoles d'authentification avec échange de clé peuvent être souhaitables :

- La propriété dite de PFS (Perfect Forward Secrecy) est garantie si la découverte par un adversaire du ou des secrets à long terme ne compromet pas les clés de session générées précédemment : les clés de session passées ne pourront pas être retrouvées à partir des secrets à long terme.

- On considère généralement que cette propriété assure également que la découverte d'une clé de session ne compromet ni les secrets à long terme ni les autres clés de session.

- A ce sujet, on parle d'attaque de Denning-Sacco lorsqu'un attaquant récupère une clé de session et utilise celle-ci, soit pour se faire passer pour un utilisateur légitime, soit pour rechercher les secrets à long terme (par attaque exhaustive ou attaque par dictionnaire).

- La propriété dite de BTP (Back Traffic Protection) est fournie si la génération de chaque clé de session se fait de manière indépendante : les nouvelles clés ne dépendent pas des clés précédentes et la découverte d'une clé de session ne permet ni de retrouver les clés de session passées ni d'en déduire les clés à venir.

- On dit qu'il y a authentification directe (Direct Authentication) si, à la fin du protocole, les valeurs servant à générer le secret partagé sont authentifiées ou si chaque tiers a prouvé qu'il connaissait la clé de session.

- Par opposition, l'authentification est dite indirecte (Indirect authentication) si elle n'est pas garantie à la fin du protocole, mais dépend de la capacité de chaque tiers à utiliser, dans la suite des échanges, la ou les clés mises en place précédemment.

- On parle de protection de l'identité (Idenlity Protection) lorsque le protocole garantit qu'un attaquant espionnant les échanges ne pourra pas connaître les identités des tiers communiquant.

- Enfin, l'utilisation du temps (Timestamps) afin d'éviter la rejouabilité est très controversée du fait, principalement, de sa trop grande dépendance d'horloges synchronisées.

1.10.3.4 Hiérarchie des clés

On distingue plusieurs types de clés suivant leurs rôles :

- Clé de chiffrement de clés : situées en haut de la hiérarchie, ces clés servent exclusivement à chiffrer d'autres clés et ont généralement une durée de vie longue. On notera que leur utilisation, restreinte au chiffrement de valeurs aléatoires que sont des clés,

rend les attaques par cryptanalyse plus difficiles à leur niveau. La cryptographie à clé publique est souvent utilisée pour le transport de clés en chiffrant la clé à transporter à l'aide d'une clé publique.

- Clés maîtresses : les clés maîtresses sont des clés qui ne servent pas à chiffrer mais uniquement à générer d'autres clés par dérivation. Une clé maîtresse peut ainsi être utilisée, par exemple, pour générer deux clés : une pour le chiffrement et une pour la signature.

- Clés de session ou de chiffrement de données. D'une durée de vie généralement faible (elle peut aller jusqu'à changer à chaque message), une telle clé, par opposition aux précédentes, sert à chiffrer des données et se situe donc au bas de la hiérarchie. Du fait de leur lenteur, les algorithmes asymétriques sont très peu utilisés en chiffrement de données, et les clés de session sont donc généralement des clés secrètes.

1.11 Conclusion

Dans ce premier chapitre, une introduction sur les méthodes, les mathématiques et les principes de la cryptographie ont été donnés. Comme la cryptographie est l’art de protéger l’information à l’aide de différents procédés et des théories mathématiques, il est primordial de rappeler ses fonctions de base utilisées.

Les différents protocoles et algorithmes ont été définis succinctement, ainsi ils constituent le fondement, la spécificité et la complexité d’un cryptosystème. Les fonctions et protocoles responsables de l’intégrité et la confidentialité des messages ont été aussi évoqués.

CHAPITRE 2
TRAITEMENTS MULTIMEDIAS

2.1 Définitions

Tout d'abord, nous précisons quel sera le champ couvert par le terme « multimédia ». Quelques définitions permettent de cerner ce champ [46].

Définition 2.01 :

- On définit par « multimédia » tout ce qui « associe plusieurs modes de représentation des informations telles que texte, son, image, vidéo »

Définition 2.02 :

- Le « multimédia » est une « technique de communication associant sur un seul support des données audiovisuelles et informatiques permettant une utilisation interactive tout en visant les métiers eux-mêmes développés pour la création et la diffusion des produits multimédias ».

Définition 2.03 :

- Le « multimédia informatique » est une « Technologie de l'information permettant la manipulation simultanée de sons, d'images et de textes, au moyen d'un seul ensemble informatique et de façon interactive ».

Remarque : un média est un moyen physique par lequel les données sont perçues, représentées, stockées ou transmises, ce qui nous amène à distinguer les divers cas présentés dans le tableau ci-dessous :

Stockées ou transmises	Représentées	Perçues	
Systèmes de stockage ou de transmission	Objets d'information	Systèmes d'entrée/sortie	Usagers
Média d'échange (média de stockage ou de transmission)	Média de représentation	Média de présentation	Média de perception
Disquette, CD-ROM, Bande magnétique, Vidéocassette, RNIS, Satellite, Câble, Faisceau hertzien	Caractères, Graphiques, Photographies, Vidéos	Ecran, Moniteur, Papier, Haut-parleur…	Clavier, Souris, Microphone, Caméra…

Tableau 2.01 : *Les différents médias*

2.2 Caractérisation du domaine

2.2.1 Médias discrets / Médias continus

On distingue deux types de médias [46]:

Définition 1.04 :

- Ceux qui n'ont aucune dépendance temporelle, comme le texte, les graphiques, les images fixes, qui seront appelés médias discrets; l'information contenue est donnée exclusivement par l'ensemble des éléments individuels qui le compose, sans aucune référence temporelle. Leur traitement, même s'il doit se faire dans un temps contraint par l'application, n'est pas conditionné par des contraintes temporelles, l'intégrité des données d'un média discret étant indépendante du temps. Leur volume est aussi limité.

Définition 1.05 :

- Ceux qui ont une dépendance temporelle, comme le son, la musique, la vidéo, qui seront appelés médias continus, le terme continu se référant à la perception de l'usager, pas à la représentation interne. L'information qu'ils contiennent ne peut se comprendre que par rapport aux évolutions temporelles de grandeurs physiques. Leur traitement est critique du point de vue temporel. Par exemple, un groupe d'échantillons audio n'ont pas de sens s'ils sont lus après ceux qui les suivaient initialement. Leur volume n'est limité que par la durée, et leur débit est en général élevé.

2.2.2 Pluridisciplinarité

Le multimédia est un domaine pluridisciplinaire si on veut traiter de manière exhaustive ses différents aspects [47], [48]. La figure 2.01 ci-après montre de manière structurée les principales composantes d'un système multimédia.

On peut y distinguer quatre sous-ensembles principaux qui pour chacun d'entre eux recouvrent des concepts et des techniques très variés :

- systèmes de stockage, de compression et interfaces avec les réseaux. Ceci regroupe des composants matériels et logiciels de plus en plus standardisés et qui sont la concrétisation de recherches de base en optique, optoélectronique, traitement du signal et des images, conception et spécification de protocoles, ...

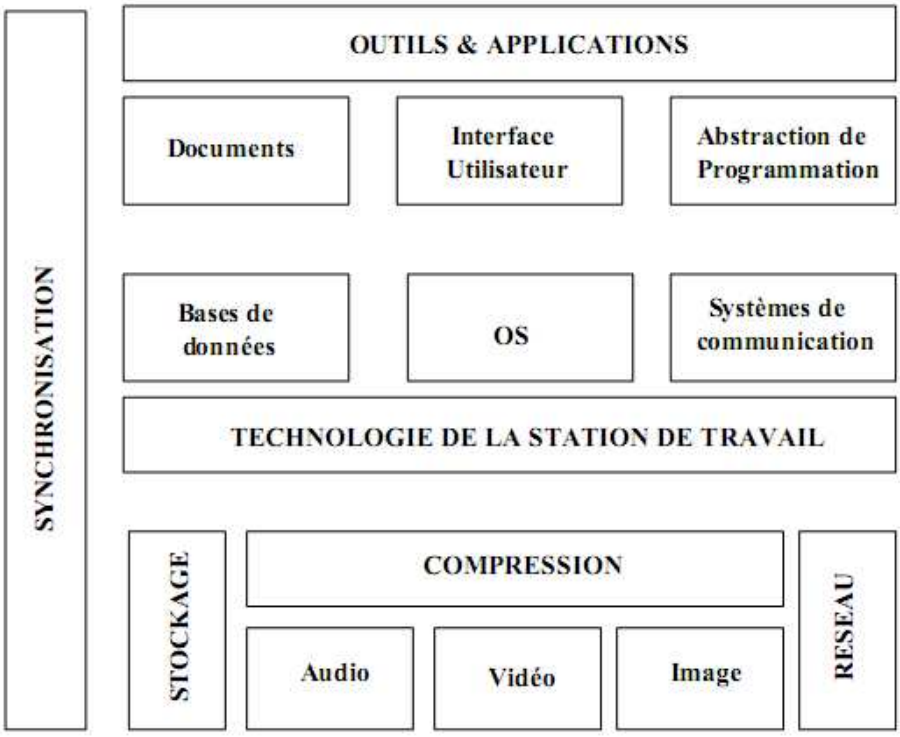


Figure 2.01: *Le multimédia, domaine pluridisciplinaire.*

- systèmes informatiques (systèmes d'exploitation, bases de données, systèmes de communication). Les fortes contraintes temps réel des applications multimédias doivent être prises en compte dans la conception des systèmes d'exploitation, en particulier les techniques d'ordonnancement de tâches. Les bases de données multimédias constituent un domaine de recherche aujourd'hui très actif. Enfin, le système de communication doit prendre en charge la transmission de données en tenant compte des contraintes temps réel et de fiabilité d'applications multimédias communicantes.

- services et outils génériques pour la conception d'applications multimédias. Les concepts mis en œuvre sont la conception orientée objets, les abstractions de programmation et la synchronisation spatiale et temporelle dans une architecture de documents multimédias.

- mécanismes de synchronisation. Tous les composants du système décrits ci-dessus concourent à maintenir des relations temporelles entre différents médias. Le respect de celles-ci est la garantie d'une qualité de service offerte aux utilisateurs.

2.3 Les techniques multimédia

2.3.1 Compression

Les techniques de compression jouent un rôle crucial dans les applications multimédias. Les signaux audios, les images, la vidéo correspondent à des quantités d'information importantes, à prendre en compte aussi bien du point de vue du stockage que de celui de la transmission. Les systèmes multimédias ont besoin de la compression essentiellement pour trois raisons [49], [50], [51]:

- les données multimédias nécessitent des capacités de stockage impes supports de stockage actuels, relativement lents, ne peuvent pas rejouer des données multimédias, surtout la vidéo, en temps réel ;

- la bande passante des réseaux de transmission ne permet pas (en général) la transmission de vidéo en temps réel.

Par exemple, une image de vidéo couleur 24 bits à la définition de 620 x 560 pixels correspond à environ 1 Mo. A la cadence de 30 images par seconde, une seconde de vidéo correspond à 30 Mo. Une application multimédia typique peut faire appel à 30 minutes de vidéo, 2.000 images et 40 minutes de son stéréo. Cette application nécessite donc 50 Go de stockage pour la vidéo, 15 Go pour les images et 0,4 Go pour la partie audio, pour un total de 65,4 Go.

Les techniques actuelles de compression opèrent, à des qualités variables, à des taux de compression de 10 à 50 pour des images fixes, et à des taux pouvant atteindre 2.000 pour la vidéo. Il faudrait ainsi un débit d'entrée/sortie de 30 Mbps, alors que, par exemple, la technologie CD-ROM actuelle est limitée à des transferts à environ 300 Kbps. L'unique solution est donc de comprimer les données avant stockage, et de les décomprimer à la relecture.

Définition 1.06 :

Dans ces différentes techniques, on peut distinguer deux méthodes de compression : les approches sans perte de celle avec perte. Les méthodes sans pertes (ou encore réversibles) peuvent reconstruire la représentation originale d'un signal exactement.

A l'inverse, les méthodes avec perte (irréversibles) ne reconstruisent le signal qu'approximativement, le niveau de dégradation étant proportionnel au taux de compression, qui est, dans ce cas, plus important.

Les méthodes avec perte, les plus souvent employées en audiovisuel, se divisent en techniques prédictives et fréquentielles. Dans la première approche, on prédit les valeurs futures d'après l'observation des valeurs passées. Les approches fréquentielles utilisent une représentation du signal dans un espace transformé, comme la Transformée en Cosinus Discret (TCD) liée à la Transformée de Fourier.

Les techniques de compression hybrides combinent ces plusieurs approches. Des normes ont été précisées ces dernières années à partir desquelles se développent les applications multimédias. Le tableau 2.02 en donne une liste sommaire [50].

Sigle	Intitulé	Groupe de normalisation	Taux de compression typique
JPEG	Compression numérique et codage d'images fixes	Joint Photographic Expert Group	15 couleurs
H.261	Codeur/décodeur vidéo pour services audio-visuels	Specialist Group on Coding for Visual Telephony	100 à 2000 (Télécommunication video)
MPEG	Codage de la vidéo et de l'audio associé	Moving Picture Expert Group	200 (Application aux Vidéo/TV)

Tableau 2.02 : *Trois normes du multimédia*

2.3.2 Réseaux multimédias

De nombreuses applications comme le courrier vidéo, la vidéo conférence et les systèmes de travail coopératifs nécessitent des échanges par *« réseaux multimédias »*. Dans ces applications, les objets multimédias sont stockés sur un serveur et rejoués sur le site du client. Ces applications peuvent nécessiter la diffusion des données multimédias vers plusieurs destinations.

Les environnements de réseaux locaux traditionnels ne peuvent supporter des accès à des sources de données multimédia distantes pour de nombreuses raisons :

- Les réseaux multimédias ont pour première caractéristique un débit (bande passante) élevé, même si les données sont comprimées. Par exemple, une session MPEG-1 requiert un débit d'environ 1,5 Mbps. MPEG-2 entre 4 et 10 Mbps. La TVHD nécessitera 20 Mbps. Les réseaux traditionnels sont utilisés pour fournir des transmissions sans perte.

Cependant, la plupart des applications multimédias peuvent tolérer des erreurs dans la transmission, sans retransmission ni correction. Dans certains cas, pour diminuer le temps de transit, ou pour la synchronisation, des paquets de données sont même éliminés délibérément. Les protocoles de transmission des réseaux multimédias ne peuvent donc pas accepter de retransmission qui pourrait introduire des délais inacceptables [47], [48].

- Les réseaux multimédias doivent fonctionner à faible latence pour assurer les interactions. Les données multimédias devant être synchronisées lorsqu'elles arrivent à leur site de destination, le réseau doit fournir une transmission synchronisée à faible gigue. Les communications multimédias sont la plupart du temps multipoints. Par exemple, les conférences entre plus de deux personnes nécessitent de distribuer l'information des divers médias à tous les participants.

- Les réseaux traditionnels répondent mal ou imparfaitement aux besoins du multimédia. Ethernet a un débit limité à 10 Mbit/s, son temps d'accès n'est pas borné, et sa latence et sa gigue ne sont pas prédictibles. Les réseaux à anneau à jeton ont un débit de 16 Mbit/s et sont déterministes. Sur ces deux critères, ils peuvent convenir au multimédia. En revanche, le temps de latence en accès est le plus important, s'il peut être prédit, on peut avoir une valeur élevée.

2.3.3 Qualité de service

La qualité de service est un ensemble de paramètres tels que [46], [51] :

- délai moyen (average delay) : temps moyen séparant l'entrée d'un paquet dans le réseau de sa sortie ;

- délai maximum (maximum delay) : temps maximum séparant l'entrée d'un paquet dans le réseau de sa sortie ;

- gigue : glissement de fréquence (jitter) : quantifie concrètement l'incertitude dans le temps d'arrivée du paquet autour de la valeur moyenne ;

- taux d'erreur binaire (bit error rate) : rapport entre le nombre de bit reçus erronés et le nombre de bit émis ;

- taux d'erreur paquets (packet error rate) : rapport entre le nombre de paquets reçus erronés et le nombre de paquets émis.

	Visiophonie	Transmission vidéo
Delai moyen	0,25 s	0,2 s
Gigue maximale	10 ms	5 ms
Taux d'erreur binaire max	0,01	0,1
Taux d'erreur paquet max	0,001	0,01

Tableau 2.03 *: Paramètres de qualité de service en visiophonie et vidéo transmission*

2.3.4 Synchronisation multimédia

Les systèmes multimédias associent divers médias discrets et continus pour créer des documents multimédias composites [46]. Parmi les divers modes de composition, on distingue :

- la composition spatiale qui permet de lier divers objets multimédias discrets en une seule entité via des règles de placement et de déformation ;

- la composition temporelle qui permet de créer une présentation multimédia par arrangement d'objets multimédias selon des relations temporelles.

Cette dernière technique de composition, encore appelée synchronisation, se décompose à son tour en :

- synchronisation continue, qui traite de la mise en synchronisme de médias continus. Par exemple la visiophonie, où les signaux audio et vidéo sont créés en un site distant, transmis sur un réseau, et synchronisés en permanence sur le site de réception.

- synchronisation ponctuelle, qui s'adresse à la synchronisation médias discrets et continus. Par exemple, lors d'une projection de diapositives, les commentaires audio doivent être synchronisés avec la projection.
- synchronisation séquentielle, qui détermine le débit auquel les événements doivent arriver à l'intérieur d'un même flot de données (synchronisation intramédia).
- synchronisation parallèle, qui détermine le séquencement relatif de deux flots de synchronisation séparés (synchronisation intermédia).

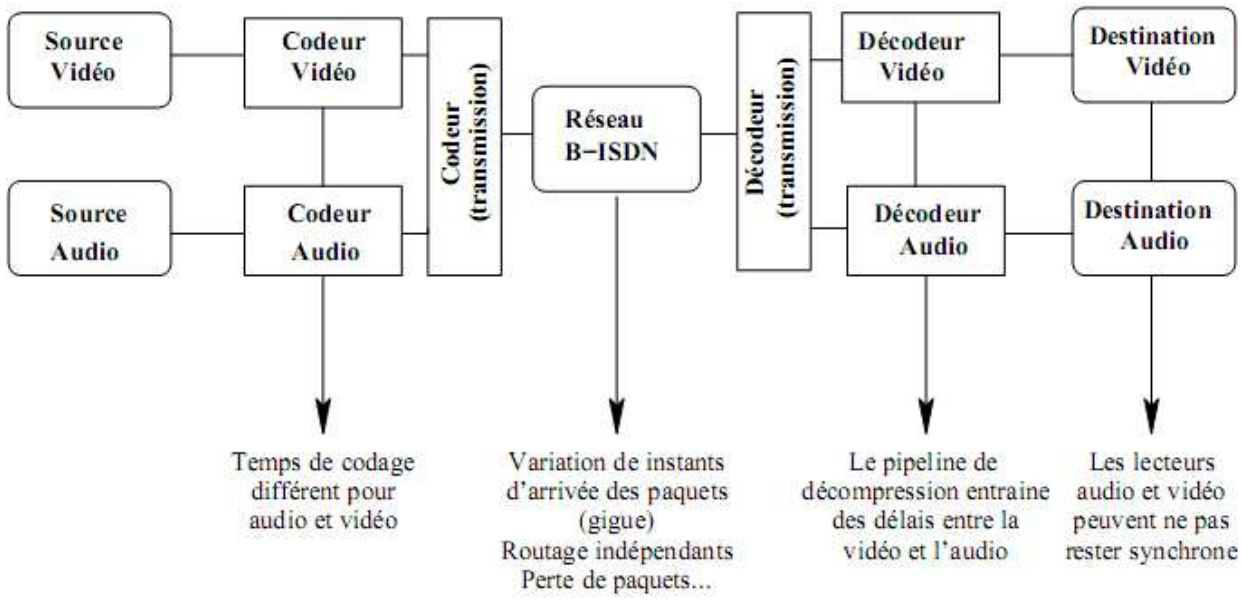


Figure 2.02: *Causes de désynchronisation dans un système de visiophonie.*

La figure 2.02 montre un exemple de système de visiophonie et les diverses causes de perte de synchronisation en divers endroits du système.

La synchronisation intermédia dépend de la configuration des sources et des destinations. On distingue habituellement quatre configurations types :

- Une source locale, par exemple un CD-ROM, distribue le flot de données aux divers systèmes de lecture. Tant que ces derniers ont la même vitesse, aucune synchronisation n'est nécessaire.
- Plusieurs sources locales. Plus d'une source distribue l'information aux systèmes de lecture. Exemple : projection de diapositives accompagnée d'une bande audio. La synchronisation est nécessaire à l'intérieur de l'ordinateur.

- Une source distribuée, comme une bande vidéo, distribue à travers un réseau à un ou plusieurs sites de lecture. Exemple : TV par câble. Outre le problème de synchronisation d'horloge, il faut maintenir la vitesse de déroulement des lecteurs constante.

- Sources multiples distribuées. Il s'agit du cas le plus complexe, où plus d'une source distribue à plusieurs systèmes de lecture répartis sur des nœuds multiples. Cas particuliers : multiple sources sur le même nœud, distribué sur un autre nœud (exemple la visiophonie) ; multiple sources depuis un ou plusieurs nœuds vers un autre nœud ; multiple sources depuis un nœud vers un ou plusieurs nœuds (exemple la TVHD) : multiple sources depuis un ou plusieurs nœuds vers un ou plusieurs nœuds (ex. visioconférence sur plusieurs sites).

2.3.5 Systèmes multimédias

La technologie actuelle rend les systèmes multimédias techniquement et économiquement accessibles. Les avancées techniques récentes les plus importantes concernent la puissance des stations de travail, la capacité des systèmes de stockage, le débit des réseaux, les méthodes de traitement des images et de la vidéo, les méthodes de traitement de l'audio, le traitement de la parole et les algorithmes de compression de la parole, de l'audio, de l'image et de la vidéo [50].

2.3.6 Applications et Services

On distingue :

- deux classes d'applications, selon qu'elles sont locales ou distribuée.
    - Les applications locales sont celles développées en général autour d'un micro-ordinateur avec lecteur de CD-ROM. Elles sont caractérisées par la présence d'une horloge locale.
    - Les applications distribuées communiquent à travers des réseaux et imposent une maîtrise des problèmes de synchronisation.

- trois types de services de base :
  - Communication interpersonnelle
    - individuelle ou de masse
    - synchrone / asynchrone. Les communications synchrones ont des impératifs temps réels en général très stricts. (Exemple de la téléphonie sur Internet).
  - Recherche d'information : Ceci regroupe les fonctions d'archivage d'information (serveurs), de présentation (terminal, fonction kiosque..), de distribution. Le niveau de stockage varie du bas niveau (information telle qu'elle est stockée actuellement) jusqu'à un niveau de méta-information rejoignant ainsi une problématique de base de données multimédia. Les modes de recherche doivent garantir à l'utilisateur des fonctions usuelles telles que playback, navigation, zapping...
  - Édition et archivage : Ce troisième et dernier type de service de base est caractérisé par une situation très hétérogène qui pose de multiples problèmes aux auteurs, en particulier pour les fonctions d'enregistrement et d'édition de l'audio et de la vidéo.

2.4 Techniques de codage

2.4.1 Principes de base de la réduction de débit

Comme nous l'avons vu précédemment il est impératif de diminuer « l'encombrement » des signaux multimédias pour le stockage et la transmission, sans trop dégrader la qualité. Les méthodes analogiques y sont parvenues astucieusement mais avec des résultats qui, pour n'être pas négligeables, demeuraient insuffisants. La mise en œuvre de techniques numériques a totalement modifié la donne [46], [53].

Les méthodes de réduction du débit numérique (BRR : *Bit Rate Reduction*) sont aujourd'hui fondamentales dès lors qu'il y a transmission ou stockage d'information. On utilise souvent pour les désigner le terme de compression. La règle de base est « respecter autant que peut se faire l'information tout en réduisant l'encombrement du message ».

Il faut être conscient que la mise en œuvre d'algorithmes de réduction de débit entraîne inévitablement un décalage temporel qui peut devenir gênant voire rédhibitoire dans certaines utilisations « temps réel » [47].

Définition 1.07 :

Le taux de compression s'exprime :

- Soit par le rapport entre le volume initial des données et le volume après réduction.
- Soit en pourcentage du volume après réduction par rapport au volume initial.

Définition 1.08 :

On groupe ces méthodes en deux catégories : les méthodes sans pertes (*lossless*), dites aussi transparentes, qui ne détruisent aucune information ; les méthodes avec pertes (*lossy*) qui font disparaître une partie de l'information. On souhaite que ces dernières soient « virtuellement transparentes », c'est-à-dire que le récepteur ne perçoive pas la déperdition d'information.

La capacité de tolérance du récepteur est donc toujours un paramètre essentiel. Les données scientifiques ou bancaires ne supportent que des réductions absolument sans pertes tandis que les images et les sons peuvent supporter une légère dégradation sans que le spectateur ou l'auditeur ne se sente pénalisé.

Il s'agit toujours de promouvoir des modes plus économiques de description de l'information avec comme mots d'ordre « ne jamais transmettre ce qui a déjà été transmis et que l'on peut réutiliser » et « éliminer le superflu pour ne conserver que l'essentiel ». L'information superflue est en général énorme mais elle a été longtemps trop complexe à éliminer efficacement [46], [47], [48].

2.4.2 Le codage par plages

Ce type de codage plus connu sous les initiales RLC (*Run Lenght Coding*) ou RLE (*Run Lenght Encoding*) ou encore VLC (*Variable Lenght Coding*) exploite la redondance entre éléments successifs (des pixels d'une image ou des chiffes d'un tableau ou des lettres d'un texte) [53].

Le principe de cette technique de compression est le suivant : si une lettre a apparaît n fois successivement dans l'entrée, on peut remplacer les occurrences de a par le couple na. La répétition n fois de a est appelée longueur de répétition.

Algorithme 2.01 :

$répète \leftarrow 0; compteur \leftarrow 1;$

$RefCar \leftarrow entrée[compteur]; CarCour \leftarrow RefCar;$

$tantque\big(compteur \leq longueur(entrée)\big)faire$

$tantque(RefCar = CarCour)et(compteur \leq$

$longueur\big(longueur(entrée)\big)faire$

$Incrémente(compteur); incrémente(répète);$

$CarCour \leftarrow entrée[compteur];$

$ftq;$

$si\ répète < 4\ alors$

$pour\ i\ de\ 1$ à $répète\ écrire(RefCar)fpour$

$sinon\ écrire(@, répète, RefCar);$

fsi

$répète \leftarrow 0;$

$RefCar \leftarrow CarCour;$

ftq

2.4.3 Les codages entropiques

Elle caractérise la probabilité d'occurrence d'un élément d'un message ou d'un état d'un système.

Le physicien Américain Samuel Morse, proposa en 1844 un codage des lettres de l'alphabet qui tenait compte de la fréquence d'occurrence des lettres en anglais.

Le code ne disposait que de deux symboles correspondant à deux durées d'impulsions électriques sur une ligne téléphonique : brève et longue ou encore, en mode graphique, un point et un trait. Ce code binaire était conçu afin de rendre les transmissions les plus rapides possibles.

Dans ce but, la lettre E, la plus fréquente, est représentée par le mot le plus court : une brève ; la lettre T par une longue ; quant au J, assez rare, il bénéfice d'une brève et de trois longues.

Le fax met également en œuvre un codage entropique. Il s'agit, le plus souvent de transmettre des textes. Lors de l'analyse ligne à ligne on a statistiquement plus de chance de rencontrer des groupes de points noirs assez courts (les jambages des lettres) et des groupes de points blancs plus longs (les espaces entre lettres).

2.4.4 Le codage statistique

Le codage de Huffman utilise ce type de codage. Dans le codage de Huffman (1952), on commence par faire des statistiques de la fréquence d'apparition de chacun des éléments. Ils sont classés dans l'ordre décroissant de probabilité d'occurrence. On regroupe les deux éléments ayant la probabilité la plus faible pour en faire un nouvel élément dont la probabilité est la somme des probabilités des deux éléments initiaux ; ils peuvent être discriminés en utilisant un seul bit. On réitère l'opération en créant une arborescence des éléments suivant l'augmentation de la probabilité d'occurrence [53].

Propriété 2.01 :

Soient $S = \{s_1, s_2, \dots, s_m\}, \mathbb{B} = \{0,1\}$ des alphabets et soit $p = [p_1, p_2, \dots, p_m]$ la probabilité d'occurrence de chaque symbole. Un code optimal $c: S \to \mathbb{B}^*$ pour une source (S, p) a les propriétés suivantes :

(1) Si le mot de code $c(s')$ est supérieur à $c(s)$ alors $p_s \geq p_{s'}$;

(2) Parmi la longueur maximum des mots de code, il y en a deux ayant la forme $x0$ et $x1$, pour tout $x \in \mathbb{B}^*$.

Démonstration :

♣ (1) Supposant que longueur $w = c(s)$ est α et que la longueur $w' = c(s')$ est α'.

Soit c^* le code obtenu, par définition $c^*(s) = w'$ et $c^*(s') = w$. Alors la moyenne mots-longueurs $L(c) et\ L(c^*)$ satisfait :

$$L(c^*) - L(c) = (p_s\alpha' + p_{s'}\alpha) - (p_s\alpha + p_{s'}\alpha') = (p_s - p_{s'})(\alpha' - \alpha) \qquad (2.01)$$

quand c est optimal, il est non-négatif. Donc si $\alpha' > \alpha$ alors $p_s \geq p_{s'}$.

(2) Si aucun des deux mots de longueur maximum n'ont la forme indiquée, alors en effaçant le dernier bit de tous les mots de code de longueur maximum, on peut obtenir un meilleur code. ♦

Le codage de Huffman emploi deux constructions basées sur ses propriétés :

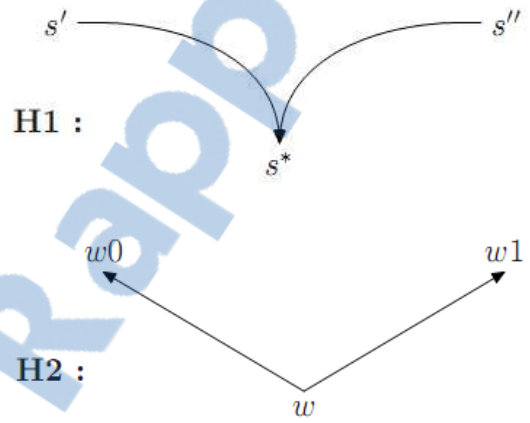
H1 : Soit la source (S, p), et soient $s' et\ s''$ deux symboles avec de faibles probabilités. Construire une nouvelle source (s^*, p^*) en remplaçant $s' et\ s''$ par un seul symbole s^*, avec une probabilité $p_{s^*}^* = p_{s'} + p_{s''}$.Tous les autres symboles gardent les mêmes probabilités.

H2 : Si nous donnons un code binaire h^* pour (s^*, p^*), avec $h^*(s^*) = w$, alors un code binaire h pour (S, p) est défini par les règles

$$h(s') = w0, h(s'') = w1, et\ h(u) = h^*(u)\ pour\ tout\ u \neq s', s'' \qquad (2.02)$$

Supposons que nous avons une source avec m symboles. La règle H1 peut être utilisée pour construire une séquence de sources avec un symbole plus petit que le précédent, ainsi le processus s'arrête sur la m-ième source laquelle a un seul symbole. Le code optimal pour la dernière source est l'une insignifiante à laquelle on attribue un mot vide pour un seul symbole. Inversement, H2 peut être utilisée pour construire des codes pour chacune des sources dans une séquence.

On trouvera ci-dessous un schéma d'arborescence pour 4 éléments.



Élément	Probabilité	Code
X1	1/2	0
X2	1/4	10
X3	1/8	110
X4	1/8	111

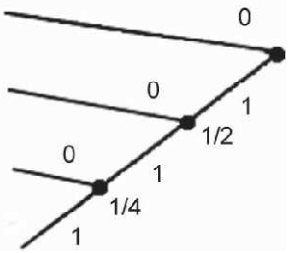


Figure 2.03: *Exemple de Codage de Huffman*

2.4.5 Les codages par dictionnaire

Appelé aussi algorithme dynamique. Le plus célèbre de ces codages est connu sous les initiales de ses auteurs LZW (Lempel, Ziv, Welch). Il est apparu en 1977 dans sa première version (LZ 77), spécialisée dans les données textuelles et qui est utilisée dans les programmes d'archivages de donnés comme PKZIP.

L'originalité du codage LZW réside dans la construction dynamique du dictionnaire au fur et à mesure de l'analyse du fichier [46], [49], [53].

Voici son fonctionnement pour une suite de caractères, mais il peut s'appliquer à tous types de données. On le trouve notamment dans les formats d'images GIF et TIFF.

Le code LZW, appliqué au texte, construit son dictionnaire à partir de la table des 256 signes et caractères du code ASCII définis par 8 bits ; il étend cette table à 4 096 « cases » définies grâce à l'utilisation de 12 bits. Il est basé sur le fait que des successions de caractères se retrouvent plus souvent que d'autres, par exemple les digrammes ES ou NE en français ou le trigramme ENT.

Le dictionnaire est construit dynamiquement d'après les caractères rencontrés.

- Lorsque l'algorithme rencontre pour la première fois un motif, il le repère par un index qui est le numéro d'une case vide de la table.
- Lorsque ce motif est à nouveau rencontré l'algorithme le remplace automatiquement par le numéro de la case.

La structure de la table est donc implicitement contenue dans le fichier à traiter. Elle pourra être reconstruite d'une manière analogue lors de la relecture du fichier.

Soit un message $X = x_1 x_2 \dots x_n$ dans l'alphabet $S = \{s_1, s_2, \dots, s_m\}$ et soit $D_0 = d_1, d_2, \dots, d_m$, où $d_i = s_i (i = 1,2, \dots, m)$. Le codage LZW construit son code $c(X) = c_1 c_2 c_3$ ... en série d'étape :

1) Etape 1 : Le premier symbole x_1 est une entrée $d_p \; dans \; D_0$. Par définition, coder x_1 veut dire :

$$c_1 = p. \tag{2.03}$$

La chaîne x_1x_2 n'est pas dans D_0.

$$d_{m+1} = x_1x_2 \qquad D_1 = (D_0, d_{m+1})$$

2) Etape k $(k \geq 2)$: Supposons que les étapes 1,2, … , $k-1$ ont été complétées. Ceci veut dire que les codes $c_1, c_2, \dots, c_{k-1}$ pour les $x_1x_2 \dots x_i \; de \; X$, et un dictionnaire $D_{k-1} = d_1, d_2, \dots, d_{m+k-1}$, ont été construits.

   Trouver une chaîne de longue w de la forme

$$w = x_{i+1} \dots x_j \quad (j \geq i+1) \tag{2.04}$$

   telle que w est dans D_{k-1}, c'est-à-dire $w = d_t$. Une chaîne existe certainement, car x_{i+1} est un unique symbole et se trouvant initialement dans le dictionnaire D_0. Par définition, wx_{j+1} n'est pas dans D_{k-1}. Coder le segment $x_{i+1} \dots x_j$ de X par $c_k = t \; et \; d_{m+k} = wx_{j+1}$, on a :

$$D_k = (D_{k-1}, d_{m+k}) \tag{2.05}$$

3) répéter la procédure jusqu'à la fin du message.

Théorème 2.02 :

Un code LZW construit suivant la définition précédente est exclusivement décodable

Démonstration :

♣ Supposons que les codes partiels $c_1c_2 \dots c_{k-1}$ peuvent être décodés avec succès en $x_1x_2 \dots x_i$. Supposant aussi qu'un dictionnaire D_{k-1} a été créé en ajoutant $k-1$ chaines dans D_0. L'argument donné ci-dessus montre que ces suppositions sont justifiées quand $k-1=1$, ainsi pouvons supposer que $k \geq 2$. Nous donnons les règles pour décoder c_k et construire $D_k = (D_{k-1}, d_{m+k})$.

Les règles de codage impliquent que c_k est l'index d'un chaîne $s_u \dots s_v \; dans \; D_{k-1}$.

Ces c_k sont décodés en prenant $x_{i+1} = s_u, \dots, x_j = s_v$.

Afin de construire le nouveau dictionnaire d_{m+k}, le codeur doit utiliser les valeurs de c_{k+1}. Si $c_{k+1} = r \leq m+k-1$ alors d_r est dans D_{k-1}, c'est-à-dire $d_r = s_a$ …

Ainsi, $x_{i+1} = s_a$, et l'entrée du nouveau dictionnaire est $d_{m+k} = s_u \dots s_v s_a$. Donc, la construction de c_{k+1} utilise seulement D_k, la seule possibilité est le cas difficile $c_{k+1} = m + k$. Maintenant, d_{m+k} est une chaîne de la forme $s_u \dots s_v s_z$, où $s_z = x_{j+1}$. D'où dans ce cas, $c_k c_{k+1}$ est la forme codée de $s_u \dots s_v s_u \dots s_v s_z$, et en fait $x_{j+1} = s_u$. En d'autre terme $d_{m+k} = s_u \dots s_v s_u$.♦

Prenons la suite de caractères E A M O E A E G :

• première entrée E qui se trouve dans la table des caractères ASCII sous le code 69 ;
• deuxième entrée la chaîne EA, ne se trouve pas dans la table et devient le motif 257 ;
• troisième entrée la chaîne AM, ne se trouve pas dans la table et devient le motif 258 ;
• quatrième entrée la chaîne MO, ne se trouve pas dans la table et devient le motif 259 ;
• cinquième entrée la chaîne OE, ne se trouve pas dans la table et devient le motif 260 ;
• sixième entrée la chaîne EA, se trouve dans la table case 257, elle est remplacée par son index 257 ;
• septième entrée, l'entrée précédente étant indexée, on s'intéresse à la chaîne EAE qui devient le motif 261.

C'est ainsi que pas à pas le signal établit lui même la table qui le décrit.

On remarque que de la case 256 à la case 512 les 256 motifs indexés, ce sont ceux qui apparaissent le plus souvent, sont définis à l'aide de 9 bits. Lorsque ces motifs sont des suites de 2 bits, le passage par la table fait gagner 7 bits (9 bits au lieu de 2×8). Lorsqu'il s'agit d'un motif de 3 lettres le gain est de 15 bits (9 bits au lieu de 3×8). S'il s'agit d'un motif de 4 lettres le gain est de 23 bits. Plus le motif est long plus le gain est important.

Il suffit de remplacer les séquences de deux lettres par celles de deux pixels pour comprendre l'intérêt du codage pour les images.

Cet algorithme astucieux et peu exigeant donne de bons résultats sur des images simples définies sur peu de bits (par exemple sur des images N & B ou en 256 couleurs). Mais il devient peu efficace dès que les images deviennent plus complexes. Il ne dépasse pas un taux de compression de 2:1 pour des images en 8 bits par couleur (24 bits).

Les algorithmes JPEG qui seront étudiés plus en détail sont de type codage à dictionnaire, mais les motifs retenus sont bien plus complexes.

2.4.6 Codage arithmétique

Le codage arithmétique (CA) est un codage statistique qui attribue à une suite de symboles une valeur réelle. Il consiste à découper l'intervalle des réels [0, 1) en sous-intervalles, dont les longueurs sont fonctions des probabilités des symboles. Le codage arithmétique n'attribue pas un code à chaque symbole comme Huffman et les autres codages par blocs, mais un code au message tout entier [50], [51].

Définition 2.09 :

Supposons qu'une source émet des symboles S et que la probabilité sur S^r est connue.

Pour $X \in S^r$, on a $P = P(X)$, définie n_p comme étant le dernier entier tel que $2^{n_p} \geq \frac{1}{P}$, et on a :

$$n = n_p + 1 \tag{2.06}$$

Le codage arithmétique $c: S^r \to \mathbb{B}^*$ est défini en prenant $c(X)$ comme étant le mot $z_1 z_2 \dots z_n$ qui représente un unique entier c tel que $c - 1 \leq 2^n a(X) < c$. La condition $n = n_p + 1$ garantie que le nombre $0.z_1 z_2 \dots z_n$ est dans l'intervalle I_X.

Les tableaux suivants présentent un exemple de codage arithmétique avec le message AAOEU.

Alphabet	Probabilité	Probabilité Cumulée	Partition Initiale
A	0.2	0.2	[0 0.2)
E	0.4	0.6	[0.2 0.6)
I	0.1	0.7	[0.6 0.7)
O	0.2	0.9	[0.7 0.9)
U	0.1	1.0	[0.9 1.0)

Message	Gauche G	Taille T	Droite D
A	0.0000	0.2000	0.2000
A	0.0000	0.0400	0.0400
O	0.0280	0.0080	0.0360
E	0.0296	0.0032	0.0328
U	0.0325	0.0003	0.0328

Tableau 2.04 : *Codage arithmétique*

Soit l'alphabet {A,E,I,O,U} avec les probabilités {0.2, 0.4, 0.1, 0.2, 0.1}. Le codage arithmétique est fait à partir de l'intervalle initial [0, 1) et au fur et à mesure du codage, la longueur de l'intervalle diminue en tenant compte du sous-intervalle précédent.

Nous nous servons des formules :

$$(G = GPI + GM * TPI)\ et\ (T = TPI * TM) \quad (2.07)$$

pour construire le nouvel intervalle où les lettres signifient : G-gauche, T-taille, M-message et P-précédant.

Le premier symbole A du message réduit l'intervalle initial à [0 0.2). Le deuxième symbole A du message réduit ce dernier intervalle à [0 0.04) (1/5 de l'intervalle précèdent).

Le symbole O réduit l'intervalle à [0.028 0.036). Le symbole E diminue l'intervalle à [0.0296 0.0328). Enfin, le symbole final U réduit à [0.03248 0.0328). Finalement, tout réel dans l'intervalle [0.03248 0.0328) codera le message AAOEU. Le codage arithmétique est présent dans la norme JPEG (dans les modes Extended DCT-based processes et Lossless processes) et JPEG2000.

Les méthodes de codage statistiques construisent les mots-codes à partir d'un dictionnaire prédéfini, basé sur les statistiques de l'image elle-même. Ce dictionnaire est indispensable pour le décodage.

2.4.7 Codage par prédiction linéaire

Les algorithmes qui utilisent le codage par prédiction exploitent la redondance spatiale DPCM (*Differential Pulse Code Modulation*) et ADPCM (*Adaptive Differential Pulse Code Modulation*). La modulation DPCM code les différences entre les valeurs successives en PCM. Ces valeurs sont en général faibles et conduisent donc à un petit nombre de bits.

On estime que la réduction de débit audio peut atteindre 25 % par rapport à un signal PCM. Il est à noter que ce mode de réduction de débit n'entraîne aucune perte d'information [53].

L'ADPCM (*Adaptive* DPCM) est plus complexe. On utilise un algorithme de prédiction qui prend en compte « l'histoire du signal » en étudiant la variation de plusieurs échantillons antérieurs. Il est suivi d'un comparateur entre signal réel et signal prédit qui détermine une valeur d'erreur.

C’est cette valeur d’erreur qui est transmise. Lors du décodage elle est ajoutée à la valeur prédite élaborée par l’algorithme de prédiction du décodeur. La valeur de l’erreur, généralement faible, détermine le pas de quantification qui lui est appliquée (souvent 3 ou 4 bits seulement).

L’ADPCM réduit le débit dans une forte proportion avec un taux supérieur à 4:1 dans le cas d’échantillons codés sur 16 bits et d’une différence transmise sur 4 bits.

Cependant, elle peut générer des pertes d’information.

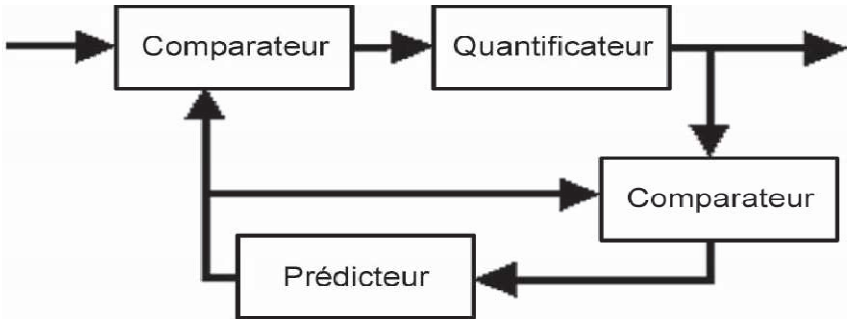


Figure 2.04: *Synoptique d’un codeur ADPCM*

2.4.8 Les codages de type psychophysiologique

Le mot d’ordre est ici « ne conserver que ce que le spectateur, ou l’auditeur, peut effectivement utiliser ». La connaissance des limites de nos possibilités de perception peut permettre d’éliminer, sans dommages apparents, une partie des informations initialement contenue dans le signal.

Cette élimination vient en appui des algorithmes de réduction de débit sans pertes. Elle permet, pour la distribution des programmes d’obtenir des débits faibles, voire très faibles, au prix de dégradations peu perceptibles.

Ces méthodes de réduction de débit entraînant des pertes peu sensibles doivent être évitées (ou maniées avec la plus extrême réserve) dans le cadre de la production ou de la postproduction où la perte de certaines données peut devenir très pénalisante lors des traitements ultérieurs [49].

2.4.9 Remarque

Dans toute la suite de l'ouvrage on parlera d'une information multimédia qui est l'image. La qualité visuelle des images numériques augmente continuellement avec le développement de nouvelles techniques d'affichage (multirésolution, transmission progressive) et de nouvelles technologies d'acquisition (haute définition, nouveau hardware). Cependant, la taille de ces images augmente proportionnellement à leur qualité et leur stockage et transmission constituent donc les enjeux principaux dans le monde numérique. La compression s'impose comme une étape incontournable pour optimiser l'utilisation de ces grands volumes d'informations dans les réseaux informatiques. L'objectif principal de la compression d'image est de réduire la quantité d'information nécessaire à une représentation visuelle fidèle à l'image originale.

2.5 Les codages d'images

On distingue plusieurs manières de coder les images. Nous différencions les schémas de compression selon la perte d'informations. Les méthodes réversibles, utilisent uniquement le principe de la réduction de la redondance et n'engendrent pas de perte. Les méthodes irréversibles définissent une représentation approximative de l'information.

Dans les paragraphes suivants, nous présentons quelques théories et aspects des nouvelles techniques de compression des images numériques.

2.5.1 Généralités sur la compression d'image

Une image peut être représentée sous une forme vectorielle ou sous la forme d'une matrice de points, bitmap. Dans l'ouvrage, le terme image correspond au type bitmap ou matrice de points.

Une image est une matrice de (M × N) points appelés pixels et à chaque pixel est associé une ou plusieurs valeurs d'intensité qui se combinent pour déterminer la couleur.

2.5.2 Taux de compression et redondance

Le taux de compression soumis à une image est directement proportionnel à la quantité de redondance d'information qu'elle possède [53].

2.5.2.1 Taux de compression

Le taux de compression est utilisé pour mesurer le résultat d'un procédé de compression. Il est représenté :

- soit comme une formule : $\sigma = \frac{I_o}{I_c}$
- soit comme un facteur : $\sigma = (I_o/I_c) : (1)$
  Dans ces deux équations, I_o est la taille de l'image originale en octet et I_c la taille de l'image comprimée.
- Le taux de compression peut être aussi quantifié par le nombre moyen de bits par pixel (bpp), par $\sigma = \frac{Bits_{I_c}}{Pixels_{I_o}}$.
  L'élément $Bits_{I_c}$ est le nombre total de bits de l'image comprimée et $Pixels_{I_o}$ est le nombre total de pixels de l'image originale.

2.5.2.2 Redondance

Une image numérique présente la particularité de posséder des corrélations importantes entre les pixels voisins. Cette corrélation est vue comme une redondance des informations pertinentes. La redondance peut être de deux natures : la redondance spatiale qui apparaît directement entre les pixels voisins de l'image originale et la redondance spectrale qui est liée aux fréquences et qui est acquise avec les transformations de domaines. La redondance dans le domaine spatial n'est pas facilement identifiable et généralement ne fournit pas toujours un bon taux de compression. Il est donc nécessaire de faire une transformation pour obtenir une décorrélation de l'information spatiale et un groupement d'énergie fréquentielle.

2.5.3 Critères psychovisuels et compression

Les méthodes de compression sans perte ne causent aucun problème visuel car ils sont totalement réversibles. Par contre, les procédures de compression avec pertes diminuent la qualité de l'image. Plus le taux de compression est important, plus les distorsions apparaissent dans l'image. Le point critique est la définition de la quantité de distorsions par rapport à la qualité de l'image. Le SVH (pour Système Visuel Humain) possède des caractéristiques particulières qui doivent être prise en compte.

Notre œil est capable de distinguer environ sept millions de couleurs [53]. Quand nous regardons une image notre système visuel doit résoudre beaucoup de contraintes : perception 3D, ombres, objets cachés, etc. En fait, le SVH tente de donner un sens visuel à chaque objet. Notre perception est influencée par ce que nous nous attendons à voir, c'est le cas des illusions optiques par exemple.

La perception de la couleur est influencée fortement par la saturation et la luminance. La saturation est la quantité de blanc ajoutée à une couleur, et la luminance est la mesure de lumière réfléchie par un objet. L'œil humain possède des sensibilités différentes suivant l'orientation des contrastes. Il est beaucoup plus sensible à la luminance qu'à la chrominance. Les systèmes de compression avec pertes changent généralement d'espace couleur. Il est conseillé de ne pas dégrader beaucoup la composante de luminance. La norme JPEG, par exemple, sous-échantillonne les deux composantes de chrominance, et ne change pas la composante de luminance.

2.5.4 Contours et Texture

La perception des contours fait partie des fonctions essentielles du SVH. Nous délimitons mentalement les objets qui sont dans une image grâce à leurs contours, et nous sommes très sensibles à la dégradation de ces contours. La théorie des formes explique leur importance pour la perception correcte des objets partiellement cachés. La fermeture des contours pour produire une forme visuelle connue par notre mémoire est essentielle.

2.5.5 Codage sans perte

La compression sans perte ou codage entropique ou codage réversible, dans une image, permet de retrouver la valeur exacte du signal comprimé lorsqu'il n'y a aucune perte de données sur l'information d'origine. En fait, la même information est réécrite d'une manière plus concise. Le processus de codage sans perte crée des "mots-codes" à partir d'un dictionnaire statistique ou d'un dictionnaire construit dynamiquement. Ces processus s'appuient sur des informations statistiques de l'image. Les codes statistiques les plus répandus sont le codage d'Huffman, le codage LZW, le codage par plages et le codage arithmétique. Le codage statistique permet de s'approcher au mieux de l'entropie. Ils ont pour principe d'associer aux valeurs les plus probables les mots binaires les plus courts.

2.5.6 Codage avec pertes

Les méthodes avec pertes (lossy) ou irréversibles sont des méthodes qui tirent parti d'une corrélation existante dans l'image. L'information perdue est due à l'élimination de cette redondance, ceci rend possible une compression plus importante. La perte d'information est toujours discutable et nous nous posons alors la question de la limite acceptable. Cette limite est définie par le type d'application, comme les images médicales ou satellites par exemple. La quantification est un des mécanismes utilisé dans les algorithmes de compression, qui produit des pertes d'information.

2.5.6.1 Quantification

La quantification fait partie de plusieurs méthodes de compression d'image. L'objectif est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image.

2.5.6.2 Quantification Scalaire

La quantification scalaire SQ - (Scalar Quantization) est une procédure qui associe à une variable continue X une variable discrète x. Pour cela on associe à x la valeur quantifiée

$$xq = Q(X) \tag{2.08}$$

où Q est une fonction (non linéaire) de quantification de $\mathbb{R} \rightarrow \mathbb{Z}$.

La quantification scalaire utilisée, en pratique, dans les images sont des quantifications basées en zone morte dans lesquelles l'intervalle de quantification est centré à l'origine et est de taille multiple de la taille des autres intervalles de quantification.

2.5.6.3 Vectorielle

La quantification vectorielle VQ - (Vector Quantization) a été développée par Gersho et Gray [54] et elle fait aujourd'hui l'objet de nombreuses publications dans le domaine de la compression numérique. Le principe de la quantification vectorielle est issu du travail de Shannon qui montre qu'il est toujours possible d'améliorer la compression de données en codant non pas des scalaires, mais des vecteurs.

Un quantificateur vectoriel Q associe à chaque vecteur d'entrée $X_i = (x_j, j = 1 \dots k)$ un vecteur $Y_i = (y_j, j = 1 \dots k) = Q(X_i)$, ce vecteur Y_i étant choisi parmi un dictionnaire (code-book) de taille finie. La VQ produit de meilleurs résultats que la SQ, néanmoins la VQ nécessite un codage complexe et de grandes capacités de mémoire.

2.5.6.4 Codage prédictif avec pertes

Il existe des techniques qui exploitent la redondance spatiale, cependant la prédiction est faite par approximation. Ces algorithmes ont comme objectif de rechercher un modèle de représentation le plus adéquat de l'information à coder afin d'obtenir un coût de codage minimal. L'idée est de coder l'erreur de prédiction au-dessus d'un seuil. Ce seuil peut être défini par rapport à la qualité de l'image ou le niveau de compression espéré.

2.5.6.5 Codage par transformation

Les méthodes qui utilisent cette technique utilisent des transformations pour produire une décorrélation des redondances spectrales. Les pixels passent d'un espace où ils sont fortement corrélés dans un autre espace où leur corrélation est moindre. Lors de chaque transformation, le signal d'origine est remplacé par sa représentation dans un autre domaine. Dans divers algorithmes, cette transformation d'espace est accompagnée d'une quantification et d'un codage entropique pour accomplir la compression de l'image. Ceci est le cas des normes standards de compression : l'algorithme JPEG qui utilise la transformation en cosinus discrète DCT et l'algorithme JPEG2000 qui utilise la transformation en ondelettes DWT.

Un codage par transformée décompose les signaux sur une base orthonormée $B = \{g_m\}$ $0 \leq m < n$ et optimise la compression des coefficients de décomposition. On étudie la performance d'un tel code d'un point de vue bayésien, en modélisant le signal comme une réalisation d'un processus aléatoire *F[n]* de taille *N*, dont la distribution de probabilité est supposée connue a priori.

Décomposons *F* sur *B* :

$$F = \sum_{m=0}^{N-1} F_B [m] \, g_m \qquad (2.09)$$

Chaque coefficient $F_B[m]$ est une variable aléatoire définie par :

$$F_B[m] = \langle F, g_m \rangle = \sum_{n=0}^{N-1} F[n]\, g_m^*[n] \tag{2.10}$$

Pour centrer en zéro les variations de $F_B[m]$, on code $F_B[m] - E\{F_B[m]\}$ et on garde en mémoire la valeur moyenne $E\{F_B[m]\}$. Cela revient à supposer que $F_B[m]$ est de moyenne nulle [66].

a. Transformée en ondelettes

La transformée en ondelettes est une description multi-résolution d'une image. Elle décompose une image en plusieurs sous-bandes dans 3 directions différentes : horizontale, verticale et diagonale.

Elle consiste à décomposer le signal x[n] en basses et hautes fréquences en utilisant respectivement des filtres passe-bas et passe-haut [61] :

$$H(\omega) = \sum_k h[k]\, e^{-jk\omega} \text{ et } G(\omega) = \sum_k g[k]\, e^{-jk\omega} \tag{2.11}$$

où $H(\omega)$ et $G(\omega)$ doivent être orthogonaux :

$$|H(\omega)|^2 + |G(\omega)|^2 = 1 \tag{2.12}$$

Les coefficients obtenus sont :

$$\begin{aligned} c[j-1,k] &= \sum_n h[n-2k]\, c[j,n] \\ d[j-1,k] &= \sum_n g[n-2k]\, c[j,n] \end{aligned} \tag{2.13}$$

La reconstruction du signal original notée IDWT est le processus inverse du DWT. Elle est résumée par la formule suivante :

$$c[j,n] = \sum_k h[n-2k]\, c[j-1,k] + \sum_k g[n-2k]\, d[j-1,k] \tag{2.14}$$

Le résultat est une image d'approximation qui a une résolution divisée par deux et trois images de détails qui donnent les erreurs entre l'image originale et l'image d'approximation.

Cette transformation est répétée autant de fois que nécessaire pour obtenir le nombre voulu de sous-bandes. Après quelques niveaux de décomposition, les fréquences basses sont concentrées sur le coin haut à gauche de la transformée et ressemblent à une version comprimée de l'image originale.

b. Quantification

Pour construire un code fini, chaque coefficient $F_B[m]$ est approché par une variable quantifiée $\widetilde{F_B}[m]$, qui prend ses valeurs dans un nombre fini de nombres réels. Une quantification scalaire approxime chaque $F_B[m]$ indépendamment. Lorsque les coefficients $F_B[m]$ sont très indépendants, un quantificateur vectoriel qui quantifie ensemble les *N* coefficients du vecteur F_B, peut nettement améliorer la performance d'un quantificateur scalaire. Cependant la base *B* est généralement choisie pour réduire l'interdépendance des coefficients, ce qui limite l'amélioration d'un quantificateur vectoriel, et comme les quantificateurs scalaires demandent moins de calculs, ils sont le plus souvent utilisés. Après quantification, le signal reconstruit est :

$$\widetilde{F} = \sum_{m=0}^{N-1} \widetilde{F}_B [m] \, g_m \tag{2.15}$$

- Quantification scalaire [66]

Si la source *X* prend des valeurs réelles arbitraires, elle ne peut être codée sur un nombre fini de bits. Un quantificateur scalaire approche *X* par $\widetilde{X} = Q(X)$ qui prend ses valeurs dans un ensemble fini.

On étudie l'optimisation d'un tel quantificateur afin de minimiser le nombre de bits nécessaire au codage de $\widetilde{X}$ pour une erreur quadratique moyenne donnée :

$$d = E\left\{(X - \widetilde{X})^2\right\} \tag{2.16}$$

Supposons que X prenne ses valeurs dans [*a,b*] qui peut être éventuellement l'axe réel. On décompose [a,b] en k intervalle $\{]y_{k-1},y_k]\}$ avec $1 \le k \le K$ de longueurs variables, avec $y_0=a$ *et* $y_k=b$

Un quantificateur scalaire approche :

$$\text{tout } x \in \left]y_{k-1}, y_k\right] \text{ par } x_k \text{ , } Q(x) = x_k \tag{2.17}$$

Les intervalles :

$$\left]y_{k-1}, y_k\right] \tag{2.18}$$

s'appellent des boites de quantification.

- Quantificateur à haute résolution [21]

Soit *p(x)* la densité de probabilité de la source aléatoire *X*. L'erreur de quantification est :

$$d \;=\; E\left\{(X-\tilde{X})^2 = \int_{-\infty}^{+\infty}\left(x-Q(x)\right)^2 p(x)dx\right\} \tag{2.19}$$

Un quantificateur est dit à haute résolution si *p(x)* est approximativement constant sur chaque boite de quantification $\left]y_{k-1},y_k\right]$, de taille $\Delta_k = y_k - y_{k-1}$. C'est le cas si les tailles Δ_k sont suffisamment petites par rapport au taux de variation de *p(x)*, si bien qu'on peut négliger ces variations à l'intérieur de chaque boite de quantification. On a alors :

$$p(x) = \frac{p_k}{\Delta_k} \;\text{ pour }\; x \in \left]y_{k-1}, y_k\right] \text{ où } \; p_k \;=\; Pr\{\, X \in \left]y_{k-1}, y_k\right]\} \tag{2.20}$$

- Quantificateur uniforme

Le quantificateur uniforme est un cas particulier important où toutes les boites de quantification sont de même taille : $\Delta = y_k - y_{k-1}$ pour $1 \le k \le K$.

Pour un quantificateur uniforme à haute résolution, la distorsion quadratique moyenne devient :

$$d \;=\; \frac{\Delta^2}{12\sum_{k=1}^{K} p_k} \;=\; \frac{\Delta^2}{12} \tag{2.21}$$

Elle est indépendante de la densité de probabilité *p(x)* de la source.

- Quantificateur entropique

On veut minimiser le nombre de bits nécessaire au codage de valeurs quantifiées $\tilde{X}=Q(X)$ pour une distorsion fixée $d = E\left\{(X-\tilde{X})^2\right\}$. Le théorème de Shannon montre qu'en moyenne le nombre minimum de bits nécessaire pour coder $\tilde{X}$ est l'entropie $H\left(\tilde{X}\right)$. Les codes arithmétiques ou de Huffman requièrent un nombre proche de cette borne inférieure. Il faut donc trouver un quantificateur qui minimise $H\left(\tilde{X}\right)$.

2.5.7 Domaine spatial

Les méthodes de compression d'images dans le domaine spatial exploitent la redondance entre un pixel et son voisinage, ou entre certaines régions de l'image [46], [49].

2.5.7.1 Quadtree

La technique de décomposition par quadtree est basée sur une approche récursive d'un codage arborescent. Une image hétérogène de taille $2^n \times 2^n$ est alors divisée en quatre sous-régions, de taille $2^{n-1} \times 2^{n-1}$. Ce processus est refait jusqu'à ce que chacune des régions soit déclarée homogène selon un critère choisi. Certains travaux proposent d'optimiser ce compromis en découpant l'image en blocs de tailles variables. Des zones considérées homogènes seront découpées en grands blocs alors que des zones texturées ou contenant des contours seront découpées en blocs plus petits comme le montre la figure 2.05

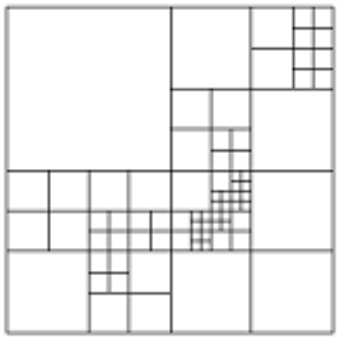


Figure 2.05: *Décomposition en quadtree*

2.5.7.2 Décomposition en plans binaire

Cette technique est principalement employée pour les images à niveau de gris. L'idée est de décomposer l'image en huit images binaires, une pour chaque bit de niveau de gris, en commençant par les bits de poids les plus forts.

Figure 2.06: *Décomposition en plan binaires des bits de poids forts jusqu'aux bits de poids faibles*

Sur chaque image binaire sont lancées des procédures particulières pour effectuer la compression. L'image binaire représentant le bit de poids le plus fort, première image (8), apporte le plus haut taux de compression, tandis que la compression sur l'image binaire de poids le plus faible, dernière image (1) a quasiment une compression nulle.

Maniccamam et Bourbakis [55] ont présenté une méthode de compression sans perte basée sur la décomposition de l'image en plans binaires. Dans leur approche, chaque plan binaire est également décomposé en régions, et chaque région est parcourue selon un ensemble de chemins différents. Cet ensemble de chemins est composé de 32 chemins standards. Pour chaque chemin de cet ensemble est réalisé un codage RLE pour déterminer le taux de compression optimal. La compression de l'image est déterminée pour l'ensemble des compressions de chaque région.

2.5.7.3 Fractales

La compression par fractale est une technique de compression avec pertes encore peu utilisée. Un fractal est une structure géométrique qui se reproduit, dans une boucle infinie, par transformation affine (translation, rotation et mise à l'échelle). Cette structure se refait à toutes les échelles de forme réduite et légèrement déformée. La compression par fractale est basée sur le principe qu'il existe des similarités entre différentes régions isolées d'image. Elle exploite les récurrences des motifs qui, après quelques traitements, peuvent permettre une compression.

2.5.8 Domaine fréquentiel

Les techniques de compression dans le domaine fréquentiel s'appuient sur une transformation de l'image vers un nouvel espace de représentation d'énergie fortement décorrélée. Cette décorrélation provoque une nouvelle représentation de l'image par la redistribution de l'énergie dans un nombre restreint de coefficients transformés. Cette énergie de l'image transformée est distribuée sous la forme de tranches énergétiques de basse, moyenne et haute intensités. Les transformations les plus courantes sont la DCT et la DWT.

2.5.8.1 La DCT ou Discrete Cosine Transform

Il s'agit d'analyser le signal $f\ (t)$ à l'aide d'une famille de fonctions mathématiques, dites fonctions de base, assez simples et telles qu'une série d'entre elles puisse représenter le signal considéré :

$$f\ (t)\ =\ \sum C_i\ \Psi_{\mathrm{i}}(t) \tag{2.22}$$

Les fonctions de base Ψ_{i} étant fixées, l'information relative au signal est portée par les coefficients $\mathrm{C_i}$ représentant le poids de chacun des termes de la série.

Les fonctions mathématiques sinusoïdales, sinus et cosinus (c'est la même chose à un décalage près dans le temps), assez simples d'emploi et bien adaptées à la description des phénomènes physiques périodiques, représentaient un bon candidat pour ce type de calcul.

La « décomposition en série de Fourier » conduit à représenter la fonction $f(t)$ de période :

$$T\ =\ 2\,\pi/\omega \tag{2.23}$$

par l'expression :

$$f\ (t)\ =\ a_0\ +\ \sum(a_{\,k} sin\ k\omega t\ +\ b_k\ cos\ k\omega t) \tag{2.24}$$

Elle permet d'obtenir la représentation d'un phénomène périodique par une superposition de composantes sinusoïdales dont les fréquences, ainsi que les amplitudes et les phases sont bien spécifiées.

Pour k = 1, on obtient la fondamentale de pulsation ω dont la période $T = 2\pi/\omega$ est identique à celle de la fonction $f(t)$; pour k = 2, 3…, on obtient les harmoniques dont les fréquences sont des multiples de celle de la fondamentale à laquelle ils viennent ajouter les détails qui affinent la coïncidence avec la fonction à représenter. Plus on utilisera d'harmoniques meilleure sera la représentation.

On a su étendre, sous le nom de Transformation de Fourier, cette méthode à des signaux non périodiques (ou non stationnaires) en supposant que leur période est infinie, ce qui conduit à un spectre de composantes pouvant comporter une infinité de composantes. Il s'agit d'une représentation dont l'existence n'est pas localisée dans le temps (sons) ou dans l'espace (image). C'est le jeu des interférences (additives ou destructrices) entre les différentes composantes qui fait apparaître ou disparaître à un instant (ou un lieu) donné une composante (une note de musique ou un motif graphique). Les informaticiens ont su mettre au point des algorithmes relativement simples (FFT : Fast Fourier Transform) pour réaliser ces opérations sur ordinateur. C'est ce type de transformée qui est notamment utilisé dans la compression JPEG [53].

2.5.8.2 La compression JPEG

Il existe plus d'une cinquantaine de types de formats d'image. Pour chacun d'entre eux la structuration des données et les attributs sont différents. La standardisation d'un format d'image permet de régler l'utilisation, la divulgation et la production de logiciels et de hardware compatibles avec le format standard. Le format standard JPEG est le format d'image le plus populaire [50], [53]. Son successeur, le JPEG2000, semble s'établir dans le domaine de l'image numérique. Le JPEG2000 possède des fonctionnalités supplémentaires par rapport au format JPEG. Cependant, la plupart des appareils numériques (appareils photos, caméscopes, téléphones portable, etc.) et les logiciels qui capturent et traitent les images sont au format JPEG.

- **La norme JPEG**

Le comité « Joint Photographic Expert Group » a été créé en 1986 par la jonction de plusieurs groupes qui travaillaient sur la photographie. Ce comité a produit la norme de compression d'images photographiques qui a été standardisée (ISO/IEC/10918-1/1994) et a reçue son nom JPEG.

Il est devenu le format le plus populaire très rapidement parce qu'il a été conçu avec différentes contraintes :

- L'algorithme JPEG doit être implémentable sur une grande variété de types de CPU (unité centrale de calcul) et sur des cartes plus spécialisées (appareil photo numérique et téléphone portable par exemple).
- Il doit pouvoir compresser efficacement tout type d'images réelles (images photographiques, médicales) avec pertes et sans perte.

Il possède quatre modes de fonctionnement : séquentiel (baseline), progressif (extended DCT-based), sans perte (Lossless), hiérarchique (hierarchical).

Entre les 4 modes de compression de la norme JPEG, le séquentiel ou baseline est le mode principal le plus répandu. Il est basé sur la transformation DCT, quantification scalaire et le codage d'Huffman sur pixels de 8 bits par plan de couleur [50].

- **Performances**

Le traitement par DCT est très performant. L'expérience montre qu'un facteur de compression de 10 produit généralement une image indiscernable de l'original. Les résultats dépendent évidemment de la complexité de l'image ; des images fortement bruitées (le bruit est un phénomène complètement incorrellé qui ne présente pas de redondances) seront notamment difficiles à traiter.

On peut par ailleurs noter que le système ne donne a priori totale satisfaction que pour les structures dont les dimensions sont de l'ordre de celle de la fenêtre, de largeur constante, et qui peuvent être représentées correctement par des fréquences donnant des longueurs d'onde du même ordre que cette largeur. Toute erreur sur le premier motif « en haut à gauche » qui représente la valeur moyenne du bloc entraîne un effet de mosaïque ou effet de bloc (*blockiness*) ; et c'est bien ce qui se produit dès lors que le taux de compression devient trop élevé. Cette structure régulière des artefacts peut évidemment devenir gênante voire désastreuse.

Les algorithmes JPEG ont prouvé leur efficacité ainsi que leur souplesse aussi bien dans les applications professionnelles (bibliothèques numérisées, stockage d'images médicales,

traitement d'images…) que pour des utilisations « grand public » : photographie numérique, transmission de documents par Internet… JPEG est devenu un standard incontournable dans ces domaines de l'archivage et de la transmission, mais ce n'est pas un standard de production.

2.5.8.3 JPEG-LS - [ISO/IEC/14495-1/ 1999]

La compression JPEG en mode sans perte n'est pas optimisée. Les objectifs du comité étaient de concevoir un mode réversible permettant une compression de l'image à 50%.

Malheureusement, un code réversible est quasiment impossible avec l'utilisation de la DCT. Les erreurs d'arrondis dues à la précision limitée de calcul sont toujours présentes.

Le nouveau standard de compression sans perte, le JPEG-LS est basé sur une variation de la méthode LOCO-I - (LOw COmplexity lossless compression method). Dans JPEG-LS le procédé de compression est composé de trois parties : la prédiction de la valeur du pixel qui est faite par rapport aux pixels voisins en utilisant l'approche de prédiction MED (Median Edge Detection) et la détermination d'un contexte.

Ce contexte représente l'environnement du pixel à coder et ses voisins. L'idée est de prendre le meilleur environnement qui affine la prédiction avant le codage et de réduire le nombre de paramètres de l'erreur de prédiction ; le codage de l'erreur de prédiction dont l'approche est de réinsérer l'erreur de prédiction dans le système puis de la comparer à d'autres mesures d'erreur.

Le JPEG-LS possède une option de haut taux de compression, mais cette option est quasiment sans perte (near lossless). Les observations expérimentales montrent que pour des taux supérieurs à 1,5 bpp, le JPEG-LS en mode near lossless donne de meilleures performances que JPEG.

2.5.8.4 JPEG2000 - [ISO/IEC/15444-1/ 2000]

Le JPEG2000 remplace le JPEG comme le format standard pour la compression des images. Il a été réalisé dans la perspective de répondre aux exigences des nouvelles applications les plus diversifiées, comme la multirésolution par exemple.

La compression JPEG2000 est composée de plusieurs étapes selon les schémas avec pertes et sans perte. Tout d'abord, un changement d'échelle est effectué dans chaque composante couleur RGB, l'échelle est changée de (0, 255) à l'échelle (-128, 127) par une simple soustraction de 128 de chaque valeur. Après le changement d'échelle, l'image est soumise à une transformation de plans couleurs, facultative, de RGB à YCbCr.

Cette transformation peut être réversible (sans perte) ou irréversible (avec pertes). Chaque plan chromatique de l'image est découpé en petites images appelées tuiles, futile. Chaque tuile est considérée comme une image et est traitée de façon indépendante [49], [53].

Du fait de la complexité du mécanisme du JPEG2000, la décomposition de l'image en tuiles rend possible l'application du JPEG2000 sur des images de taille importante.

La transformation de domaine est faite à l'aide d'une décomposition en ondelettes, décrite précédemment. Malgré ces nombreuses fonctionnalités, le JPEG2000 possède quelques inconvénients. Il nécessite entre deux et six fois plus de cycles de CPU que JPEG et il n'est pas indiqué pour les machines avec faibles ressources comme les appareils photos numériques par exemple. L'algorithme JPEG est beaucoup moins complexe et il peut être implémenté en hardware.

2.6 Insertion de données cachées

2.6.1 Introduction

La cryptographie a été une première proposition pour sécuriser des transferts de documents numériques. Aujourd'hui les algorithmes de chiffrement modernes, avec des clefs de longueur importante, permettent d'assurer la confidentialité. Néanmoins, une fois décrypté, le document n'est plus protégé et il peut être distribué ou modifié malhonnêtement. La dissimulation d'information plus particulièrement l'insertion de données cachées peut être une réponse à ce problème. L'insertion d'une marque dans un document permet de l'authentifier et de garantir son intégrité, selon [59], [60], [61], [62], [63].

La figure 2.07 présente plusieurs techniques de dissimulation d'information. Ces techniques disposent de différentes caractéristiques et sont classifiées selon leurs applications et objectifs [63].

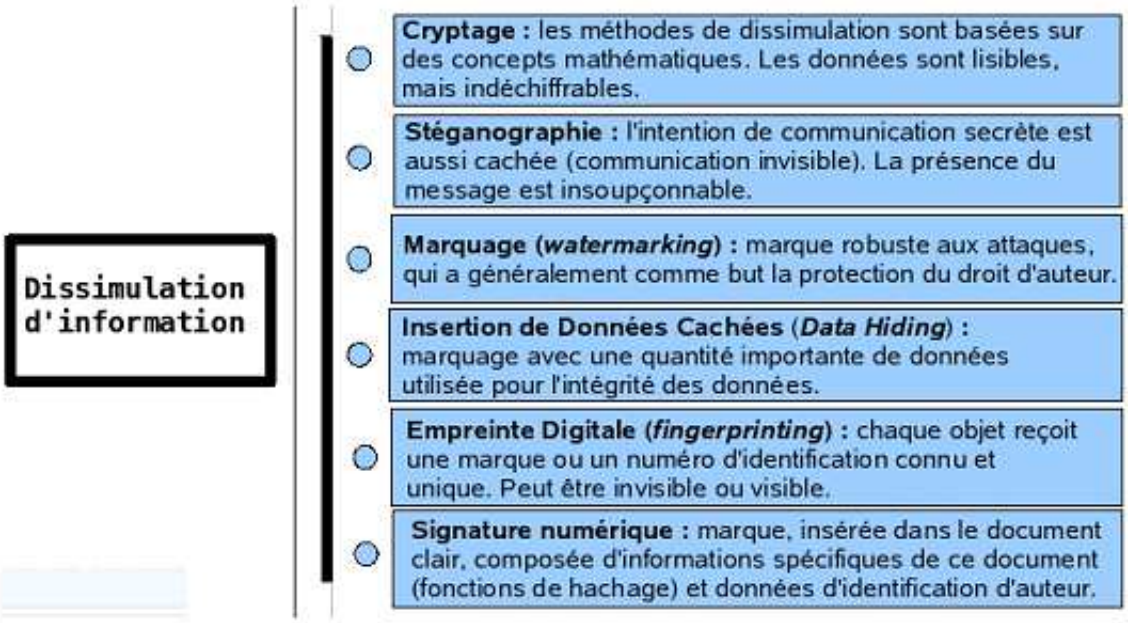


Figure 2.07: *Techniques de dissimulation d'information*

La figure 2.08 présente les termes particuliers aux techniques d'insertion de données cachées (IDC), qui seront utilisées tout au long de ce document. La Couverture (I_c) est le fichier original qui va être utilisé pour couvrir ou cacher une information. Le Message ou la marque est l'information à cacher. Le Porteur (I_p) est le fichier qui porte un message encapsulé. Plusieurs supports numériques peuvent être utilisés comme porteur pour cacher des informations comme par exemple les protocoles de communication et les cellules de mémoire. Cependant, le support numérique le plus employé est le fichier sous toutes ses formes, particulièrement celui sous la forme de données multimédias (documents texte, son, image, vidéo).

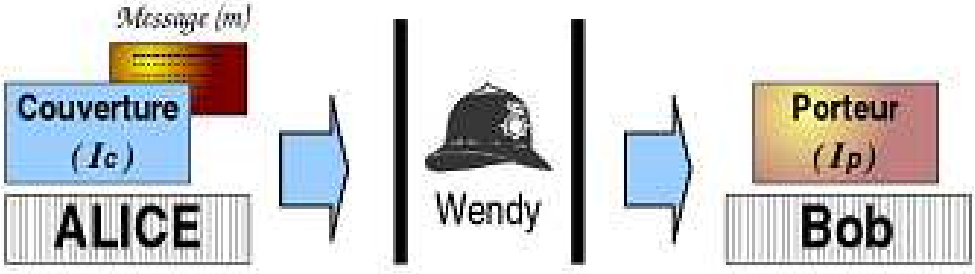


Figure 2.08: *Modèle de communication 3-tiers*

Dans les sections suivantes, nous présentons quelques théories et aspects des nouvelles techniques d'IDC, stéganographie et marquage ou tatouage d'image.

2.6.2 Généralités sur l'IDC

2.6.2.1 Conditions requises

Les méthodes d'insertion de données cachées demandent différentes propriétés selon leurs domaines d'application et leurs finalités. Nous donnons ces quelques définitions [63], [70], [71].

a. Indétectabilité

L'indétectabilité est la capacité d'un message à ne pas être détecté par des analyses statistiques. Cette condition est indispensable pour les systèmes de communication secrète basés sur la stéganographie.

b. Invisibilité/imperceptibilité

Les données cachées doivent être entièrement invisibles par le système visuel humain (SVH). L'opération d'insertion ne doit pas détériorer le porteur d'une façon perceptible.

L'invisibilité est une propriété fortement liée au marquage invisible et à la stéganographie.

c. Spécificité

Le message caché, après certains types d'attaques, peut subir des distorsions et devenir illisible. Le message doit être donc suffisamment spécifique pour être clairement identifiable lors de son extraction. Cette caractéristique est liée au marquage non-aveugle et marquage semi-aveugle.

d. Robustesse

La robustesse est l'aptitude à préserver les données cachées (message) face aux attaques. La robustesse correspond donc à la quantité d'énergie que possède la marque insérée. Une marque de forte énergie est robuste. Cette demande est fortement attachée à la plupart des types de marquages, particulièrement ceux pour la protection des droits d'auteurs.

e. Capacité

La capacité est la quantité d'information que le fichier couverture peut dissimuler. Elle est généralement mesurée en bits. Dans le contexte de marquage pour la protection des droits d'auteurs, la capacité n'est pas primordiale. L'insertion d'un numéro d'identification codé sur 64 bits suffit dans la plupart des applications. Néanmoins pour la stéganographie et l'IDC - (data hiding), cette propriété est très importante.

f. Détection et extraction

La détection est la découverte de la présence d'un message caché m, sans en connaître son contenu. Il existe donc une probabilité P_{ed} d'erreur de ne pas détecter ce message, même s'il est présent. D'autre part, il existe aussi une probabilité P_{fa} de fausse alarme, laquelle est la probabilité de détecter la présence d'un message qui n'existe pas.

Dans la détection, le souci est de trouver le meilleur arrangement de la probabilité d'erreur et de la probabilité de fausse alarme. Une forte probabilité P_{ed} pour les schémas de gestion de propriété n'est pas tolérée, néanmoins pour les schémas stéganographiques celle-ci peut être acceptable. Une forte probabilité P_{fa} n'est, en général, pas souhaitable dans les deux cas à cause du chargement du système.

L'extraction est l'obtention du message m à partir du fichier porteur I_p. L'extraction du message est interprétée différemment de la détection. Dans l'extraction, nous considérons toujours que l'image est marquée. Nous prenons en compte la probabilité P_{ee} d'erreur d'extraction (extraction d'un message incorrect). Cette probabilité est très liée à la capacité du message.

Plus la marque est invisible et robuste moins l'extraction sera performante et plus grande sera la probabilité d'erreur. Il existe une catégorie de marquage dont la probabilité P_{ee} est très faible, il s'agit du marquage non-aveugle où la couverture I_c est demandée lors de l'extraction.

2.6.2.2 Domaines d'insertion

Les techniques d'IDC, cas des images, sont liées aux différents espaces de représentations appelés domaines, et chaque domaine d'insertion dispose de divers schémas de marquage.

a. Spatial

Dans ce domaine, les méthodes modifient directement la valeur de la couleur du pixel. Ce sont des méthodes simples et peu coûteuses en temps de calcul. Elles sont consacrées aux marquages en temps réel demandés dans des environnements de faible puissance. Certaines techniques dans le domaine spatial peuvent être robustes aux attaques de type transformations géométriques [67].

b. Fréquentiel

Des schémas de marquage peuvent effectuer l'insertion du message dans des espaces transformés. Un espace transformé est obtenu après l'emploi d'une DCT (Transformée en Cosinus Discrète) ou DFT (Transformée de Fourier Discrète) ou autres. Cette stratégie rend le message plus robuste à la compression, puisqu'elle utilise le même espace qui sert au codage de l'image. Contrairement au domaine spatial, la marque insérée dans le domaine fréquentiel est très sensible aux transformations géométriques parce que ce genre de transformations modifie considérablement les valeurs des coefficients transformés [73].

c. Multi-résolution

L'espace multirésolution pour les images est devenu populaire après la création de la norme JPEG2000, décrite précédemment, qui utilise la DWT (Transformée par ondelettes discrète) pour transformer le domaine. L'image est décomposée en sous-bandes, ceci permet un isolement affiné des composantes basse-fréquences, qui est un espace d'insertion moins sensible. En plus, le contenu spatial de l'image est aussi conservé et peut être utilisé pour l'insertion d'information. D'autre part, la DWT est considérée comme une décomposition en canaux perceptifs qui facilite l'utilisation d'un modèle psychovisuel [76].

2.6.2.3 Mesures et modèles perceptuels

Dans les systèmes d'insertion de données cachées, la mesure de la perturbation apportée sur l'image lors de l'insertion du message est très importante. La démarche la plus employée est alors d'utiliser une métrique d'erreur quadratique moyenne (EQM) pour calculer le PSNR [77], [79], [81], [86], [87].

a. PSNR ou Peak Signal to Noise Ratio

Le PSNR est la mesure de la distorsion entre le signal marqué et le signal original.

Il est défini par :

$$PSNR = 10log_{10}\left(\frac{max^2}{EQM}\right) \quad , \quad EQM = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|I_c(i,j) - I_p(i,j)\right\|^2 \qquad (2.25)$$

où I_c est l'image couverture, I_p l'image porteuse, m le nombre de lignes, n le nombre de colonnes et max la dynamique du signal (pour les images la valeur maximale du pixel codé sur un octet est max = 255). L'unité du PSNR est le décibel dB, plus il est élevé et moins la distorsion est importante [85].

Malgré l'utilisation courante du PSNR pour mesurer la qualité des images, celui-ci n'est pas bien ajusté au Système Visuel Humain - SVH. Le SVH ne perçoit pas tous les signaux de la même façon, comme la sensibilité au contraste par exemple. L'utilisation seule du PSNR ne peut donc pas être considérée comme une mesure objective de la qualité visuelle d'une image [56].

La figure 2.09 montre que, même si le PSNR de l'image (b) est inférieur à celui de l'image (c), l'image (b) possède une meilleure qualité visuelle.

Figure 2.09: *Evaluation du PSNR comme mesure de qualité visuelle. (a) Image originale, (b) Image fortement contrastée PSNR=24,63 dB, (c) Image fortement comprimée PSNR=28,80 dB*

D'autres métriques à prendre en compte sont les phénomènes de perception humaine. Des méthodes de mesures perceptuelles des distorsions qui prennent en compte des éléments psychologiques et physiologiques de la perception humaine pour l'évaluation qualitative des images tatouées [57].

Néanmoins, l'évaluation qualitative est encore d'actualité et il n'y a pas encore une métrique standard. Les approches les plus connues sont les métriques avec pondération perceptuelle et par seuil de perception présentées ci-après.

b. Pondération perceptuelle

L'approche la plus pratique est l'introduction d'une pondération perceptuelle w au sein de la mesure classique d'erreur quadratique moyenne. Le $wPSNR$ (weighted PSNR) est défini par :

$$wPSNR = 10log_{10}\left(\frac{max^2}{wEQM}\right) \quad , \quad wEQM = \frac{1}{n}\sum_{i=1}^{n}\varphi_i^2(x_i - y_i)^2 \tag{2.26}$$

où φ_i est une pondération représentant l'importance du i-ème échantillon. Plusieurs pondérations ont été proposées [11].

La pondération la plus connue est celle de Watson, définit par les relations suivantes :

$$\varphi_i^2 = \frac{1}{\sigma_{b_i}^2 + V_i^2} \quad , \quad V_i = \frac{1}{\|\phi_i\|}\sum_{j\epsilon\phi_i}|x_j|^\rho \tag{2.27}$$

Dans ces formules, pour le i-ème coefficient, V_i est une mesure d'activité de voisinage, ϕ_i est l'ensemble qui représente les indices des voisins et enfin la variable σ_{b_i} est un seuil de visibilité qui dépend de la distance d'observation. Ce seuil est fixé à 10^{-2} pour JPEG2000. Les meilleurs résultats sont obtenus pour ρ = 1/2.

Le $wPSNR$ de Watson a été conçu pour les images dans le domaine DCT notamment JPEG. Il utilise une table de niveau de sensibilité pour les 64 coefficients d'un bloc DCT. Ceci permet de prendre en compte la sensibilité fréquentielle et les phénomènes de masquage dûs à la luminance et au contraste. Une version plus simplifiée du $wPSNR$ de Watson est utilisée dans JPEG2000.

c. Seuils de perception

Contrairement aux critères de qualité sous forme de pondérations vus précédemment, ce type de seuil ne permet pas de quantifier la distorsion perceptuelle introduite. Néanmoins, il indique le niveau de distorsion maximal JND (Just Noticeable Difference) acceptable afin que le changement sur l'image ne soit pas visible. Au-dessous de ce seuil, la modification ne pourra pas être perçue, mais au-dessus elle pourra être remarquée.

Les seuils de perception imposent des problèmes de contraintes, rester au-dessous de ce seuil ne permet donc pas d'introduire une marque de forte énergie, et la robustesse est une propriété primordiale pour la plupart des types de marquage.

L'utilisation de seuils de perception rend difficile la conciliation entre distorsion perceptuelle et énergie insérée. Watson et al. [58] a précisé expérimentalement des seuils de perception du bruit pour les coefficients DWT.

2.6.3 Algorithmes de tatouage

Les algorithmes de tatouage se distinguent essentiellement par quatre points [81], [82]:

- La façon de sélectionner les différents points de l'image originale (ou blocs) qui contiendront les données du tatouage ;
- La manière de faire correspondre l'image hôte avec l'information à enfouir (relation binaire entre les bits par exemple). C'est ce que l'on appelle la modulation
- Le pré-traitement de l'information avant son enfouissement : pré-formatage ou encore redondance de l'information ;
- Le choix du domaine de travail : spatial ou fréquentiel.

Nous analyserons d'une part des exemples de type d'algorithme travaillant sur le domaine spatial, en notant notamment leurs principaux avantages et défauts, puis nous détailleront des algorithmes opérant sur le domaine fréquentiel (DFT, DCT, DWT).

2.6.3.1 Domaine spatial

a. Tatouage LSB

Il s'agit certainement de la méthode la plus basique de la dissimulation d'informations. En reprenant la définition de la valeur d'un pixel nous savons donc que pour les images en teinte de gris cette valeur varie de 0 à 255 correspondants à différents niveaux de gris (0 étant le Noir et 255 le Blanc). Chaque pixel est donc codé sur 8 bits. Si nous considérons le fait qu'il est imperceptible pour l'œil humain un changement une variation d'une unité de gris, nous pouvons raisonnablement considérer que le dernier bit (bit de poids faible) n'est pas important, donc que nous pouvons le changer à notre guise [79].

C'est la technique utilisée pour cacher par exemple une image binaire (noir et blanc) dans une image en nuance de gris, en ne reprenant simplement que le dernier bit de chaque pixel. Pour les images en couleurs, il suffit de travailler sur la luminance.

Cette méthode ne présente néanmoins aucun des critères abordés précédemment [82], [88]:

- Robustesse : Il est très simple d'enlever ce marquage en mettant par exemple à 0 tous les bits de poids faible. De plus, tous les types de transformations fréquentielles, tels des filtres, sont radicaux pour ce marquage. Entre autres, la compression JPEG ne lui laisse quasiment aucune chance.
- Visibilité : Contrairement, à ce que l'on peut penser, l'œil humain est très sensible aux contrastes dans les gris de faibles intensités et beaucoup moins dans les teintes proche du blanc.

b. Moyenne de blocs

L'image est divisée en blocs de pixels. La moyenne de ces blocs est incrémentée pour coder un « 0 » et décrémentée pour coder un « 1 » (ou vice versa). Ce code est dit bidirectionnel. Il est aussi possible d'incrémenter la moyenne d'un bloc (4 ou 8 connexité) pour coder un « 1 » et la laisser inchangé pour coder un « 0 ». Ce code est alors dit unidirectionnel.

L'emplacement des blocs dans l'image peut être fixe ou déterminée par une clé secrète. La détection nécessite l'image originale non tatouée à laquelle est soustraite l'image tatouée.

c. Algorithme « Patchwork »

Pour renforcer un peu plus la robustesse de la méthode précédente, une idée basique, proposée par Bender & Al en 1995 [114], consiste à répéter le même bit un grand nombre de fois pour qu'une étude statistique nous donne le bit marqué.

Toujours dans le domaine spatial, cette amélioration reste néanmoins relativement faible : il est très facile de vérifier qu'une image est tatouée. En effet, bien que faisant partie des tatouages « invisibles », une étude statistique des bits de poids faible nous renseigne sur l'existence du watermark [97].

Algorithme 2.02 : Insertion Patchwork

Voyons à présent les étapes constituant cet algorithme

1. Sélectionner grâce à une clé générée aléatoirement des séquences de **n** paires de **pixel**.

2. Modifier la luminance de chaque paire (p_i, q_i) en (p'_i, q'_i) de cette façon :

$$\begin{cases} p'_i = p_i + 1 \\ q'_i = q_i + 1 \end{cases}$$

Algorithme 2.03 : Extraction Patchwork

1. Récupérer d'une part tous les n pairs grâce à la clé secrète

2. Calculer S

$$S = \sum_{i=1}^{n} (p'_i - q'_i)$$

Pour n suffisamment grand, l'équation suivante est vérifiée :

$$\sum_{i=1}^{n} (p_i - q_i) = 0$$

Seul un utilisateur possédant la clé secrète obtiendra un score S différent de 0. La clé permet ici par conséquent la localisation de zones secrètes ou la donnée sera cachée.

d. Etalement de Spectre (Spread-spectrum)

L'étalement de spectre est une technique utilisée dans les télécommunications radio, notamment par les militaires, pour disperser un signal sur une large bande de fréquence, de façon à le rendre discret et résistant aux interférences. Au cours des paragraphes précédentes, seules les idées de base imposant une relation entre d'une part le message et d'autre part l'image ont été présentées.

Nous allons décrire à présent une nouvelle approche, proposée par F. Hartung et Al qui se base sur un pseudo pré-formatage de la donnée à enfouir en « l'étalant » au niveau de la taille de l'image. Il génère ensuite une clé aléatoire de la taille de la donnée pré-formatée, puis applique, en terme simpliste, un opérateur binaire « XOR » de cette clé et de la donnée étalée.

Il suffit d'ajouter le résultat obtenu à notre image pour obtenir une image marquée. Le principe en lui même de spread-spectrum est utilisé sur un domaine DCT par l'algorithme de Hartung [69], mais il peut très bien être appliqué dans différents domaines, tels le spatial, ou encore des domaines compressés.

Algorithme 2.04 : Insertion Bender et Al.

1. Etant donné un signal v_i

2. Etant donné une séquence binaire $a_j \in \{-1, +1\}$ à cacher

   - Etaler ou plus exactement sur-échantillonner la séquence a_j d'un facteur « cr » afin d'obtenir une séquence b_i (que nous supposerons ici de la même longueur que v_i pour des raison de simplicité)

   - Amplifier la séquence b_i d'un facteur α ; puis la moduler avec un bruit pseudo aléatoire (ce bruit sert de clé secrète) $p_i \in \{-1, +1\}$ afin d'obtenir la marque suivante

   $$w_i = \alpha\, b_i\, p_i$$

   - L'image tatouée est obtenue par addition des deux signaux : image originale et marque précédemment mise en forme,

   $$v'_i = v_i + w_i$$

Algorithme 2.05 : Extraction Bender et Al.

1. Calculer la séquence s, en démodulant l'image tatouée à l'aide du bruit,

$$s_j = \sum_{cr} p'_i v_i = \sum_{cr} p_i (v_i + w_i)$$

$$= \sum_{cr} p_i w_i = \alpha\, b_i$$

$$= cr\ \alpha\ b_i = cr\ \alpha\ a_j$$

Note : Afin que l'hypothèse $\sum_{cr} p_i v_i = 0$ soit vérifiée au mieux, l'auteur propose d'extraire la marque à partir d'une version filtrée $v"_i$ de v'_i.

2. Chaque a'_j est donné ensuite par le signe de s_j

e. Etalement de Spectre multi-couches

L'insertion et la détection dans le domaine CDMA suivent dans la majorité des algorithmes le schéma classique décrit ci-dessous :

- D'une part on génère une Séquence Binaire Pseudo Aléatoire S à l'aide d'une clé secrète, uniquement connue du propriétaire. Cette séquence est composée uniquement de $+1$ et de -1 et a une moyenne nulle.

- Le message binaire à insérer (par exemple $M = \{1,\ 0,\ 0\}$) est ensuite modulé par la Séquence S. Pour une image 12×12, il faudra donc une séquence S de 144 échantillons.

- La détection se fait le plus souvent par corrélation : en effet la marque ayant une moyenne nulle, on peut considérer que l'intercorrélation de la marque avec l'image est négligeable par rapport à l'autocorrélation de marque. Pour détecter la signature, il suffit donc de calculer l'intercorrélation de la marque avec l'image marquée. Ce calcul se fait simplement en multipliant pixel par pixel les deux images et en faisant ensuite la somme des produits. Le processus est répété pour chaque bit inséré pour obtenir à la fin le message détecté.

Appelons I l'image initiale, W la marque, W_1 une marque différente et I_W l'image marquée et supposons que toutes ces images sont de taille 100×100. La détection suit alors le schéma ci-dessous :

- On forme l'image initiale marquée : $I_W = I + W$
- On calcule l'intercorrélation :

$$\langle I_W, W \rangle = \langle I + W, W \rangle = \langle I, W \rangle + \langle W, W \rangle = \varepsilon + 10000$$

$Avec\ \varepsilon = 10000$

- Pour une marque différente on aurait :

$$\langle I_W, W \rangle = \langle I + W, W_1 \rangle = \langle I, W_1 \rangle + \langle W, W_1 \rangle = \varepsilon + \varepsilon$$

$Et\ \ \varepsilon + \varepsilon \ll \langle I_W, W \rangle$

On prend donc une décision sur la présence ou non d'une signature si l'intercorrélation est supérieure à un seuil préalablement fixé.

2.6.3.2 Domaine Fréquentiel

Constatant la mauvaise performance des algorithmes de tatouage dans le domaine spatial vis-à-vis de certaines transformations, la plupart du temps étant involontaire comme entre autres de la compression JPEG, de nombreuses méthodes ont été développées à partir de connaissance acquise en traitement de signal.

Une bonne partie de ces méthodes travaille sur le domaine DCT, espérant surtout renforcer la robustesse du tatouage sur les compressions utilisant ce type de domaine. Le domaine DFT offre aussi une bonne performance car le tatouage dans ce domaine résiste aux rotations et aux translations.

a. Tatouage dans le domaine DFT

- Calcul de la DFT

La transformée DFT d'un signal bidimensionnel et son inverse sont données respectivement par les relations suivantes :

$$dft(i,j) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} p(x,y)e^{-j\left(\frac{2\pi}{M}\right)im}e^{-j\left(\frac{2\pi}{N}\right)jn}$$

$$i = 0,1,\dots,M-1\ et\ j = 0,1,\dots,N-1$$

Où :

- $p(x,y)$ désigne la valeur du pixel de coordonnées (x,y)
- $dft(i,j)$ le coefficient de la transfert de Fourier
- $M \times N$ la taille de l'image
- $p(x,y) = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{y=0}^{N-1} dft(i,j)e^{-j(2\pi/M)im}e^{j(2\pi/N)jn}$
- $x = 0,1,\dots,M-1\ et\ y = 0,1,\dots,N-1$

- Insertion

Les données de la marque sont insérées directement dans le spectre de l'image.

L'insertion s'effectue généralement dans les moyennes fréquences. Les basses fréquences ne sont pas modifiées pour éviter les dégradations visuelles trop importantes. Les données ne sont pas non plus insérées dans les hautes fréquences pour respecter la robustesse du système.

Selon Ruanaidh [115], le filigrane w est inséré dans le spectre de module de la DFT. Pour assurer un bon compromis entre la résistance maximale aux attaques et une dégradation minimale de l'image, on fixe une bande dans les fréquences moyennes. La séquence spread spectrum est alors ajoutée aux valeurs des amplitudes de la bande choisie, la phase étant laissée inchangée. Pour une plus grande sécurité les positions des amplitudes affectées sont choisies en fonction de la clef. La DFT ainsi modifiée est finalement inversée pour obtenir l'image tatouée. La force du filigrane peut être choisie interactivement, ou bien calculée d'une manière adaptative en fonction de la moyenne et la variance des amplitudes dans la bande de fréquences choisie.

Pour extraire le filigrane, l'on considère le module de la DFT de l'image tatouée. Étant donné que les positions des coefficients de la bande de fréquences auxquelles les composants du filigrane, ont été ajoutés, sont connues, l'on peut extraire la séquence spread spectrum : $w' = w + e,$ où w est le spread spectrum inséré et e un bruit additif. Le message m est alors décodé comme suit. Pour décoder le bit i du message m à partir de w', on calcule le produit scalaire (qui correspond à une corrélation) entre $w' = \sum_{i=1}^{M} b_i v_i + e$ et v_i

$$B'_j = \langle w', v_j \rangle = \sum_{i=1}^{M} b_i \langle v_i v_j \rangle + \langle e, v_j \rangle$$

b. Tatouage dans le domaine de la DCT

La DCT a longtemps été la méthode privilégiée de compression d'images numériques, de ce fait elle a été utilisée dans la norme JPEG (Joint Pictures Experts Group). Mais elle possède aussi des propriétés intéressantes qu'on peut exploiter en tatouage d'image.

- Principe de la DCT

Dans le cas d'images fixes, la DCT permet d'analyser un signal ne variant plus en fonction du temps, mais bien en fonction de la position du pixel dans le pseudo-plan de l'écran ; c'est une transformation de même nature que la transformée de Fourier, mais opérant en fonction des coordonnées x et y du pixel. La DCT permet de décomposer le signal lumineux composant l'image (x et y pour les coordonnées du pixel, p pour sa définition lumineuse) selon une combinaison de fonctions trigonométriques, identifiables par leur fréquence et leur amplitude. Elle traduit une information spatiale en une information fréquentielle (spectrale) [97].

Comme la DFT, la DCT est une transformation entièrement réversible (IDCT, inverse de la DCT restitue fidèlement les données initiales).

Pour un bloc d'image de 8x8 pixels, la DCT produit elle même une matrice de 64 coefficients (8x8), rangés selon les fréquences croissantes selon les axes horizontal et vertical.

Généralement, la plus grande partie de l'énergie est concentrée dans les basses fréquences du spectre qui sont exprimées par les coefficients les plus proches du coin supérieur gauche de la matrice. C'est cette propriété qu'on exploite en tatouage : on insère la marque dans les basses ou moyennes fréquences selon un compromis robustesse / imperceptibilité.

- Calcul de la DCT

Si x et y désignent les dimensions spatiales de l'image, i et j les dimensions dans le domaine des fréquences de l'image, N le nombre d'échantillons en x et y (N=8 en général), p l'image (ou matrice) originale et dct l'image transformée, la DCT fait correspondre à chaque valeur de $p(x,y)$ une valeur de $dct(i,j)$ donnée par la formule :

$$dct(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) cos\left[\frac{(2x+1)i\pi}{2N}\right] cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$\left(o\grave{u}\ C(u) = \frac{1}{\sqrt{2}}\ si\ u = 0; C(u) = 1\ si\ u \neq 0\right)$$

Où :

- $p(x,y)$ désigne la valeur du pixel de coordonnées (x,y)
- $dct(i,j)$ le coefficient repéré par la ligne i et la colonne j dans la matrice DCT
- N la largeur de l'image

La transformation inverse s'effectue alors grâce à la formule suivante :

$$p(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)dct(i,j) cos\left[\frac{(2x+1)i\pi}{2N}\right] cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

La matrice transformée par DCT regroupe les valeurs les plus élevées dans le coin supérieur gauche de la matrice (basses fréquences) et les valeurs les plus faibles dans le coin inférieur droit (hautes fréquences). Ainsi le maximum d'informations sur l'image se trouve concentré sur la partie supérieure gauche de la matrice.

- Principe du tatouage

Insertion

1. On calcule la transformée DCT (ou la transformée DCT d'un bloc 8x8) de l'image ;
2. On ajoute la marque dans l'image transformée en modifiant certains coefficients de la transformée.
3. La transformation inverse est appliquée à la transformée marquée pour générée l'image marquée.

Extraction

On calcule la DCT inverse de l'image tatouée et on extrait la marque grâce à la clé.

Détection de la marque

Comme dans le domaine spatial, la détection se fait en calculant la corrélation

Algorithme 2.06 : Insertion Koch et Zhao

Une approche consiste à extraire un certain nombre de carré de 8x8 pixels de l'image, de calculer la transformée DCT de ces blocs et d'aller marquer un bit sur les moyennes fréquences correspondantes, sachant que la modification des basses fréquences de l'image la changerait trop, les basses fréquences correspondant aux zones homogènes les plus grandes sur l'image, par exemple un noir uniforme dans les zones sombres, et que les hautes fréquences sont enlevées par la compression JPEG, correspondant aux zones homogènes les plus petites d'une image, à savoir les détails au niveau de chaque pixel.

Cette méthode a été proposée par E. Koch et J. Zhao en 1995. Elle est à la base de nombreux travaux, notamment dans le cadre de projets Européens comme Talisman. Voici une description formelle des algorithmes [116] d'insertion et d'extraction de cette méthode.

1. Soit une séquence de k bits $(b_1, \dots, b_k)$ à cacher dans l'image
2. Sélectionner dans l'image (selon une clé secrète) k blocs $B(B_1, \dots, B_k)$ de taille (8×8)

3. Calculer les coefficients DCT $(a_{11}, \dots, a_{88})$ de chaque bloc sélectionné

4. Pour i allant de 1 à k :

- Soient (a_{kl}) et (a_{mn}) deux des coefficients DCT du bloc B_i, et b_i le bit à cacher

- Si $\{(b_i = 1) et\ (a_{kl})_i > (a_{mn})_i\}\ ou\ \{(b_i = 0) et\ (a_{kl})_i < (a_{mn})_i\}$ alors ne rien faire

- Sinon modifier les valeurs pour que la relation précédente soit vérifiée.

5. Calculer la DCT inverse à partir des valeurs ainsi modifiées afin d'obtenir l'image marquée, et revenir dans le domaine spatial, si besoin est.

Algorithme 2.07 : Extraction Koch et Zhao

1. Retrouver les blocs marqués grâce à la clé secrète.

2. Calculer les coefficients DCT associés aux blocs sélectionnés.

3. Comparer les valeurs des coefficients DCT afin de déterminer si le bit concerné du message était un « 0 » ou un « 1 ».

Cox et Al proposent une méthode appliquant la DCT à toute l'image et insèrent la marque dans les basses fréquences. Ils modifient les fréquences de plus grandes amplitudes selon les formules suivantes :

$$v'_i = v_i + \alpha\, x_i$$
$$v'_i = v_i(1 + \alpha\, x_i)$$
$$v'_i = v_i e^{\alpha\, x_i}$$

Avec :

- v_i : coefficients de l'image originale
- v'_i : coefficients de l'image marquée
- α : coefficient d'indivisibilité
- x_i : coefficient réel issu d'une distribution gaussienne centrée normée (pour insérer la marque dans une zone prédisposée de l'image).

Bien qu'étant nettement plus robuste à des attaques involontaires de type fréquentiel, cet algorithme présente néanmoins quelques inconvénients :

- Le fait de cacher l'information dans des blocs permet au mieux de stocker un bit dans ces blocs, donc limite le ratio.

- Le fait même d'utilisé des blocs a toujours cet inconvénient d'être vite mis en difficulté face a des attaques géométrique. Le simple fait d'appliquer une rotation modifie le maillage d'origine de l'image, donc la segmentation en bloc de 8x8 ne correspond plus du tout à l'originale.

- Enfin le terme de « moyennes fréquences » est particulièrement dur à définir, ce qui entraîne assez vite un conflit visibilité / robustesse : si l'on choisit des fréquences relativement basses, le tatouage est certes plus robuste, mais devient visible, parallèlement plus nous prenons des fréquences hautes plus le marquage se fond dans l'image, mais perd en robustesse (même face à une compression JPEG).

Cet algorithme a subi de nombreuses modifications pour essayer de palier à ces problèmes. Entre autres le compromis robustesse / visibilité a été particulièrement affiné.

A noter aussi la robustesse de cette approche a engendré de nombreuses méthodes bien plus performantes telles que le tatouage dans le domaine de l'ondelette.

c. Tatouage dans le domaine de la transformée en ondelettes

La transformée en ondelettes a déjà fait ses preuves dans le domaine de la compression d'images, ce qui a motivé certains chercheurs qui ont voulu inventer de nouvelles méthodes de tatouage plus robuste que la DCT.

Presque tous les algorithmes de tatouage opèrent dans le domaine fréquentiel, qui permet d'accéder à toutes les composantes fréquentielles de l'image hôte. Le chapitre 3 parlera de ce type de tatouage.

2.6.4 Manipulations et attaques sur les images

Il existe des manipulations appliquées sur les images qui peuvent apporter des distorsions importantes au message incrusté et changer son comportement. Ces manipulations peuvent être vues comme des attaques malveillantes (suppression, modification ou détérioration du message) ou simplement innocentes afin d'optimiser la qualité (filtres) ou l'enregistrement (compression) de l'image. Pendant longtemps les recherches dans le domaine du marquage ont été focalisées sur la protection des droits d'auteur et dont la robustesse est la propriété la plus importante. La marque doit être robuste à certains types de manipulations habituellement utilisées dans l'imagerie numérique. Nous allons présenter dans cette section une liste non exhaustive de ces manipulations [90], [93], [94].

a. Compression

La compression avec pertes est le mode de compression le plus utilisé. Elle a pour but de diminuer la taille du fichier image. Les techniques de compression avec pertes suppriment les informations redondantes des images. Comme la marque n'est pas généralement visible, elle peut donc être considérée comme non significative et donc aussi être supprimée.

b. Rehaussement et lissage

Le rehaussement correspond à l'augmentation des composantes hautes fréquences de l'image. L'image devient alors plus contrastée. Le lissage est l'opération contraire du rehaussement, il atténue les composantes hautes fréquences de l'image qui devient alors plus floue. Ces opérations peuvent modifier également les composantes hautes fréquences du message et leur faire perdre leurs particularités

c. Transformations géométriques usuelles

Parmi les transformations géométriques, la plus usuelle est la modification des dimensions de l'image et les transformations affines telles que la rotation, la translation et zoom. Ce genre de transformation provoque dans la plupart des cas une désynchronisation de la marque insérée lors de la détection et l'extraction.

d. Conversions analogique-numérique

La conversion analogique/numérique peut provoquer une perte de qualité ou ajouter du bruit dans l'image. Ceci peut être obtenu à partir de l'impression suivie d'une acquisition

par scanner, ou encore à partir d'un film réalisé à l'aide d'un caméscope dans une salle de cinéma. Ce type de conversion peut apporter des déformations géométriques provoquant une perte de synchronisation.

e. Modification valumétriques

Il existe une catégorie de traitement (étalement d'histogramme, égalisation d'histogramme) qui ne prend en compte que la luminance pour améliorer l'image. Comme le changement est fait sur la luminance, les informations marquées sur la chrominance peuvent être désynchronisées.

f. Débruitage

L'objectif de cette manipulation malveillante est d'approcher au mieux la forme d'onde du message, pour pouvoir l'enlever. Le message peut être estimé en utilisant le filtrage de Wiener. Cette estimation est alors soustraite à l'image originale pour l'obtention d'une copie du message.

g. Gigue

Le gigue (Jittering) est un phénomène connu en télécommunications. Lorsque le délai de transmission du signal varie, il en résulte une réplication ou une suppression d'un morceau du signal. Ceci peut se produire dans le domaine spatial ou temporel sur les images, il peut y avoir un ajout ou une suppression de lignes ou de colonnes.

h. Attaque par mosaïques

Il s'agit de découper l'image marquée en plusieurs morceaux. Cette attaque vise les moteurs de recherche automatique (crawlers) des marques dans les images sur Internet. L'image est ainsi envoyée par morceaux et assemblée dans une page HTML.

i. Stirmark

L'attaque Stirmark est un logiciel qui permet d'apprécier la robustesse d'un procédé de marquage. Ce logiciel propose un banc de tests avec une grande variété de traitements sur les images, comme les manipulations présentées précédemment et plusieurs distorsions géométriques. La qualité de l'image résultante n'est pas dégradée, mais la marque est fortement modifiée ou effacée.

2.6.4.1 Techniques robustes aux distorsions synchrones

Les méthodes de marquage ont différents comportements selon la technique et le domaine d'insertion du message. La technique et le domaine de marquage peuvent rendre la marque plus ou moins résistante aux distorsions synchrones ou asynchrones. Une insertion par substitution apporte une robustesse plus importante mais ciblée sur un type de distorsion donnée comme par exemple une technique de compression. Dans cette section, les méthodes sont présentées en fonction de leur type et de leur domaine d'insertion.

a. Marquage additif dans le domaine spatial

Parmi les nombreux schémas spatiaux d'insertion basés sur l'ajout d'une séquence aléatoire 2D, nous distinguons la technique d'étalement de spectre de Hartung [102] et la technique du patchwork de Bender [114]. La technique du patchwork consiste à diviser l'image en deux ensembles disjoints A1 et A2 de pixels qui dépendent d'une clef secrète. Ensuite les pixels de l'image, notés $p(i,j)$, sont modifiés différemment selon l'ensemble A1 et A2 auxquels ils appartiennent.

b. Marquage additif dans le domaine fréquentiel

Les méthodes classées dans ce groupe permettent d'obtenir une bonne robustesse à la compression JPEG. Les données sont insérées soit sur les blocs de pixels transformés soit directement sur la transformée de l'image complète.

c. Marquage additif dans le domaine multirésolution

Il existe de nombreuses méthodes de marquage qui agissent dans le domaine multirésolution [104], [105]. Barni et al. ont proposé un schéma aveugle où le message est inséré dans les trois sous-bandes de détails (LH_0, HL_0, HH_0) pour avoir un meilleur résultat de la robustesse par rapport à l'invisibilité. Les coefficients des trois sous-bandes XLH_0 , XHL_0 et XHH_0 sont marqués par addition d'une séquence pseudo-aléatoire W de même taille que les trois sous-bandes.

d. Marquage substitutif dans le domaine spatial

Contrairement aux méthodes additives, qui dans l'ensemble ont des schémas similaires, les méthodes substitutives se distinguent par leur diversité. Dans le domaine spatial, l'insertion peut se faire par la modification des bits de poids faibles LSBs, par la quantification vectorielle spatiale, ou encore par l'insertion de similarités.

La modification du LSB est une des techniques les plus simples. Le message est inséré dans l'image par le remplacement du LSB d'une composante couleur du pixel par le bit du message. La quantification vectorielle consiste à remplacer des vecteurs de l'image (bloc de pixels) par des vecteurs appartenant à un dictionnaire prédéfini. Afin de restreindre l'impact sur l'image, les blocs du dictionnaire sont choisis de façon à être les plus proches possibles des blocs originaux. Les méthodes basées sur l'insertion de similarités exploitent le changement des similarités existants entre les blocs de l'image an d'insérer le message. Ainsi, des composantes de l'image sont substituées par des composantes qui possèdent une relation de similarité avec d'autres composantes [106].

e. Marquage substitutif dans le domaine fréquentiel

La norme JPEG est composée de plusieurs étapes qui peuvent être utilisées pour le marquage d'information. L'insertion du message après le codage entropique (étape de compression) nous permet d'éviter d'effectuer une décompression totale de l'image lors de l'extraction. La méthode JPEG-JSTEG [107] insère le message sur les coefficients DCT quantifiés non nuls afin que la dégradation due à la compression soit négligeable. L'idée est d'utiliser le LSB de chaque coefficient quantifié dont la valeur est strictement supérieure à 1 pour insérer le bit du message. L'inconvénient de cette approche réside dans la forte dépendance de la capacité de stockage en fonction du contenu de l'image.

f. Marquage substitutif dans le domaine multirésolution

La transformation par ondelettes admet diverses perspectives pour l'insertion de données [108]. Les ondelettes permettent aussi de tatouer une image compressée au format JPEG2000 sans pour autant avoir à décompresser totalement l'image à tatouer.

2.6.4.2 Techniques robustes aux distorsions asynchrones

Une distorsion est dite asynchrone si elle détériore le synchronisme existant entre les données insérées et l'image couverture. Le repère d'insertion initial du message est déplacé et ceci pose des problèmes lors de la détection et de l'extraction. Quand une image marquée subit une distorsion asynchrone, elle doit alors être resynchronisée. Cette opération peut être extrêmement coûteuse en temps de calcul. Les techniques de distorsions asynchrones les plus couramment utilisées sont les transformations géométriques comme rotations, translations et l'attaque stirmark précédemment [95].

a. Insertion insensible à la géométrie

Afin de résister aux attaques géométriques l'approche générale consiste à insérer le message sans faire appel à la géométrie structurelle de l'image. Entre les nombreux schémas insensibles à la géométrie, nous allons présenter les deux plus courant : la luminance et l'histogramme.

- La luminance est une composante de certains espaces colorimétriques (comme YUV par exemple) qui possède des propriétés intéressantes pour le marquage. En effet, la luminance moyenne d'une image est conservée après des distorsions asynchrones [99]. Le schéma le plus utilisé est basé sur l'étalement de spectre. L'insertion du message est faite en ajoutant une composante continue générée à l'aide d'une séquence aléatoire. La détection est obtenue en calculant la corrélation entre la luminance moyenne et la séquence aléatoire utilisée lors de l'insertion.
- L'histogramme représente le nombre d'occurrences de l'image. Il est aussi, comme la luminance, peu sensible aux déformations asynchrones. L'insertion peut être faite par la reclassification des pixels en comparant leur valeur puis la moyenne des valeurs associées à des différents voisinages. Nous pouvons aussi rendre l'histogramme périodique pour insérer l'information.

b. Insertion périodique du message

L'insertion périodique d'information permet de réduire la complexité de la détection après une distorsion asynchrone. L'espace de recherche est alors réduit à la taille de la période de base. Un calcul de corrélation permet de récupérer le message après les distorsions asynchrones éventuelles.

Toutefois, la redondance introduite rend ces méthodes facilement apparentes. De plus, elles ne sont pas robustes aux transformations géométriques non affines. Delanay et al. [109] ont suggéré une méthode qui génère un message périodique dont le support de base dépend d'une clef secrète.

c. Insertion d'un motif de resynchronisation

La stratégie ici est d'identifier directement la transformation géométrique afin de l'inverser et de détecter le message dans son repère initial. Ceci est obtenu par l'utilisation de motifs resynchronisant. Kutter [110] propose une méthode dont l'information est insérée quatre fois dans l'image. Il devient ainsi redondant spatialement et donc repérable par auto-corrélation de l'image.

d. Utilisation de l'image originale

L'utilisation de l'image originale pour la détection (approche non-aveugle) aide l'identification de possibles transformations géométriques appliquées sur l'image tatouée. Davoine et al. [111] ont proposé un schéma dont l'image originale et l'image tatouée sont manipulées pour compenser les déformations géométriques produites par stirmark. Dans leur approche, l'idée est de déplacer les différents sommets de la partition de l'image tatouée.

e. Utilisation du contenu de l'image

L'utilisation du contenu de l'image permet de construire un repère d'insertion spécifique qui subit les mêmes transformations asynchrones que l'image. Comme ce repère est lié au contenu de l'image il remporte une robustesse additionnelle.

Il existe plusieurs approches pour la description du contenu de l'image. Elle peut être faite par détection des contours ou de régions [112] ou encore par détection de points d'intérêts particuliers. Lovarco et al. [113] ont présenté une approche de descripteurs basée sur la détection de régions et sur le calcul du centre de gravité des objets de l'image. Les objets dans l'image sont identifiés et les coordonnées verticales et horizontales du barycentre de chaque élément sont calculées.

Pour identifier la forme des objets, deux vecteurs propres indiquant les axes majeur et mineur sont calculés en utilisant une méthode dérivée de l'analyse en composantes principales. Ces vecteurs forment le repère pour l'insertion du message. Le message est inséré en blocs unitaires localisés autour de ces vecteurs plusieurs fois dans les différents espaces couleurs.

2.7 Conclusion

Ce chapitre donne un panorama sur le multimédia comme étant un moyen physique par lequel les données sont perçues, représentées, stockées ou transmises. On a vu les différentes méthodes de codages qu'on peut utiliser pour comprimer les flux multimédia telles que le son, les images et la vidéo. Cette compression peut se faire avec ou sans perte d'information et ayant le but de faciliter la transmission et le stockage de ces données. Des exemples de codage d'image ont été présentés du fait qu'on va axer une partie de notre recherche sur le transfert sécurisé d'image.

Dans un deuxième temps, on a fait une synthèse des techniques de dissimulation de données en particulier l'Insertion de Données Cachées ou IDC qui englobe plusieurs manières de cacher des informations à l'intérieur d'autres informations multimédia comme le tatouage et la stéganographie et tout cela dans le cadre de la sécurisation d'information.

CHAPITRE 3
ALGORITHMES DE TRANSFERT SECURISE D'INFORMATION

3.1 Introduction

Ce chapitre fait l'objet de deux publications dans une revue de niveaux nationaux MADA-ETI et deux publications dans des revues de niveaux internationaux dont, IEEE Xplore – USA et International Journal of Future Computer and Communication :

- « *Performance des cryptosystèmes basés sur les courbes elliptiques* », MADA-ETI, ISSN 2220-0673, Vol.2, pp.1-9, 2010, www.madarevue.gov.mg
- « *Optimisation des algorithmes de calcul cryptographique basés sur les Courbes Elliptiques* », MADA-ETI, ISSN 2220-0673, Vol.1, pp.1-7,2013.
- « *Modélisation Mathématique d'un appel SIP* », IEEE Xplore ISBN: 978-1-4673-1520-3 pp : 48-50, 2012
- « *Modélisation Mathématique d'un serveur VOIP* », International Journal of Future Computer and Communication, vol1 N°1,2012/ISSN 2010-3751 pp : 26-28.

Les versions originales en anglais de ces articles sont présentées en annexe 1 et annexe 2.

Ce chapitre constitue les modèles mathématiques que l'on va proposer et exploiter pour permettre et optimiser une sécurisation lors d'un transfert d'information. Les chapitres précédents nous ont informés sur les recherches et techniques existantes faites par d'autres chercheurs. Après des études plus approfondies sur ces dernières, nous avons pu constater quelques inconvénients et améliorations à faire sur leurs approches. Nous consacrons ce chapitre sur l'étude de novelles approches et techniques tout en apportant notre contribution.

3.2 Optimisation des algorithmes cryptographiques

3.2.1 Introduction

L'optimisation d'un programme ou algorithme cryptographique rend ce dernier encore plus performant. En effet, les méthodes de programmation diffèrent selon le type d'information à protéger ainsi que son environnement d'implémentation dans les systèmes informatiques.

Ce chapitre constitue notre apport et notre contribution dans notre recherche en proposant des nouveaux modèles mathématiques et algorithmes cryptographiques se basant sur des modèles déjà existants [117], [118]

3.2.2 Modélisation mathématique des algorithmes cryptographiques

3.2.2.1 L'algorithme AES-I (AES-Improved)

L'algorithme AES ou Rijndael sont définies soit au niveau de l'octet, soit sur des mots de 32 bits. Un octet est représenté comme un élément de corps fini $\mathbb{F}_{2^8}$ (comme les codes cycliques) tandis que les mots sont représentés par des polynômes à coefficient dans $\mathbb{F}_{2^8}$.

Nous allons donc adopter les modèles d'opération suivants pour notre algorithme AES-I (Advanced Encryption System Improved) :

a. Opération dans $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/m(x)$

On travaille dans le corps d'extension de $\mathbb{F}_2$ défini au moyen du polynôme $m(x)$ qui est un irréductible sur $\mathbb{F}_2$. Le polynôme $m(x) = 1 + x + x^3 + x^4 + x^8$ peut être représenté en binaire à l'aide de l'inverse de la représentation polynomiale par 100011011 de manière équivalente, par sa valeur hexadécimale $(11B)_H$ [117].

Dans cette structure, un élément de base est un polynôme de $\mathbb{F}_2[x]$ de degré au plus 7. $b_7x^7 + b_6x^6 + \cdots + b_1x + b_0$ représente un octet [117], [118].

- Addition : par addition des coefficients. Par exemple,

$$\begin{gathered}(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2 \\ \leftrightarrow (57)_H + (83)_H = (D4)_H\end{gathered} \tag{3.01}$$

- Multiplication : multiplication des polynômes $modulo\ m(x)$. Par exemple,

$$\begin{gathered}(x^6 + x^4 + x^2 + x + 1).(x^7 + x + 1) \equiv x^7 + x^6 + 1\ mod\ m(x) \\ \leftrightarrow (57)_H.(83)_H = (C1)_H\ mod\ m\end{gathered} \tag{3.02}$$

- Multiplication par x : c'est une opération de décalage des bits suivi d'un ou exclusif avec $m(x)$ si le coefficient du monôme de degré 8 est 1

b. Codage d'un mot de 32 bits dans $\mathbb{F}_{2^8}[x]$

Un mot de 32 bits est représenté par un polynôme de degré au plus 3 dans l'anneau $\mathbb{F}_2[x]/(\mathrm{x}^4+1)$.

Ainsi on peut utiliser les opérations suivantes [117]:

- L'addition usuelle de deux polynômes de degré au plus 3 se fait par addition des coefficients

- La multiplication usuelle $c(x) = a(x).b(x)$ de polynômes a pour inconvénient de rendre éventuellement un polynôme $c(x)$ de degré supérieur à 3. Aussi faut-il considérer une multiplication modulaire des polynômes $a\ et\ b\ modulo\ M(x) = x^4 + 1$. En fait, dans AES, on ne considère que des produits par un polynôme a fixé.

- Multiplication modulaire par $a(x)\ modulo\ M(x) = x^4 + 1$.

$c(x)\ mod\ M(x) = d(x) = a(x) \otimes b(x)$, pour a fixé, peut être réalisé au moyen d'une matrice circulante :

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{3.03}$$

Comme $x^4 + 1$ n'est pas un irréductible de $\mathbb{F}_{2^8}$, la multiplication par a n'est pas nécessairement inversible. Dans AES, le polynôme fixé $a(x)$ est inversible.

c. Etapes de chiffrement proposées de l'AES-I

AES est un chiffre itéré de taille de blocs et de clés à valeurs dans $\{128,192,256\}$ bits. Les différentes transformations utilisées travaillent sur un résultat appelé *état* qui est représenté par un tableau d'octets à des dimensions à 4 lignes et Nb colonnes. Les clés de tour admettent une représentation analogue à 4 lignes et Nk colonnes. Le texte en clair et la clé sont rangés octets par octets dans les colonnes des tableaux.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

***Tableau 3.01** : Rangement du texte en clair et de la clé*

Ces tableaux peuvent être transformés en tableau de dimension 1 indicé de 0 à $4Nb - 1$ pour le premier et de 0 à $4Nb - 1$ pour le second en les lisant colonne par colonne (pour l'entrée et la sortie). Le nombre de tour Nr est une fonction de la taille des blocs et de la taille de la clé :

Nr	$Nb = 4$	$Nb = 6$	$Nb = 8$
$Nk = 4$	10	12	14
$Nk = 6$	12	12	14
$Nk = 8$	14	14	14

***Tableau 3.02** : Nombre de tour*

- Transformations de tour

Avant le premier tour, on effectue une opération de ou exclusif avec les bits de la clé de tour (*AddRoundKey*).

A chaque tour, en fonction d'une clé de tour obtenue au moyen d'un algorithme de séquencement, décrit plus loin, on effectue les opérations suivantes sur l'état :

- *ByteSub* : utilisation « d'une boîte –S » qui travaille octet par octet
- *ShiftRow :* on applique une permutation circulaire sur les lignes de l'état. Sa valeur dépend du numéro de la ligne et du nombre de tours Nr. Il est à noter que la première ligne (de numéro 0) ne subit pas cette permutation. Le tableau ci-dessous donne la valeur des décalages.

Nr	$C1$	$C2$	$C3$
4	1	2	3
6	1	2	3
38	1	3	4

***Tableau 3.03 :** Valeur des décalages*

- *MixColumn :* on multiplie chaque colonne de l'état vue comme un polynôme par le polynôme fixé $a(x) = 03_H x^3 + 01_H x^2 + 01_H x + 02_H \; mod \; x^4 + 1$ au moyen d'une matrice circulante :

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 01 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \tag{3.04}$$

  L'inverse de $a(x)$ est le polynôme $d(x) = 0B_H x^3 + 0D_H x^2 + 09_H x + 0E_H$.

- *AddRoundKey* : on effectue un *xor* entre les bits de l'état et ceux de la clé de tour.

Au dernier tour on omet la multiplication par $a(x) mod \; x^4 + 1$ pour rendre le déchiffrement plus aisé.

- Description des boîtes-S

Toute la sécurité des chiffres de ce type réside dans la conception des boites-S. Nous décrivons ci-dessous le fonctionnement de celles qui sont utilisées dans AES-I :

- On calcule l'inverse multiplicatif de chaque octet dans $\mathbb{F}_{2^8}$

- On applique la transformation affine suivante sur chaque octet résultant de l'opération précédente :

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \tag{3.05}$$

Notons que l'opération inverse (pour le déchiffrement) est une substitution sur les octets : on applique la transformation affine inverse puis on calcule l'inverse multiplicatif du résultat.

- Algorithme de séquencement de clé

Les différentes clés de tour sont obtenues par dérivation de la clé principale au moyen de l'algorithme de séquencement de la clé qui se déroule en deux étapes :

- Une phase d'expansion de la clé
- Une phase de la sélection de la clé de tour

Algorithme 3.01 : Le principe de base de l'algorithme est le suivant :

- Le nombre de bits pour la clé de tour est égal à la longueur du bloc multiplié par le nombre de tour plus un (par exemple, pour une taille de bloc de 128 bits et 10 tours, il faudra 1408 bits)
- La clé principale est étendue en une clé étendue
- Les clés de tour sont issues de la clé étendue de la façon suivante : la première clé de tour consiste en les Nb premiers mots de la clé étendue, la seconde en les Nb suivants et ainsi de suite.

- Expansion de la clé

La clé étendue est un tableau à une dimension de mots de 32 bits noté $W[Nb * (Nr + 1)]$ dont les Nk premiers sont la clé principale key. Les autres mots sont obtenus récursivement à partir des mots précédents. La fonction d'expansion de la clé dépend de la valeur Nk. Il y en a en fait deux fonctions d'expansion : une lorsque $Nk \leq 6$ et une autre lorsque $Nk > 6$.

- **Expansion de la clé quand $Nk \leq 6$:**

Algorithme 3.02

$KeyExpansion(key[4 * Nk], W[Nb * (Nr + 1)])$

$pour\ i\ de$ 0 à $Nk - 1\ faire$

$W[i] \leftarrow (Key[4 * i], Key[4 * i + 1], Key[4 * i + 2], Key[4 * i + 3])$

$fpour$

$pour\ i\ de\ Nk$ à $Nb * (Nr + 1) faire$

$temp \leftarrow W[i-1]$

$si\ i\ mod\ Nk = 0\ alors$

$temp \leftarrow SubByte(RotByte(temp)) \otimes Rcon[i/Nk]$

fsi

$W[i] \leftarrow W[i-Nk] \otimes temp$

$fpour$

Dans cet algorithme, la fonction *SubByte(W)* renvoie un mot de 32 bits composé de quatre octets. Chaque octet est issu de l'application de la « *boîte-S* » à l'octet de la position correspondante du mot d'entrée. La fonction *RotByte(W)* retourne un mot de 32 bits après avoir appliqué la permutation circulaire$(a, b, c, d) \mapsto (b, c, d, a)$.

- **Expansion de la clé quand $Nk > 6$**

Algorithme 3.03 :

$KeyExpansion(key[4 * Nk], W[Nb * (Nr + 1)])$
$pour\ i\ de$ 0 à $Nk - 1\ faire$
$W[i] \leftarrow (Key[4 * i], Key[4 * i + 1], Key[4 * i + 2], Key[4 * i + 3])$
$fpour$
$pour\ i\ de\ Nk$ à $Nb * (Nr + 1) faire$
$temp \leftarrow W[i-1]$
$si\ i\ mod\ Nk = 0\ alors$
$temp \leftarrow SubByte(RotByte(temp)) \otimes Rcon[i/Nk]$
$sinon\ si\ i\ mod\ Nk = 4\ alors$
$temp \leftarrow SubByte(temp)$
fsi
fsi
$W[i] \leftarrow W[i-Nk] \otimes temp$
$fpour$

La constante *Rcon* est indépendante de Nk. Elle est définie par :

$$Rcon[i] = (RC[i], 00_H, 00_H, 00_H)$$

Où $RC[i]$ représente un élément de $\mathbb{F}_{2^8}$ défini inductivement par :

$$RC[1] = 1 = 01_H$$

$$RC[i] = x.RC[i-1] = 02_H.RC[i-1] = x^{(i-1)}$$

- Sélection de la clé de tour

La clé du i-ème tour est le contenu de W entre les positions $W[Nb * i]$ et $W[Nb * (i+1)]$, comme illustré par la figure suivante :

W_0	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}
Clé de tour 0						Clé de tour 1					

Tableau 3.04 *: Sélection de la clé de tour*

- Déchiffrement

Rijndael a été conçu pour fonctionner de la même manière au chiffrement qu'au déchiffrement en remplaçant chaque opération élémentaire par son inverse au seul prix d'un changement de l'algorithme d'expansion de la clé qui devient :

- Appliquer l'algorithme normal d'expansion de la clé
- Appliquer *InvMixColumn* à toutes les clés de tour hormis pour la première et pour la dernière, où *InvMixColumn* représente le produit par le polynôme $d(x)$, inverse multiplicatif du polynôme fixé $a(x)$

3.2.2.2 RSA

Pour le chiffrement RSA, Rivest, Shamir et Adleman ont proposé la méthode [118]:

1) On choisit deux entiers premiers assez grands $p\ et\ q$, de l'ordre de 10^{100}
2) On fixe $n = p * q$ et on publie n
3) On calcule $\varphi(n) = (p-1)(q-1)$

4) On choisir un nombre entier e tel que $e > 2$ et $pgcd(e, \varphi(n)) = 1$ et on le publie

5) On calcule d tel que $d * e \equiv 1\ mod\varphi(n)$

On chiffre alors un message M en calculant

$$M^e * mod(n) = \mathrm{C} \tag{3.06}$$

pour un entier M tel que $0 < M < n$ et on déchiffre selon :

$$C^d * mod(n) \equiv (\mathrm{M^e})^{\mathrm{d}} mod(n) \equiv M^{1+k\varphi(n)} mod(n) \equiv M\ mod(n) \tag{3.07}$$

La sûreté de ce système repose sur l'hypothèse RSA pour laquelle il est aussi difficile de décrypter que de factoriser n, ce qui est un problème très difficile.

- Exponentiation modulaire

Pour implémenter RSA, on a besoin d'un algorithme d'exponentiation modulaire rapide pour calculer $a^b mod(n)$ [117], [118]. On propose ci-dessous un algorithme, qu'on a optimisé, qui travaille en temps proportionnel au logarithme de b : pour ce faire, on élève successivement au carré a, a^2 ...

Soit $\langle b_k, b_{k-1}, \dots, b_0 \rangle$ la représentation binaire de b telle que $b = \sum_{i=0}^{k} b_i 2^i$. La procédure suivante calcule $a^b mod(n)$.

Algorithme 3.04 :

$GrandModulo(a, b, n)$
$c \leftarrow 0; d \leftarrow 1;$
$Soit\ \langle b_k, b_{k-1}, \dots, b_0 \rangle\ la\ représentation\ binaire\ de\ b$
$pour\ i \leftarrow k\ jusqu'$à $0\ pas - 1\ faire$
    $c \leftarrow 2.c; d \leftarrow (d.d) mod(n);$
    $si\ b_i = 1\ alors$
        $c \leftarrow c + 1; d \leftarrow (d.a) mod(n);$
    fsi
$fpour$
$Renvoie\ d$

Chaque exposant c calculé est soit le double de l'exposant précédent soit le double de la valeur de l'exposant précédent plus 1. La représentation binaire de $b$est lue de la droite cers la gauche afin de contrôler quelles opérations doivent être effectuées ; chaque itération utilise une des deux identités suivantes :

$$a^{2c}\, mod(n) = (a^c)^2\, mod(n)$$
$$a^{2c+1}\, mod(n) = a.(a^c)^2\, mod(n)$$

selon la valeur de b_i qui est respectivement 0 ou 1.

L'élévation au carré est la seule opération effective des l'itération. La variable c permet d'expliquer le fonctionnement de l'algorithme. L'invariant de l'algorithme est $d = a^c\, mod(n)$ puisque c ne fait que doubler ou doubler et croître de un jusqu'à ce que $c = b$.

On observe en outre que la valeur de c juste aprèsle traitement du bit b_i est la même que le préfixe $\langle b_k, b_{k-1}, \dots, b_0 \rangle$ de la représentation binaire de b.

Si les entiers $a\ et\ b$ sont des nombres sur $k\ bits$, le nombre total d'opérations arithmétique nécessaires est de l'ordre$\mathcal{O}(k)$ et le nombre d'opérations sur les bits est en $\mathcal{O}(k^3)$.

3.2.2.3 Courbe elliptique

a. *Loi de Groupe*

Les applications des courbes elliptiques en cryptographie sont principalement dues à l'existence d'une loi de groupe que nous pouvons définir sur ces dernières. En effet, l'ensemble $E \cup \mathcal{O}$ peut être équipé avec une opération d'addition qui produit un groupe abélien dont l'élément neutre est le point à l'infini$\mathcal{O}$ [119].

En cryptologie, les courbes elliptiques sont utilisées dans le corps $\mathbb{F}_p$ avec p un nombre premier strictement supérieur à 3.

Soit E une courbe elliptique définie sur$\mathbb{F}_p$. L'équation affine de Weierstrass de E peut être simplifiée, pour$K \neq 2,3$, par :

$$y^2 = x^3 + ax + b \tag{3.08}$$

La courbe définie par cette formule admet un unique point à l'infini (i.e. avec $z = 0$), de coordonnées (0 :1 :0). C'est en général ce point qui sera distingué. On appelle discriminant de cette courbe l'élément $-16(4a^3 + 27b^2)$ de K.

Le facteur entre parenthèse est le discriminant du polynôme membre de droite :

$$4a^3 + 27b^2 \not\equiv 0(modp) \tag{3.09}$$

Soient $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ deux points sur E :

- Le point $(x_1, -y_1)$ est l'opposé du point P et il est noté −P.
- Si $Q \neq P$ et$Q \neq -P$, alors le point $R = P + Q = (x_3, y_3)$ est défini par :

$$\begin{cases} x_3 = \left(\dfrac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2 \\ y_3 = y_1 + \left(\dfrac{y_1 - y_2}{x_1 - x_2}\right)(x_3 - x_1) \end{cases} \tag{3.10}$$

- Si $P = Q$, alors le point $2P = (x_3, y_3)$ est défini par :

$$\begin{cases} x_3 = \left(\dfrac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \\ y_3 = x_1 + \left(\dfrac{3x_1^2 + a}{2y_1}\right)(x_3 - x_1) \end{cases} \tag{3.11}$$

- Si $x_1 = x_2$ mais$y_1 \neq y_2$, alors $R = \mathcal{O}$
- Si $P = Q$et $y_1 = 0$, alors $R = \mathcal{O}$

b. *Algorithmes de calcul sur les courbes elliptiques*

Les algorithmes suivants permettent de calculer le doublement et l'addition.

Pour ce premier algorithme nous allons procéder ainsi :

- Parcourir chaque bit de la clé.
- A chaque étape, doubler une variable qui sert de résultat,
- Initialiser à l'élément neutre,
- Si le bit en cours est à 1, alors ajouter le point P à ce résultat et continuer.
- Retourner la valeur de la variable.

Algorithme 3.05 :

$Nom : doubleAndAdd$

$Entrée : P, d = (d_{l-1}, \dots, d_0)_2$

$Sortie : Q = [d]P$

$début$

$T \leftarrow \mathcal{O}$

$pour\ i \leftarrow l - 1$ à 0 $faire$

$T \leftarrow [2]T$ // Doublement

$si\ d_i = 1\ alors$

$T \leftarrow T + P$ // Addition

$finsi$

$finpour$

$retourner\ T$

fin

Algorithme 3.06

L'algorithme 3.06 ci-après est une modification de l'algorithme *doubleAndAdd* qui permet une plus grande résistance face aux attaques simples par analyse de canaux cachés. Le but de cet algorithme est d'exécuter un doublement et une addition à chaque pas de l'algorithme, et ce, quelque soit la valeur du bit de clé traitée.

$Nom : doubleAndAddAlw$

$Entrée : P, d = (d_{l-1}, \dots, d_0)_2$

$Sortie : Q = [d]P$

$Début$

$T_0 \leftarrow \mathcal{O}$

$T_1 \leftarrow \mathcal{O}$

$Pour\ i \leftarrow l - 1$ à 0 $faire$

$T_0 \leftarrow [2]T_0$ // Doublement

$T_1 \leftarrow T_0 + P$ // Addition

$si\ d_i = 1\ alors$

$T_0 \leftarrow T_1$ // Calcul de T_1utile

$sinon$

$T_0 \leftarrow T_0$ // Calcul de T_1inutile

$finsi$

$finpour$

$retourner\ T_0$

fin

Les deux algorithmes précédents sont les plus simples que l'on peut trouver pour calculer $[d]P$ sans effectuer d'additions. Cependant, on voit que le nombre d'additions dépend du nombre de bits à 1 dans la représentation binaire de la clé. Il semble alors naturel de vouloir éclaircir cette représentation, c'est-à-dire de trouver un moyen pour avoir le moins de 1 possible. Pour cela, on va utiliser un algorithme qui ne va plus prendre en entrée la représentation binaire de k, mais une représentation dite *NAF*, pour Non-Adjacent Form. Dans cette représentation, il n'y a pas de chiffres consécutifs différents de 0, mais on introduit des chiffres à -1. Il nous faut les deux algorithmes suivant: un pour calculer cette nouvelle représentation; un autre pour effectuer la multiplication scalaire.

Le premier retourne la représentation du scalaire en base 2 où il y a un écart d'au moins un bit entre deux chiffres non nuls. Une fois cette représentation calculée, on peut effectuer la multiplication scalaire grâce au deuxième algorithme.

Algorithme 3.07

Nom : naf
$Entrée$: $k \in \mathbb{N}$
$Sortie$: $NAF\ (k)$

$début$

$i \leftarrow 0$

$tant\ que\ k \geq 1\ faire$

$si\ k\ est\ impaire\ alors$

$k_i \leftarrow 2 - (k\ mod\ 4)$

$k \leftarrow k - k_i$

$sinon$

$k_i = 0$

$k \leftarrow k/2$

$i \leftarrow i + 1$

$finsi$

$retourner\ (k_{i-1}, \dots, k_1, k_0)$

fin

Et pour le calcul de kP, nous avons :

Algorithme 3.08

Nom : $mulNaf$
$Entrée$: $P,\ k \in \mathbb{N}$
$Sortie$: $Q = [k]P$

$début$

$Calcul\ de\ naf$

$Q \leftarrow \infty$

$pour\ i\ de\ t - 1$ à $0\ faire$

$Q \leftarrow 2Q$

$si\ k_i = 1\ alors$

$Q \leftarrow Q + P$

$sinon\ si\ k_i = -1\ alors$

$Q \leftarrow Q - P$

$finsi$

$retourner\ Q$

$finpour$

fin

Nous avons aussi utilisé un algorithme beaucoup plus rapide appelé l'exponentiation de Montgomery. Pour calculer kP, l'exponentiation de Montgomery part du couple $(P, 2P)$ et donne en sortie le couple $(kP, (k\ +\ 1)P)$. Voici son algorithme :

Algorithme 3.09

$Nom : montgomery$
$Entrée : P,\ k = (1, k_{n-2}, \dots, k_0)_2$
$Sortie : Q = [k]P$

$Début$

$P_1 \leftarrow P$

$P_2 \leftarrow 2P$

$Pour\ i \leftarrow n-2$ à $0\ faire$

$P_{k_1-1} \leftarrow P_1 + P_2$ // Addition

$P_{k_1} \leftarrow 2P_{k_1}$ // Doublement

$finpour$

$retourner\ P_1$

fin

Pour le chiffrement RSA, nous proposons l'algorithme suivant :

Algorithme 3.10

$Nom : rsa$
$Entrée: n, m$
$Sortie: c$

$début$

$a\ premier$

$b\ premier$

$n \leftarrow a * b$

$rep \leftarrow 0$

$tant\ que\ rep = 0\ faire$

$e\ premier$

$tel\ que\ e > 2\ \&\ pgcd(e,n) = 1$

$retourner\ rep$

$fintant\ que$

$c \leftarrow mod\ m^e, n$

fin

3.2.2.4 Les suites pseudo-aléatoires

La génération de suite pseudo-aléatoire de bonne qualité est rendue cruciale, d'une part pour des systèmes de chiffrement probabilistes comme celui d'El Gamal, d'autre part pour les chiffres à masque jetable, qui réalise la condition du secret parfait [118].

Avec un tel chiffre, A et B peuvent échanger des messages secrets. Ils partagent une clé secrète K (une suite aléatoire de n bits). Si A veut envoyer un message M de n bits à B, il envoi le cryptogramme $C = M \oplus K$ où C est la somme modulo 2 des bits de M et de K.

Le cryptogramme reçu peut être déchiffré par B en effectuant l'opération $C \oplus K = M \oplus K \oplus K = M \oplus 0$. Si d'autres messages doivent être envoyés, il faut engendrer une nouvelle clé (usage unique).

La sécurité de ce chiffre repose sur la qualité de la suite aléatoire. Intuitivement, une suite finie est dite aléatoire si elle ne peut pas être comprimée ou sa complexité de Kolmogorov est élevée. La complexité de Kolmogorov de la suite s est définie de façon informelle comme la taille du plus petit programme qui permet d'écrire s. En faite, on peut montrer qu'il est impossible de construire une suite aléatoire parfaite. Toutes les suites que nous pouvons engendrer « mécaniquement » ne sont qu'une approximation des suites aléatoires. On les appelle des suites pseudo-aléatoires.

a. Génération de suites aléatoires

La création d'un chiffre à masque jetable requiert l'usage d'une source « naturelle » de bits aléatoires comme une source radioactive ou la répétition de tirages à pile ou face d'une pièce. De telles sources sont absolument essentielles pour fournir une clé secrète initiale pour les systèmes de chiffrement. Cependant, beaucoup de sources naturelles de bits aléatoires produisent une suite qui est biaisée (la probabilité que le bit i soit à 1est plus grande que la probabilité que le bit i soit à 0) ou dont les bits sont corrélés (la suite se répète périodiquement) [117], [118].

Dans notre approche, pour transformer une suite biaisés sans corrélation en une suite sans biais, on a regroupé les bits par paires et transformé 01 en 0, 10 en 1 et effacé les paires de la forme 00 et 11. Le résultat de cette transformation sera une suite sans biais et sans corrélation puisque les paires 01 et 10 auront une même probabilité d'apparition.

b. Génération de suites pseudo-aléatoires

Le chiffrement de Vernam n'est pas très utilisé à cause de la grande longueur de sa clé. Cependant, la taille de la clé peut être réduite s'il est possible de mémoriser seulement une petite suite aléatoire x et de l'étendre en une suite pseudo-aléatoire y au fur et à mesure des besoins. Le procédé qui permet d'engendrer une telle suite est appelé générateur de suite pseudo-aléatoire ou GPA. La suite y est appelée pseudo-aléatoire car elle est construite au moyen d'un procédé de calcul déterministe qui ne peut donc pas engendrer toutes les suites y possibles.

Néanmoins, l'idée est de produire une suite y qui ne peut pas être distinguée d'une vraie suite aléatoire de même longueur.

Pour ce faire, on va utiliser l'hypothèse suivante telle que : il existe des fonctions à sens unique pour construire de bons générateurs de suites pseudo-aléatoires binaires. On utilise pour cela des *prédicats à sens uniques.*

Soit D un ensemble fini et $f: D \to D$ une permutation à sens unique des éléments de D. Soit $B: D \to \{0,1\}$ un prédicat à sens unique tel que :

(1) étant donné $x = f^{-1}(y)$, il est facile de calculer $B(y)$.

(2) étant donné seulement y, il est difficile de calculer $f^{-1}(y)$

Théorème 3.01

L'existence de fonctions à sens unique implique l'existence de fonctions à sens unique dont on peut extraire un prédicat à sens unique.

Soit f une permutation à sens unique et soit la fonction f' définie par :

$$f'(x_1, x_2, \dots, x_{n^3}) = f(x_1).f(x_2) \dots f(x_{n^3})$$

♣Pour des $x_1, x_2, \dots, x_{n^3})$ de même longueur n. Si $b(i,x)$ représente le i-ème bit de x, le prédicat à sens unique extrait de f' est défini par :

$B(x_1, x_2, \dots, x_{n^3}) = \sum_{i=1}^{n} \sum_{j=1}^{n^2} b\left(i, x_{(i-1)n^2+j}\right) mod\ 2$ ♦

Théorème 3.02

L'existence de permutations à sens unique implique celle de générateurs pseudo-aléatoires. De plus, pour tout polynôme $Q(n)$, il existe un générateur pseudo-aléatoire qui étend des suites aléatoires de taille n en des suites pseudo-aléatoires de taille $Q(n)$.

Lemme 3.01 :

L'existence de permutations à sens unique implique celle d'un générateur pseudo-aléatoire qui étend son entrée d'un bit.

La construction d'un GPA est alors la suivante : soit f une permutation à sens unique et B son prédicat à sens unique. On propose alors comme GPA la fonction G_1 définie comme suit :

$$G_1(x) = f(x).B(x)$$

On peut alors montrer par l'absurde que G_1 est bien un GPA

Lemme 3.02 :

L'existence de générateurs pseudo-aléatoires qui étendent leur entrée d'un seul bit implique que, pour tout polynôme $Q(n)$, il existe un GPA qui étend toutes les entrées de taille n en des sorties de longueur $Q(n)$.

Soit G satisfaisant les hypothèses. On construit alors un GPA G_Q qui, pour tout polynôme $Q(n)$, va étendre son entrée de longueur n en une sortie de longueur $Q(n)$.

Pour $x = x_1x_2 \dots x_{n+1}$, on peut écrire $x = pref(x).rmb(x)$ où $pref(x) = x_1x_2 \dots x_n$ et $rmb(x) = x_{n+1}$.

On pose $m = Q(n)$et on définit $b_G^{(i)}(x) = rmb(G(x^{(i)}))$ pour $x^{(i)}$ défini par la récurrence :

$$\begin{cases} x^{(0)} = x \\ x^{i+1} = pref(G(x^{(i)})) \end{cases}$$

On définit ensuite G_Q: $G_Q(x) = b_G^{(i)}(x)b_G^{(2)}(x) \dots b_G^{(m)}(x)$

Un tel GPA possède la propriété qu'il est difficile de prédire le $(1+i)^e$ bit connaissant la valeur des bits 0 à i avec un taux de réussite supérieur à $(1+\varepsilon)/2$ pour un algorithme qui travaille en temps polynomial sur la taille de la suite aléatoire fournie en entrée et la valeur $1/\varepsilon$.

Il ne reste plus qu'à utiliser une bonne fonction à sens unique comme RSA ou ECC.

c. Générateur RSA

On prend comme permutation à sens unique $f(x) = x^e \ mod \ N$.

N est un entier de taille n produit de deux premier $p \ et \ q$ et e est un entier tel que $pgcd\big(e, \varphi(N)\big) = 1$.

On extrait le prédicat $B(x) = lsb(x)$, où $lsb(x)$ retourne le bit de poids faible de $x^{(0)}$.

Le générateur RSA prend en entrée un triplet (N, e, x) et fonctionne comme suit :

Algorithme 3.11 :

$entrée : (N, e, x)$

$x_0 \leftarrow x$

$pour \ i \leftarrow 1 \text{ à } l \ faire$

$x_i \leftarrow x_{i-1}^e \ mod \ N$

$y_i \leftarrow lsb(x_i)$

$fpour$

$sortie : la \ suite \ (y_i)_{i=1,\dots,l}$

3.3 Modélisation mathématique d’un appel SIP

3.3.1 Introduction

Le but principal de ce paragraphe est de définir une formulation mathématique d'un appel VoIP (pour Voice over IP) en utilisant le protocole SIP (Session Initiation Protocol). Cette formulation, que nous avons obtenue à partir de l'utilisation de MATLAB, nous amène dans un futur proche à modéliser le comportement des appels SIP au niveau d'un serveur Asterisk tout en prenant comme paramètre les ressources matérielles telles que le processeur et les mémoires.

Basée sur la formulation mathématique de tous les paramètres, le modèle mathématique final permet de répondre à la question :

Quel type de matériel pour combien de clients VoIP ?

Les résultats du modèle mathématique sont conçus et comparés avec les résultats acquis expérimentalement.

Afin de garantir la qualité de service pour les utilisateurs et de profiter pleinement la capacité matérielle d'un serveur VoIP, il est nécessaire de déterminer le comportement du serveur pour un appel et ensuite pour un nombre fini d'appels pour résoudre cette problématique [123].

Les recherches que nous avons effectuées se situent donc au niveau de notre serveur Asterisk, cette recherche propose de répondre à partir d'un modèle mathématique polynomial la question qui s'est posée précédemment.

L’application de ce modèle est utilisable dans tous les domaines qui utilisent la technologie IP : que ce soit dans les réseaux informatiques que télécommunication, dans les domaines de l’aéronautique, du médical et des applications militaires.

Nous allons décrire les recherches antérieures dans le domaine de la VoIP. Ensuite, nous présentons une expérimentation sur un appel VoIP que nous avons effectuée ainsi que les résultats et interprétations. Enfin, nous proposons le modèle polynomial des résultats obtenus. Nous rééditons le modèle en utilisant les données obtenues et une perspective pour nos travaux de recherche postérieure seront mentionnées.

3.3.2 Analyse et modélisation des trafics VoIP

Cette étude d'analyse présente les caractéristiques des trafics VoIP mesurées sur les deux appels et au niveau des paquets transmis [124].

Les résultats ainsi obtenus servent à soutenir le processus de Poisson (3.12) pour la modélisation des appels VoIP. Au niveau des paquets, la recherche nous montre que la modélisation exponentielle des périodes On et Off sont également inappropriée et que beaucoup de caractères ont été identifiés dans le cas de tous les codecs VoIP.

La distribution de Pareto généralisée peut être utilisée comme un modèle précis pour les périodes On et Off. Après analyse, le trafic VoIP à des caractéristiques et une suggestion pour le modèle de bruit blanc gaussien fractionnaire est proposé [122].

Définition 3.01 :

Un processus ϵ_t est qualifié de bruit blanc gaussien si :

- $\epsilon_t$est un bruit blanc indépendant
- $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ avec $\mathcal{N}(0, \sigma^2)$ une loi normale etσ^2 représente l'écart-type

3.3.2.1 Modèle Gaussien

Nous avons les deux types de variables aléatoires suivantes:

- Variable aléatoire dite discrète: X prend ses valeurs dans un ensemble χ au plus dénombrable. Dans ce cas, sa loi est complètement caractérisé par les probabilités de ses singletons à savoir $p_X(x) = \mathbb{P}\{X = x\} \geq 0$, où $x \in \chi$, avec $\sum_{x \in \chi} p_X(x) = 1$.
- Variable aléatoire dite "continue": X prend ses valeurs dans $\mathbb{R}^d$ et il existe une fonction $p_X(X) \geq 0$ définie sur $\mathbb{R}^d$ telle que la probabilité pour que $X \in \Delta \subset \mathbb{R}^d$ s'écrive :

$$\mathbb{P}\{X \in \Delta\} = \int_{\Delta} p_X(x_1, \dots, x_d) dx_1 \dots dx_d$$

avec $\int_{\mathbb{R}^d} p_X(X) dX = 1$

Le modèle gaussien joue le rôle essentiel dans la représentation du bruit qui intervient dans les systèmes de communications.

Définition 3.02 : Variable aléatoire gaussienne

On dit qu'une variable aléatoire X de variance non nulle est Gaussienne si sa loi de probabilité a pour densité de probabilité:

$$p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(\frac{(x-m)^2}{2\sigma^2}\right) \text{ où } \sigma \in \mathbb{R}^+ \tag{3.12}$$

On montre que la moyenne $\mathbb{E}\{X\} = m$ et que la variance$\mathbb{E}\{(x-m)^2\} = \sigma^2$.

Définition 3.03: Vecteur aléatoire gaussien

On dit que le vecteur aléatoire $X = [X_1\, X_2 \ldots\, X_d]^T$ est gaussien si toute la combinaison linéaire de ses composantes est un variable aléatoire gaussienne.

On montre que sa loi de probabilité a pour densité:

$$p_X(x_1 \ldots\, x_d) = \frac{1}{(2\pi)^{d/2}\sqrt{\det(C)}} exp\left(-\frac{1}{2}(x-m)^T C^{-1}(x-m)\right) \tag{3.13}$$

où $X = (x_1 \ldots x_d)^T$. Le vecteur m est de dimension d. La matrice X , de dimension $d \times d$, supposée telle que $\det(C) \neq 0$ est positive c'est à dire telle que, pour tout $v \in \mathbb{C}^d$, on a $v^H \mathrm{C} v \geq 0$.

On déduit de l'équation précédente que le vecteur moyenne $\mathbb{E}\{X\} = m$ et que la matrice de covariance $\mathbb{E}\{(X-m)(X-m)^T\} = C$.

On dit que les composantes ne sont pas corrélées si C est une matrice diagonale. Dans ce cas les variables aléatoires $X_1, X_2, \ldots\, X_d$ sont en plus indépendantes.

Définition 3.04 : *Vecteur gaussien, blanc*

Un vecteur gaussien est dit blanc si son vecteur-moyenne $m = 0$ et si sa matrice de covariance $C = \sigma^2 I_d$ où I_d désigne la matrice identité de dimension d. Il s'ensuit que, s'il est gaussien, sa densité de probabilité a pour expression:

$$p_X(x_1 \ldots\, x_d) = \frac{1}{(2\pi)^{d/2}\sigma^2} exp\left(-\frac{1}{2\sigma^2}\sum_{k=1}^{d} x_k^2)\right) \tag{3.14}$$

Dans ce cas, ses composantes $(X_1, X_2, \ldots\, X_d)$sont des variables aléatoires indépendantes.

3.3.2.2 Processus aléatoire

Définition 3.05 : Processus aléatoire gaussien

On dit qu'un processus aléatoire $X(t)$, où $t \in \mathbb{R}$ ou $t \in \mathbb{Z}$, est gaussien si, pour tout d et pour toute suite d'instants $t_1, t_2, \ldots t_d$, le vecteur $[X_{(t1)}, X_{(t2)}, \ldots X_{(td)}]$ est un vecteur gaussien.

Propriétés 3.01:

- pour des variables gaussiennes, la non-corrélation entraine l'indépendance,
- le caractère gaussien se conserve par transformation linéaire et donc en particulier par filtrage linéaire.

3.3.3 Analyse des appels

L'analyse a été réalisée dans l'environnement MATLAB et affiche les caractéristiques (moyenne et variance) des appels dans le tableau suivant :

Statistiques	**Appel établi en seconde**	**Appel émis**
Nombre d'exemples	4733	464161
Moyenne	6,0830	114,2701
Variance	55,5998	36,904

Tableau 3.05 *: Statistique de base des appels*

Le tableau 3.05 présente les statistiques de base de l'appel mesuré en temps et en séries de données. Les données d'appel émis ont été choisies parmi une période chargée d'une journée de travail représentatif (l'analyse a été effectuée en 211 jours), où le trafic d'appels est presque stationnaire. L'appel en attente peut parfois être considéré comme indépendant des jours, donc tout le temps déclaré détenir plus de 240 jours de travail ont été utilisés pour l'analyse.

Ces deux processus ont été équipés avec des distributions différentes. On a pu constater dans l'expérience que le temps d'appel émis suit la loi exponentielle de paramètre $\lambda = 0{,}164$. Par ailleurs, la fonction d'auto corrélation créée à partir des données a montré que les temps inter arrivés peuvent être considérés comme indépendants. Ceux-ci indiquent que ce modèle, qui suit la loi de Poisson, est exact pour le processus d'arrivée des appels.
Des résultats similaires ont été observés dans l'analyse des différentes journées de travail.

3.3.4 Modèle proposé

Soit X_n le nombre d'arrivées pendant la n-ième unité de temps. Nous avons donc une moyenne $\bar{X}$ comme moyenne par unité de temps avec :

$$\frac{1}{N}(X_1 + X_2 + \cdots + X_n)$$

Nous savons que $(X_1 + X_2 + \cdots + X_n)$ est une suite de la forme (X_n) qui n'est pas définie donc imprévisible. Soit nous avons :

$$\frac{1}{N}(1_{\{X_1=x\}} + 1_{\{X_2=x\}} + \cdots + 1_{\{X_n=x\}}) \sim P_x$$

Où$P_x, x \in N$, une loi de probabilité des arrivées des appels par unité de temps.

Dans cette expérience nous avons proposé comme modèle mathématique la loi de Poisson, c'est-à-dire :

$$P_x = \frac{\lambda^x}{x!} e^{-\lambda}, x \in \mathrm{N} \qquad (3.15)$$

Avec λ un paramètre et Nnombre d'appels.

3.3.5 Expérience

La figure 3.01 représente l'architecture de l'expérience que nous avons opté pour analyser le trafic du protocole SIP

3.3.5.1 Matériel déployé

- Un serveur Asterisk installé sur Débian GNU/Linux
- Deux postes client avec un soft phone x-lite installé sur Windows XP
- Une Switch pour met en réseaux l'architecture
- Des micros et des casques pour l'émission et réception des voix sur le réseau

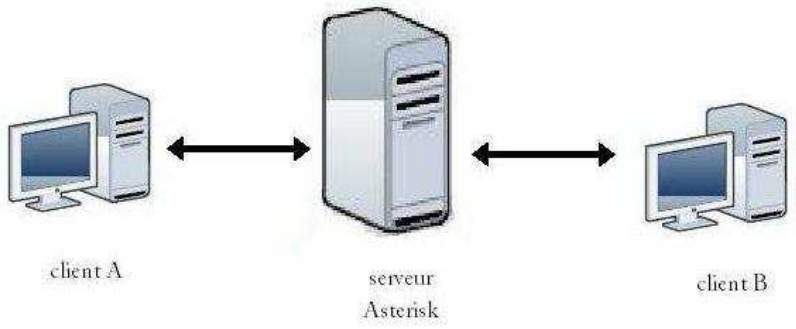


Figure 3.01: *Architecture de l'expérience*

Notre architecture réseau est minimisée par un serveur connecté à deux ordinateurs clients pour effectuer l'appel ou plus précisément pour initier une session avec le protocole SIP. Tout cela est relié par un Switch.

3.3.5.2 Logiciel déployé

a. Debian GNU/Linux

Debian est un système d'exploitation qui est développé par des communautés et a comme racine le noyau Linux

b. Asterisk

Asterisk est un autocommutateur téléphonique privé (PABX) open source et propriétaire (publié sous licence GPL et licence propriétaire1) pour systèmes UNIX, Mac OS et Windows. Il permet, entre autres, la messagerie vocale, les files d'attente, les agents d'appels, les musiques d'attente et les mises en garde d'appels, la distribution des appels. Il est possible également d'ajouter l'utilisation des conférences par le biais de l'installation de modules supplémentaires et la recompilation des binaires [120].

Asterisk implémente les protocoles H.320, H.323 et SIP, ainsi qu'un protocole spécifique nommé IAX (Inter-Asterisk eXchange). Ce protocole IAX permet la communication entre deux serveurs Asterisk ainsi qu'entre client et serveur Asterisk. Asterisk peut également jouer le rôle de registrar et passerelle avec les réseaux publics (RTC, GSM, etc.) Asterisk est extensible par des scripts ou des modules en langage Perl, C, Python, PHP, et Ruby.

c. Windows XP

Windows XP est une famille de systèmes d'exploitation multitâches propriétaires, développée par Microsoft, permettant l'usage d'un ordinateur tel qu'un ordinateur fixe, un portable ou encore un Media Center. Les lettres "XP" proviennent d'eXPerience.

d. X-lite

X-Lite est un logiciel propriétaire gratuit client de téléphonie sur IP appelé également soft phone, basé sur le protocole standard ouvert SIP. X-Lite est un logiciel multiplateforme pour Mac OS X, Windows et Linux.Associé à un compte SIP, il permet de bénéficier de tous les services téléphoniques traditionnels (conférence, doubles appels, messagerie vocal etc.)

e. Le tcpdump

Le logiciel tcpdump est un analyseur de paquets en ligne de commande. Il permet d'obtenir le détail du trafic visible depuis une interface réseau. L'outil distribué par les distributions GNU/Linux, Free BSD, Net BSD, Open BSD et Mac OS X dépend de la bibliothèque logicielle libpcap. Leur portage sous Windows est connu sous les appellations WinPCAP/WinDUMP.

Les logiciels tcpdump et libpcap sont développés en 1987 au laboratoire national Lawrence Berkeley aux États-Unis par Van Jacobson, Steven McCanne et Craig Leres, le créateur d'arpwatch. Vers la fin des années 1990, tcpdump est distribué dans de nombreux systèmes ce qui ne favorisait guère l'application de correctifs. Michael Richardson et Bill Fenner créent un site officiel en 1999 pour répondre à ce manque de coordination et deviennent alors les mainteneurs du projet.

f. Le tcpstat

Le tcpstat rapporte certaines statistiques relatives aux interfaces réseau un peu comme vmstat fait pour les statistiques du système. Tcpstat obtient ses informations par le suivi d'une interface spécifique, ou en lisant les données précédemment enregistrées tcpdump à partir d'un fichier.

Une partie de la tcpstat statistique calcule:

- La bande passante
- Le nombre de paquets par second
- La taille moyenne des paquets
- L’ écart-type de la taille des paquets

Tcpstat est écrit avec la performance et l'efficacité à l'esprit et est capable de manipuler de grandes quantités de paquets par seconde. Son interface ligne de commande compacte est conçu pour le chercheur du réseau, administrateur système et Shell utilisateur de bureau. Tcpstat a été un outil précieux dans les documents de recherche universitaires et des réseaux commerciaux.

g. Gnuplot

Gnuplot est un programme souple qui peut produire des représentations graphiques en deux ou trois dimensions de fonctions numériques ou de données. Le programme fonctionne sur tous les ordinateurs et systèmes d'exploitation principaux et peut envoyer les graphiques à l'écran ou dans des fichiers dans de nombreux formats.

Le programme est distribué sous une licence de logiciel libre qui permet de copier et de modifier le code source du programme. Les versions modifiées du programme ne peuvent être distribuées que sous forme de fichiers correctifs. Le programme n'a aucun raccordement avec le projet GNU et n'utilise pas la licence de « copyleft » GPL.

Le programme peut être utilisé interactivement, et est accompagné d'une aide en ligne. L'utilisateur entre en ligne de commande des instructions qui ont pour effet de produire un tracé. Il est aussi possible d'écrire des scripts Gnuplot qui, lorsqu' ils sont exécutés, génèrent un graphique.

Gnuplot est utilisé comme moteur de traçage d'Octave et de Maxima qui sont les équivalences de MATLAB dans le domaine de l’open source

3.3.5.3 Méthodologie

D'abord, nous commençons à lancer l'appel SIP partant du Softphone du client Windows, suivi d’une capture de paquets sur notre serveur Asterisk via la commande "tcpdump".

Ensuite, nous organisons les paquets capturés par la commande "tcpstat" de sorte qu'il peut être tracé par notre traceur de courbes Gnuplot.

Enfin, nous observons le comportement de notre graphique pour un temps déterminé avant que l'appel soit raccroché.

Après l'expérience nous avons obtenu les résultats suivants (figure 3.02)

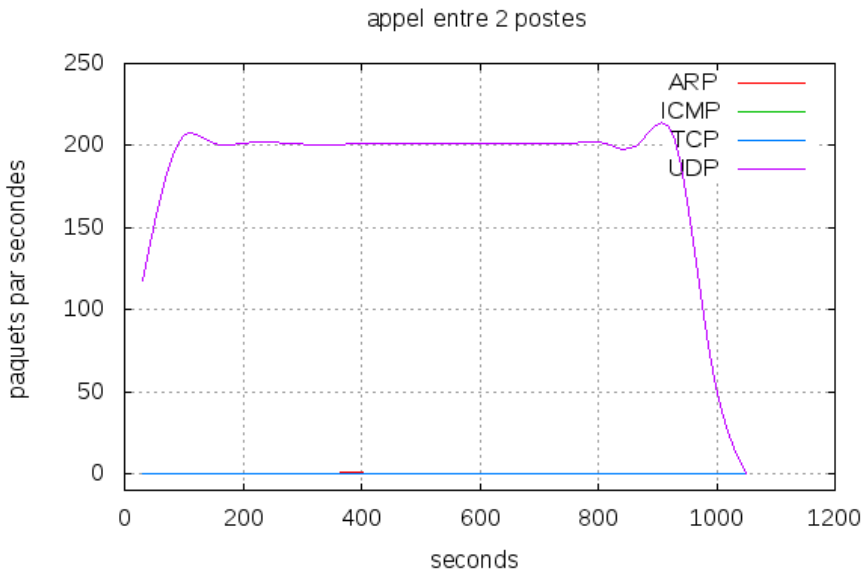


Figure 3.02: *Nombre de paquets par seconde*

La capture de paquet faite par tcpdump pendant la phase des sessions nous a donné la trame SIP suivant :

```
Via: SIP/
14:53:15.036004 IP debian.local.sip>
192.168.1.113.21460: SIP, length: 500
E....C..@.-i...o...q..S....>SIP/2.0   100
Trying
Via: SIP/2.0/UDP 192.168.1.1
14:53:15.036535 IP debian.local.sip>
192.168.1.222.44094: SIP, length: 944
E...{...@.wU...o.......>...gINVITE
sip:2222@192.168.1.222:44094;rinstance=0c
14:53:15.036616 IP debian.local.sip>
192.168.1.113.21460: SIP, length: 516
E..  .D..@.-X...o...q..S....NSIP/2.0  180
Ringing
  Via: SIP/2.0/UDP 192.168.1.
```

Figure 3.03: *Extrait du résultat de tcpdump*

Pour extraire les valeurs obtenus par tcpdump nous avons écrit un script sh dont nous présentons ci-après un extrait (figure 3.04) pour le faire tracé ensuite avec gnuplot.

```
#!/bin/bash

echo ce script permet de parser le
flux

tcpdump pour adapter a gnuplot

tcpstat    -r    traffic_2.dmp    -o
"%R\t%A\n"

60 > arp_2.data

tcpstat    -r    traffic_2.dmp    -o
"%R\t%C\n"

60 > icmp_2.data

tcpstat    -r    traffic_2.dmp    -o
"%R\t%T\n"

60 > tcp_2.data

tcpstat    -r    traffic_2.dmp    -o
"%R\t%U\n"

60 > udp_2.data

gnuplotgnuplot.script> graphe_2
```

Figure 3.04: *Extrait du script pour le traçage avec gnuplot*

3.3.5.4 Analyse de la courbe

D'après nos constatations, nous avons remarqué que lors de l'initiation de l'appel où SIP ouvre la session avec "INVITE", nous observons unpremier pic dans le nombre de paquets par seconde, un deuxième pic est constaté à la fermeture de la session, ce dernier pic correspond au même moment que le protocole SIP émet le message "BYE".

Nous pouvons donc conclure que:

- L'envoi du message INVITE requiert un grand nombre de paquets lors de l’ouverture d’une session SIP.
- De même pour "BYE" lors de la clôture d’une session SIP
- Enfin, notre simulation montre qu'un appel sans réponse peut causer une surcharge de notre serveur et pourrait à long terme mettre notre serveur en indisponibilité.

3.3.6 Modèle polynomial proposé

D'après la figure 3.02, ce dernier a été exporté dans le logiciel MATLAB et nous déduisons :

- Un modèle polynomial d'ordre 3. par le fait que nous définissons quatre (4) paramètres
- Le nombre d'appel émis
- L'espace mémoire utilisé
- L'utilisation du processeur
- L'occupation de la bande passante.

Ainsi, nous avons obtenu le polynôme $P(x)$ suivant:

$$\mathrm{P(x)} = 1.0\mathrm{e}+004(0\mathrm{x}^3 + 0\mathrm{x}^2 - 0.0005\mathrm{x} + 1.0358) \quad (3.16)$$

Définition 3.06 :

Le modèle polynomial d'un serveur est donc modélisé de la forme suivante :

$$P(x) = Ax^3 + Bx^2 + Cx + D$$

Où les matrices A, B, C, D sont des vecteurs colonnes qui représentent les variations des utilisations des paramètres du serveur.

On a :

$A = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$, a_1 à a_n : Ce sont les valeurs de l'utilisation de la mémoire en fonction du temps

$B = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$, b_1 à b_n : Ce sont les valeurs de l'utilisation du processeur en fonction du temps

$C = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$, c_1 à c_n : Ce sont les valeurs de l'occupation de la bande passante en fonction du temps.

$D = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$, d_1 à d_n : Ce sont la durée de temps de l'appel SIP

Le modèle mathématique d'un appel SIP est représenté par l'équation polynomiale à coefficients matriciels suivant :

$$P(x) = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} x^3 + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} x^2 + \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} x + \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix} \quad (3.17)$$

3.3.7 Vérification

- Dans cette partie nous avons essayé de vérifier la véracité de notre modèle analytique en exportant notre valeur sur MATLAB.
- Le résultat est presque identique à celle tracée par gnuplot

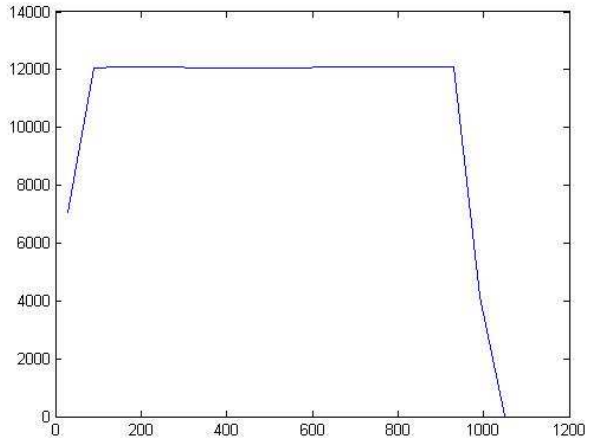


Figure 3.05: *Graphe du polynôme obtenu*

- Nous avons présenté une étude préliminaire du comportement du protocole SIP avec seulement deux machines clientes et aussi son comportement durant l'initiation de l'appel. Il a été démontré que certaines étapes de l'initiation d'appel utilisent beaucoup de paquets à savoir au moment du lancement du message invite et bye après avoir analysé les caractéristiques réelles des paquets obtenus avec tcpdump.

- Nous avons constaté que tout le processus d'initialisation de l'appel est bien modélisé par le modèle polynomial que nous avons pu obtenir par MATLAB. Nous avons également montré que ce modèle ressemble presque à celle obtenue par tcpstat et Gnuplot. Basé sur ces résultats la nature de notre courbe nous révèle beaucoup de domaine qui mérite une recherche plus approfondie à savoir s'il y a 3, 4, 5,…, n clients.

- Les caractéristiques de notre courbe nous amènent à une déduction que s'il y a plusieurs appels c'est à dire plusieurs paquets donc il aura une limite au niveau du matériel. Cependant, une étude détaillée et plus complète est nécessaire pour vérifier cette hypothèse et dans cette optique nous avons continué notre recherche dans le domaine du serveur Asterisk en prenant en considération tous les paramètres matériels du serveur.

3.4 Modélisation mathématique d'un serveur VoIP

Dans ce paragraphe, nous allons définir analytiquement un modèle mathématique d'un serveur VoIP, plus précisément le serveur Asteriskà partir des résultats de recherche de Eduardo Casilari et Pablo Montoro [118].

À partir de ces modèles, nous allons vérifier la similitude des résultats de l'expérience citée dans le tableau 3.06 avec notre approche. Notre thème met l'accent sur l'utilisation des matériels : RAM, CPU et de bande passante d'un Serveur VoIP par rapport au nombre d'appels émis par des clients utilisant le serveur.

3.4.1 Introduction

Selon les études que nous avons réalisées précédemment, nous avons constaté qu'on peut modéliser le comportement d'un appel SIP. Dans cette partie de notre ouvrage, nous allons examiner un autre aspect de leurs résultats et les interpréter à notre façon pour la simple raison que notre but c'est de modéliser le comportement d'un serveur VoIP et en plus, le Voice over Internet Protocol (VoIP) est la seule technologie qui domine l'utilisation d'Internet les dix dernières années. Dans le monde des affaires ou dans le monde de l'informatique, cette technologie nous montre son avenir pour la facilité d'utilisation et la simplicité de son déploiement, mais surtout pour son coût.

La migration vers la VoIP ne nécessite plus de nouvelle 'infrastructure tandis que pour la téléphonie traditionnelle et la téléphonie mobile, avec cette nouvelle technologie, exigent de nouveaux investissements pour son déploiement. Les grands progrès dans la recherche sur la norme protocolaire de VoIP nous conduisent à la migration vers le tout IP [120], [121].

Le choix du protocole SIP vient du fait qu'un grand nombre de recherches sont menées sur la VoIP, le protocole standard utilisé, son développement considérable et aussi sa licence GPL. Il en est de même pour l'utilisation du serveur Asterisk pour son évolution et évidemment pour sa licence (GPL) [122], [123], [124].

Notre objectif est de modéliser mathématiquement un serveur Asterisk en utilisant le protocole SIP afin de prédire ses performances.

Pour cette raison nous avons décidé de faire une recherche concernant "la modélisation mathématique d'un serveur VoIP "

Ainsi, notre étude est structurée comme suit: dans un premier temps, nous résumons les résultats des recherches faites autour de la technologie VoIP et les résultats que nous avons pris comme base de notre recherche.

Ensuite, on va décrire les étapes suivies jusqu'à l'élaboration de notre modèle qui est basé sur l'utilisation des formules statistique ou plus précisément la méthode des moindres carrés. Les résultats du modèle adopté sont discutés dans la suite de cette description.

b.1.2 Etude comparative des standards VoIP

Dans leur recherche intitulée « Étude comparative de la norme VoIP avec Asterisk », les deux auteurs (Montoro et casillari) font différentes expérimentations fondées sur le protocole utilisé dans la VoIP. Cette expérience est la suivante:

Le schéma de principe du système déployé pour l'étude, Figure 3.06, comprend trois composants: un Serveur Asterisk (l'unité de sous observation), un générateur d'appel (pour les appels envoyé vers Asterisk), et un moniteur de serveur (qui surveille le serveur Asterisk). La Figure 3.07 représente l'interconnexion de ces trois éléments: le générateur d'appel et le serveur Asterisk sont reliés par deux interfaces Ethernet différentes (eth0 et eth1).

Les appels seront initialement transmis via l'interface lien eth1, qui supporte le flux VoIP. Les équipements de suivi sont connectés au même commutateur en tant que lien eth0 qui est liés aux deux autres unités. Sur le serveur Asterisk, ils ont installé un système Débian GNU Linux sur un Pentium 4 (2,4 GHz) avec 1 Gode mémoire RAM. Asterisk 1.4 a été installé à partir des paquets de la distribution Débian, en utilisant une configuration standard.

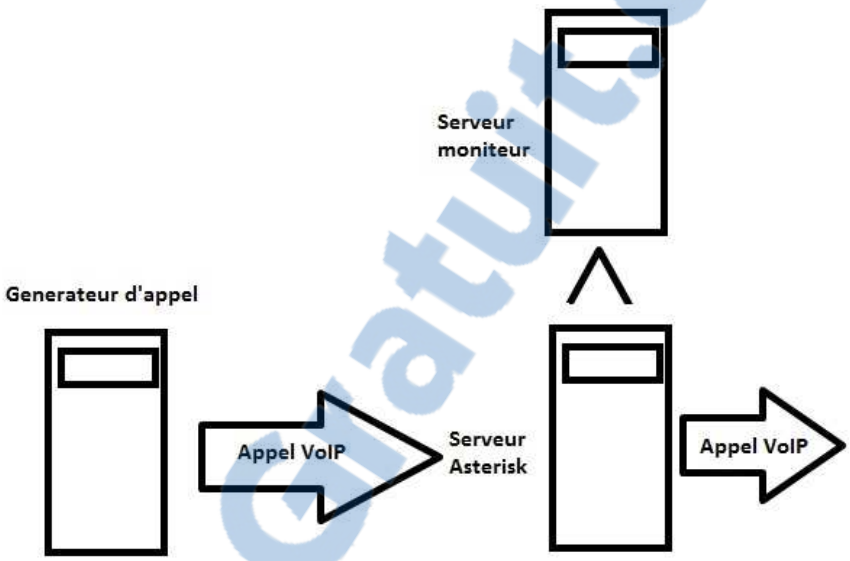


Figure 3.06: *L'architecture de surveillance*

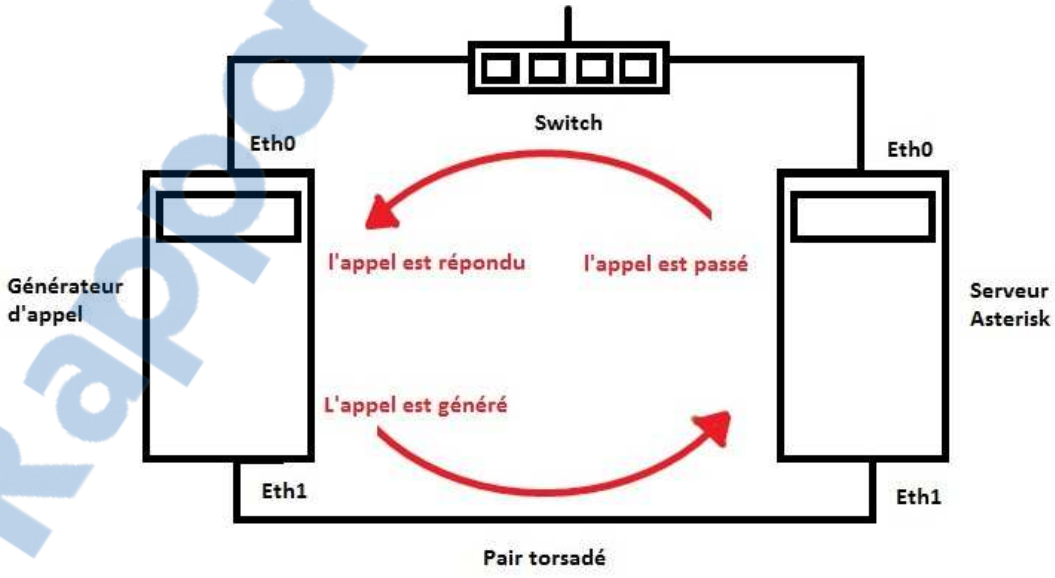


Figure 3.07: *Générateur d'appel*

Pour le résultat, nous avons pris seulement le résultat qui utilise le codec G711 et SIP et ce résultat est présenté dans le tableau 3.06

Nombre d'appels	**Utilisation du processeur (CPU :%)**	**Utilisation de la mémoire (RAM : Méga Bytes)**	**Utilisation de la bande passante (Kbps)**
10	2,36	33,38	86,05
20	4,64	38,23	86,00
30	7,12	42,09	86,00

Tableau 3.06 : *Résultat de l'expérience*

3.4.2 Description du calcul de la corrélation

Étudier la corrélation entre deux ou plusieurs variables aléatoires ou statistiques numériques, c'est étudier l'intensité de la connexion qui peut exister entre deux ou plusieurs variables. Dans notre cas, nous avons quatre variables à savoir le nombre d'appels, l'utilisation du processeur, utilisation de la mémoire et enfin l'utilisation de la bande passante.

Le résultat recherché est une relation affine.

Une mesure de cette corrélation est obtenue en calculant le coefficient de corrélation linéaire. Ce coefficient est le rapport de leur covariance et du produit non nul de leurs écarts types. Le coefficient de corrélation est compris entre -1 et 1.

$$R(x,y) = \frac{cov(x,y)}{\sigma x * \sigma y} \tag{3.18}$$

$$cov(x,y) = \frac{1}{N}\sum_{n=1}^{n} x_i y_i - \bar{x}\bar{y} \tag{3.19}$$

- Si $R\ (x,y) \sim \pm\ 1$, alors x, y sont corrélés.
- Si $R\ (x,y) \sim 0$, alors x, y ne sont pas corrélées.

Dans notre cas nous avons pris comme variables

- x=nombre d'appels
- y=utilisation de la CPU
- z=utilisation du mémoire RAM
- t=l'utilisation de la bande passante

Ainsi nous avons obtenu:

$$\begin{aligned} R(x,y) &= 0{,}99970589 \simeq 1 \\ R(x,z) &= 0{,}099785374 \simeq 1 \\ R(x,t) &= -0.8660254 \simeq -1 \end{aligned} \tag{3.20}$$

Définition 3.07 :

Le modèle mathématique d'un serveur VoIP est généralisé par l'équation suivante :

$$\begin{pmatrix} y \\ z \\ t \end{pmatrix} = \begin{pmatrix} 0.238 \\ 0.435 \\ -0.0025 \end{pmatrix} x + \begin{pmatrix} -0.053 \\ 29.19 \\ 86.06 \end{pmatrix} \tag{3.21}$$

$$M = Ax + B \tag{3.22}$$

M représente la matrice colonne $\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix}$

A représente la matrice colonne $\begin{pmatrix} 0.238 \\ 0.435 \\ -0.0025 \end{pmatrix}$

B représente la matrice colonne $\begin{pmatrix} -0.053 \\ 29.19 \\ 86.06 \end{pmatrix}$

Et x le nombre d'appel

3.4.3 Interprétation des résultats

Nous avons trouvé qu'il ya vraiment une relation entre le nombre d'appels et la variation de l'utilisation des ressources matérielles du serveur Asterisk. C'est à partir de cela qu'on en déduit un modèle mathématique basé sur le résultat précédent, c'est-à-dire, en définissant une équation qui peut représenter mathématiquement la formulation d'appel qui passe au niveau du serveur en fonction de ces ressources matérielles.

3.4.4 Interprétation et discussion

Nous pouvons dire que nous étions en mesure de proposer un modèle mathématique d'un serveur VoIP basée sur l'utilisation des ressources qui sont les caractéristiques même du serveur.

Après calcul de la corrélation nous avons bien évidemment constaté qu'il y avait une relation entre le nombre d'appels et l'utilisation du CPU, il existe une relation entre l'utilisation de la mémoire et le nombre d'appels et entre ce dernier et la bande passante. En utilisant un matériel à capacité limitée, le nombre d'utilisateurs sur un serveur Asterisk est aussi limité c'est-à-dire qu'un opérateur qui a l'intention d'avoir le maximum de client a l'obligation d'utilisé des ressources matériel puissant. Toutefois, nous avons trouvé que, en termes de variation de la bande passante, cette variable est presque insignifiant par rapport à la variation du nombre d'appels ceci prouve que la VoIP n'utilise pas forcement à fond cette variable. Nous arrivons donc à une conclusion partielle que la VoIP en général et le protocole SIP transitent sur Internet sans gêner les autres protocoles.

3.5 Conclusion

Dans cette partie, nous avons proposé des algorithmes de calcul cryptographiques pour AES, RSA, les courbes elliptiques et la génération de nombre pseudo-aléatoire. L'optimisation de ces algorithmes aura pour but de diminuer le temps de chiffrement/déchiffrement du cryptosystème en minimisant le nombre d'opération. D'autre part, nous avons fait des recherches, qui ont faits l'objet de publication scientifiques. Ces recherches constituent une base pour des études que nous allons exposer dans le chapitre suivant.

CHAPITRE 4
TRANSFERT SECURISE D'INFORMATION APPLIQUE AUX IMAGES ET A LA VOIX SUR IP

4.1 Introduction

Ce chapitre fait l'objet de deux publications dans des revues nationales et internationales dont MADA-Revue et International Journal of Computer Science and Network, Inde :

- « *Transfert sécurisé d'image dans le domaine de la TFD* », MADA-ETI, ISSN 2220-0673, Vol.1, pp.8-12, 2013, www.madarevue.gov.mg
- « *Système d'authentification par sécurisation d'index d'image utilisant SVD-ECC* », International Journal of Computer Science and Network, vol.2, 2013/ISSN 2277-5420, www.ijcsn.org/Publication

Les versions originales de ces articles sont présentées en annexe 3 et annexe 4.

❖ Le but de ce chapitre est d'implémenter les algorithmes proposés dans le troisième chapitre et de les tester, afin de faire des interprétations sur les résultats. Dans cette section, nous avons effectué trois types de simulation en manipulant à la fois des images et de la voix à travers la technologie VoIP.

4.2 Transfert sécurisé d'information dans le domaine de la Transformée de Fourier

4.2.1 Compression

Classiquement il y a deux approches pour crypter des données [129], [130].

- La première est un cryptage complet où toutes les informations sont chiffrées. L'inconvénient de ce type de cryptage est qu'il est toujours appliqué de la même manière, quel que soit l'application et le niveau de sécurité souhaité.
- La seconde approche de cryptage est adaptée au niveau de protection désiré. Le chiffrement est adapté à la sécurité désirée afin de réduire la consommation des ressources informatiques et le temps de calcul disponible.

C'est dans cette seconde approche que nous élaborons notre type de cryptage. Le cryptage sélectif (CS) est une approche qui ne chiffre qu'une partie des données afin de diminuer le temps de calcul. Les utilisateurs peuvent appliquer une sécurité proportionnelle ou réglable en fonction du niveau de protection désiré [133].

Un nombre important d'applications peut se contenter d'un niveau inférieur à un cryptage complet en utilisant un CS. Nous pouvons citer de nombreuses applications ou par exemple des parties de l'information doivent être visibles pour autoriser une recherche et une classification de données.

Des applications dans le domaine du transfert sécurisé d'image médicale présentent des images qui doivent être partiellement visibles sans révéler complètement toute l'information. Les peintures numériques doivent être présentées sur Internet avec une qualité visible réglable. Le transfert de photos depuis des téléphones portables peut également se contenter d'un cryptage partiel afin d'assurer la confidentialité. C'est aussi le cas des données transitant dans un canal de transmission peu sûr et à ressources limitées [129].

Pour des raisons vitales, dans ce type d'applications, les images doivent être transmises rapidement et surement, et dans ce cas un CS semble être la meilleure solution(compromis temps/sécurité).

Cette partie présente une nouvelle approche de CS basée sur l'utilisation de la transformée de Fourier (TFD), en l'occurrence l'utilisation de l'algorithme de la Transformée de Fourier Rapide (TFR) pour des images comprimées. Notre approche est basée sur le cryptage par AES sur certains flux binaires (coefficients des TFD) ainsi que l'utilisation de système d'Insertion de Données Cachées pour le partage sécurisé des clés de chiffrement.

4.2.1.1 Transformée de Fourier Discrète

La transformée de Fourier discrète (TFD) est un outil mathématique de traitement du signal numérique, qui est l'équivalent discret de la transformée de Fourier continue, utilisée pour le traitement du signal analogique [131].

Définition 4.01

- La transformée de Fourier discrète (TFD) pour un signal s de N échantillons est définie par l'expression :

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-2i\pi k \frac{n}{N}} \quad pour\ 0 \leq k < N \tag{4.01}$$

- La transformée de Fourier inverse (TFDI) est définie par l'expression :

$$s(n) = \sum_{k=0}^{N-1} S(k) e^{2i\pi n \frac{k}{N}} \tag{4.02}$$

Remarque :

- L'expression (4.01) permet d'obtenir une représentation spectrale discrète du signal échantillonné $s(n)$. Il est important de comprendre que la TFD ne calcule pas le spectre continu d'un signal continu. La TFD permet seulement d'évaluer une représentation spectrale discrète (spectre échantillonné) d'un signal discret (signal échantillonné) sur une fenêtre de temps finie (échantillonnage borné dans le temps) [131].
- La TFD est utilisée dans un large spectre d'applications, dans notre approche il est utilisé à des fins de compression et de représentation de données. Toutes ses applications nécessitent l'existence d'un algorithme rapide de calcul de la TFD et de son inverse, par les méthodes de transformée de Fourier rapide [132].
- Le traitement du signal en général utilise énormément d'opérations dans le domaine fréquentiel et en particulier la TFD ou une de ses variantes. En compression du son ou de l'image, des transformées proches de la TFD (par exemple la TCD) sont appliquées en général sur des portions de signal, pour réduire la complexité.
- Les coefficients S(k) sont ensuite quantifiés avec des pas de quantification plus élevés pour les hautes fréquences, qui sont considérées comme négligeables pour la perception humaine. Le gain en compression vient de la réduction de précision de ces coefficients (voire leur suppression totale) qui nécessitent alors moins de bits pour être codés. Il s'ensuit généralement une étape de codage entropique. La reconstruction du signal s'effectue alors à partir de cet ensemble réduit de coefficients quantifiés.

4.2.1.2 Transformée de Fourier Rapide

La transformée de Fourier rapide (*FFT* ou *Fast Fourier Transform*) est un algorithme de calcul de la transformée de Fourier discrète (TFD).

Sa complexité varie en $\mathcal{O}(n \log n)$ avec le nombre de points n, alors que la complexité du calcul de base s'exprime en $\mathcal{O}(n^2)$. Ainsi, pour n=1024, le temps de calcul de l'algorithme rapide peut être 100 fois plus court que le calcul utilisant la formule de définition de la TFD.

Cet algorithme est couramment utilisé en traitement numérique du signal pour transformer des données discrètes du domaine temporel dans le domaine fréquentiel, en particulier dans les analyseurs de spectre. Son efficacité permet de réaliser des filtrages en passant dans le domaine transformé [3].

Évaluer ces sommes directement coûte $(n-1)^2$ produits complexes et $n(n-1)$sommes complexes alors que seuls $\left(\frac{n}{2}\right)(log_2(n)-2)$produits et $n\,log_2(n)$ sommes sont nécessaires avec la version rapide.

En général, de tels algorithmes dépendent de la factorisation de n mais il y a des transformées de Fourier rapides de complexité $\mathcal{O}(nlog_2(n))$ pour tous les n, même les n qui sont des nombres premiers.

Comme la transformée de Fourier inverse discrète est équivalente à la transformée de Fourier discrète, à un signe et facteur $1/n$ près, il est possible de générer la transformation inverse de la même manière pour la version rapide.

4.2.2 Chiffrement AES en mode CFB

L'algorithme standard de chiffrement en blocs complexes comme DES, IDEA ou AES a été présenté dans la section 1.7.1 du premier chapitre. Il peut supporter les méthodes de chiffrement ECB, CBC, OFB et CFB, qui dépendent de la façon dont les blocs successifs du cryptogramme sont chiffrés. Les blocs de données et les clés peuvent être de longueur de 128, 192, ou 256 bits. Dans notre approche nous utilisons des clés de blocs de données de $16 \times 16\ bits$ et le mode CFB.

Dans ce mode, le flux de clés z_i est obtenu en cryptant le bloc chiffré précédent y_{i-1}. La première ronde est déclenchée avec le vecteur d'initialisation VI. Soient le cryptage $Y_0 = VI,\ z_i = e_K(Y_{i-1}),\ Y_i = z_i \oplus X_i$ et le décryptage $z_i = e_K(Y_{i-1}),\ X_i = z_i \oplus Y_i$ comme montré sur la figure 4.01 suivante :

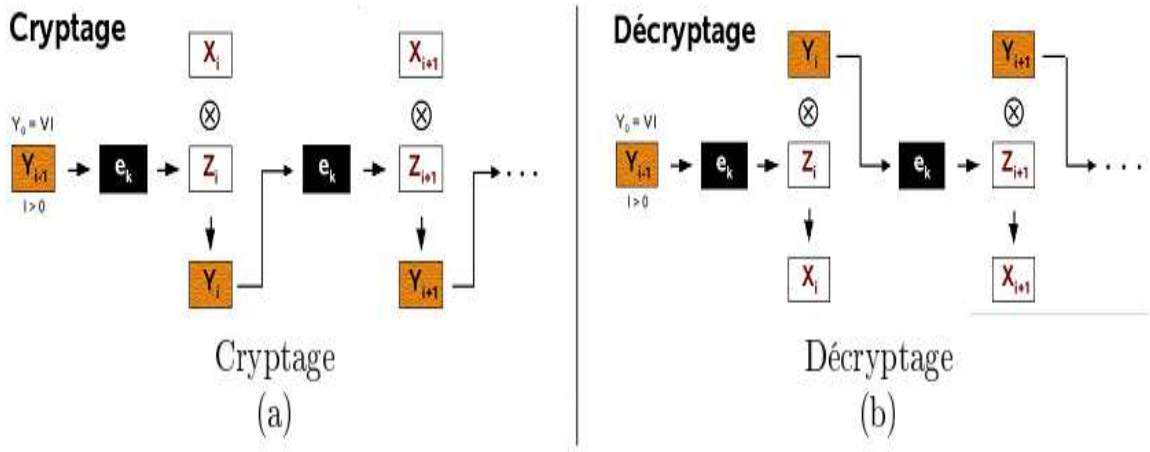


Figure 4.01: *Chiffrement AES en mode CBC*

Remarque

Dans le mode CFB de l'algorithme AES, il est important de noter que la fonction de cryptage e_K est utilisée pour la phase de cryptage (figure 4.01 (a)) mais également pour la phase de décryptage (figure 4.01 (b)).

L'opération logique utilisée est un « ou exclusif » appliqué à chaque bloc et ceci effectué en 16 rondes.

4.2.3 Tatouage d'images numériques

Le tatouage d'images numériques permet de cacher une information dans une image sous la forme d'une marque qui ne pourra être enlevée sans une altération importante de l'image [134].

Il permet de renforcer la fiabilité des données images [134], [135] :

- par l'insertion d'annotation (ou data) dans l'image source (Authenticité)
- par l'insertion d'une signature de l'image source (Intégrité)

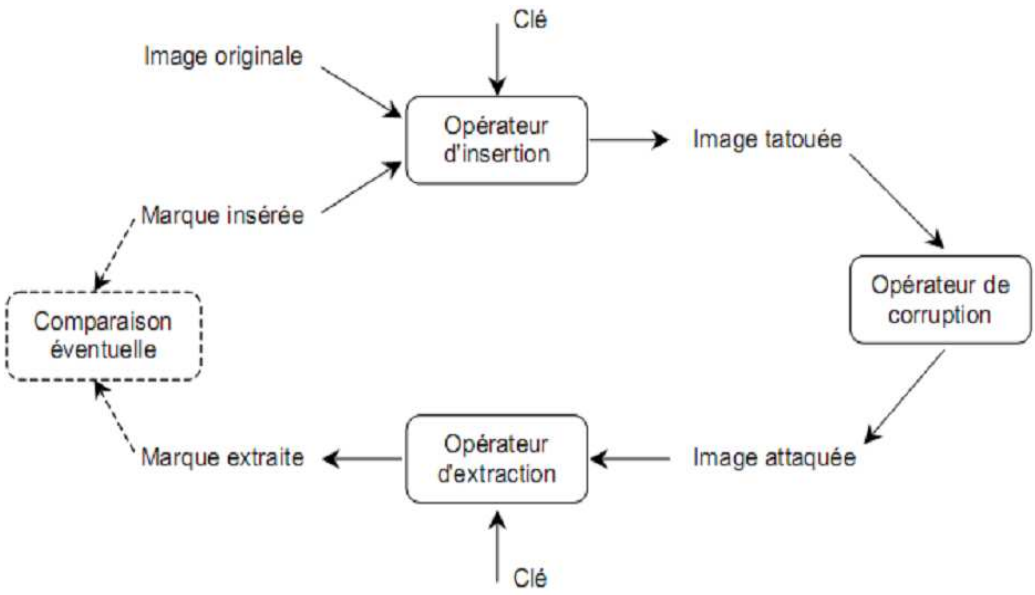


Figure 4.02: *Principe du tatouage d'image numérique*

4.2.3.2 Choix du domaine

Les systèmes de tatouage peuvent être distingués grâce à différentes caractéristiques [135], dans notre approche on a utilisé les domaines suivants :

- Tatouage fort (ou robuste) : Il s'agit de la forme la plus commune de tatouage numérique. Elle est en général imperceptible et surtout très robuste.

- Tatouage symétrique : Le parallèle avec la cryptographie prend ici toute son importance. Le tatouage symétrique signifie que l'on utilise la même clé pour insérer et détecter la marque.

- Tatouage de type I / Tatouage de type II : Le système est dit de type I si l'extraction nécessite la connaissance de la marque. L'extracteur a donc pour paramètres d'entrée le medium marqué et la marque supposée insérée et indique si cette marque est bien contenue dans le support. Si l'extracteur détermine de lui-même la marque contenue dans le support, le système est dit de type II.

- Tatouage réversible / Tatouage irréversible : Après extraction et vérification de validité de la marque, les méthodes réversibles sont capables de restituer un duplicata exact de l'image originale. Les méthodes de marquage réversibles sont des méthodes fragiles.

Généralement, on distingue les schémas de tatouage selon le domaine sur lequel ils agissent. On dispose alors essentiellement de méthodes spatiales, fréquentielles. Néanmoins, un autre critère de séparation existe en fonction de la manière dont est inséré le tatouage : soit il est ajouté au médium, soit il en remplace certains coefficients. Chaque espace de travail utilisé en tatouage possède ses propres avantages et inconvénients.

- Les méthodes agissant dans le domaine spatial modifient directement les valeurs des pixels. Comme aucun traitement initial n'est requis, ces algorithmes sont très rapides et permettent de travailler en temps réel. De plus, elles offrent souvent une bonne résistance aux transformations géométriques comme les translations ou les rotations.

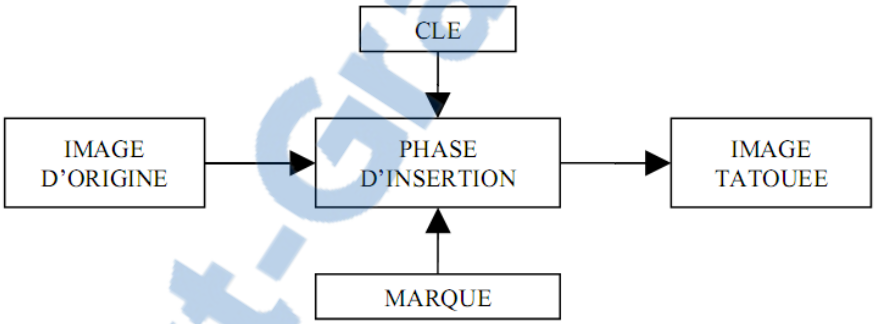


Figure 4.03: *Schéma d'insertion direct dans le domaine spatial*

- Le domaine fréquentiel

L'intérêt des schémas de tatouage s'est ensuite naturellement porté vers les domaines fréquentiels (TFD, TCD, etc.) qui sont plus stables que les domaines spatiaux.

Les schémas agissant dans ces domaines gagnent alors une certaine robustesse contre la compression.

Contrairement au domaine spatial, qui agit directement sur les pixels, les algorithmes fonctionnant avec la TCD ne sont pas très résistants aux transformations géométriques car celles-ci affectent grandement les coefficients de la TCD. Au contraire, l'espace TFD possède des propriétés d'invariance aux translations et rotations.

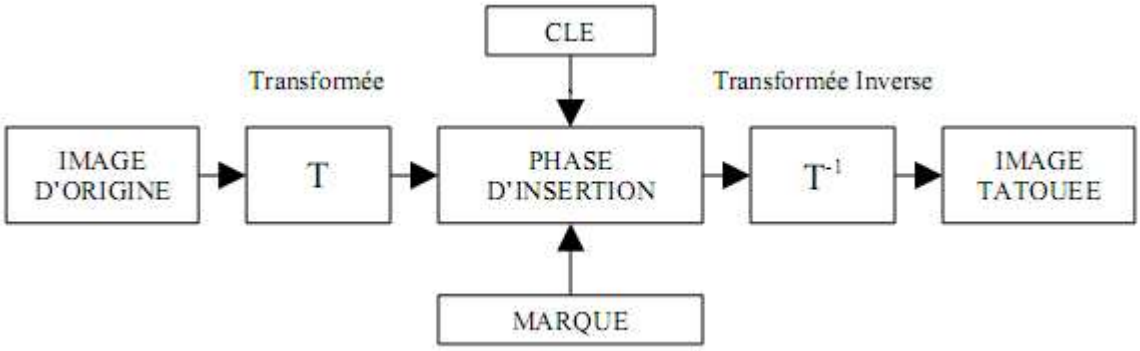


Figure 4.04: *Schéma d'insertion dans le domaine fréquentiel*

Au-delà du domaine dans lesquels ils agissent, les schémas de tatouage se distinguent essentiellement par leur manière d'insérer la marque dans le médium.

- Schémas additifs

Lors de l'insertion, le signal représentant la marque est ajouté à certaines composantes du médium. L'algorithme suivant présente le principe d'insertion par addition. La génération de la marque se fait généralement par étalement de spectre.

Algorithme 4.01 :

Exemple de schéma de tatouage par addition [134]

1. Extraire des coefficients du médium initial m

2. Réordonner ces coefficients selon une permutation paramétrée par une clé k pour obtenir un vecteur $c_k(m)$

3. Générer une marque w_k, dépendante ou non du médium initial

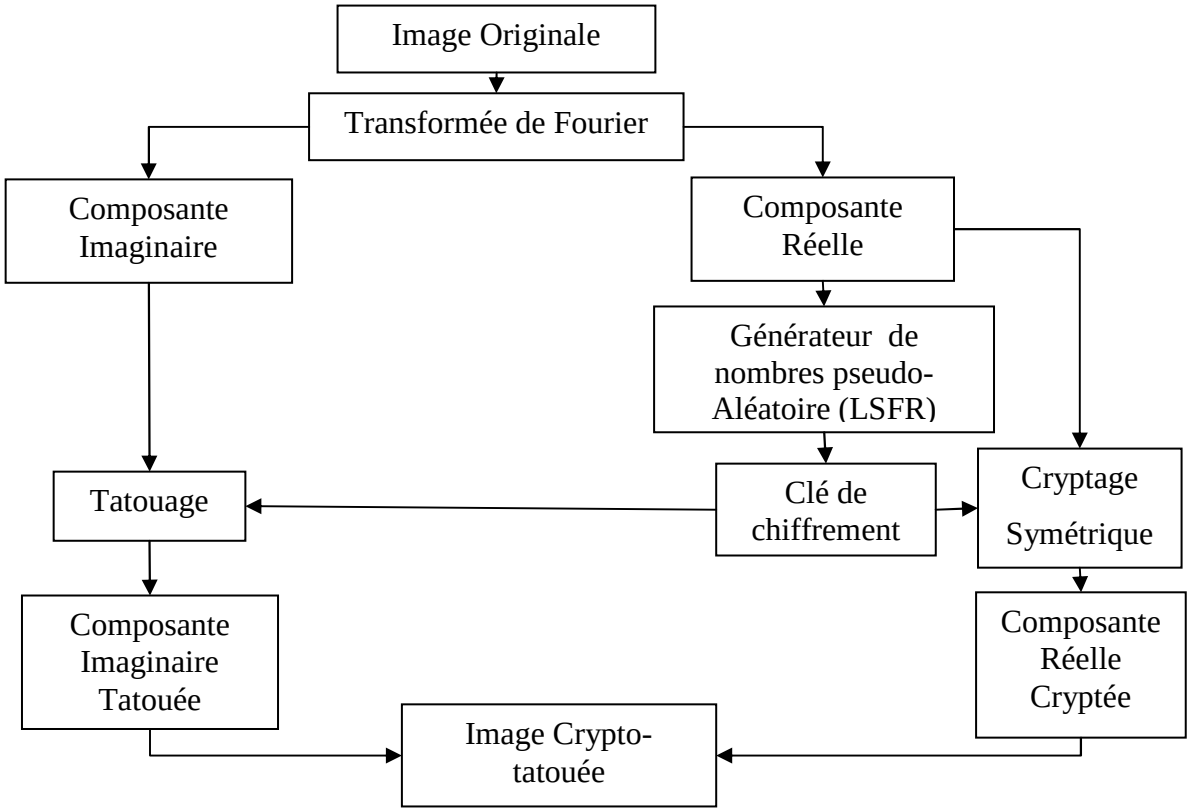
4. Tatouer le médium en ajoutant la marque aux coefficients :

$$c_k(m) = c_k(m) + w_k$$

5. Réordonner les coefficients puis reconstruire le médium tatoué m_w.

4.2.4 Approche proposé

A l'émission, nous avons le schéma des opérations suivantes :



A la réception, la restitution de l'information est donnée par le schéma ci-après :

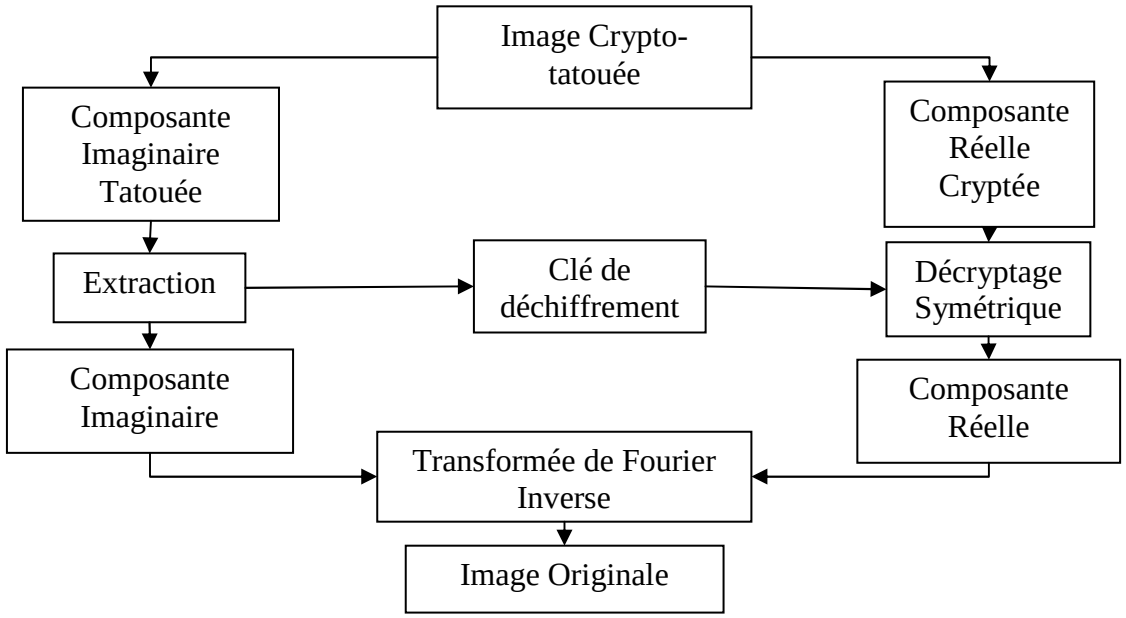


Figure 4.05: *Approche proposée : schéma à l'émission et à la réception*

4.2.5 Résultats et interprétation

Notre recherche est basée sur la combinaison de trois méthodes de traitement d'information.

Premièrement, l'information à transférer, c'est-à-dire l'image, subit un codage source qui est une compression du signal utilisée dans le but d'éliminer toutes redondances et d'optimiser la puissance de calcul. Pour cela nous avons utilisé une méthode par transformée qui est la Transformée de Fourier sur une information discrète. Le choix de ce mode est qu'il simplifie la représentation matricielle de l'image et diminue ainsi le nombre de calcul et les manipulations à faire.

Deuxièmement, nous avons mis au point un algorithme de génération de clé aléatoire qui serait en mesure de donner des clés de session servant à crypter l'information. L'algorithme utilisé est l'algorithme de cryptage symétrique AES, connu, utilisé et implémenté dans divers systèmes informatiques, du fait de sa rapidité et sa robustesse face aux différents types d'attaques connues.

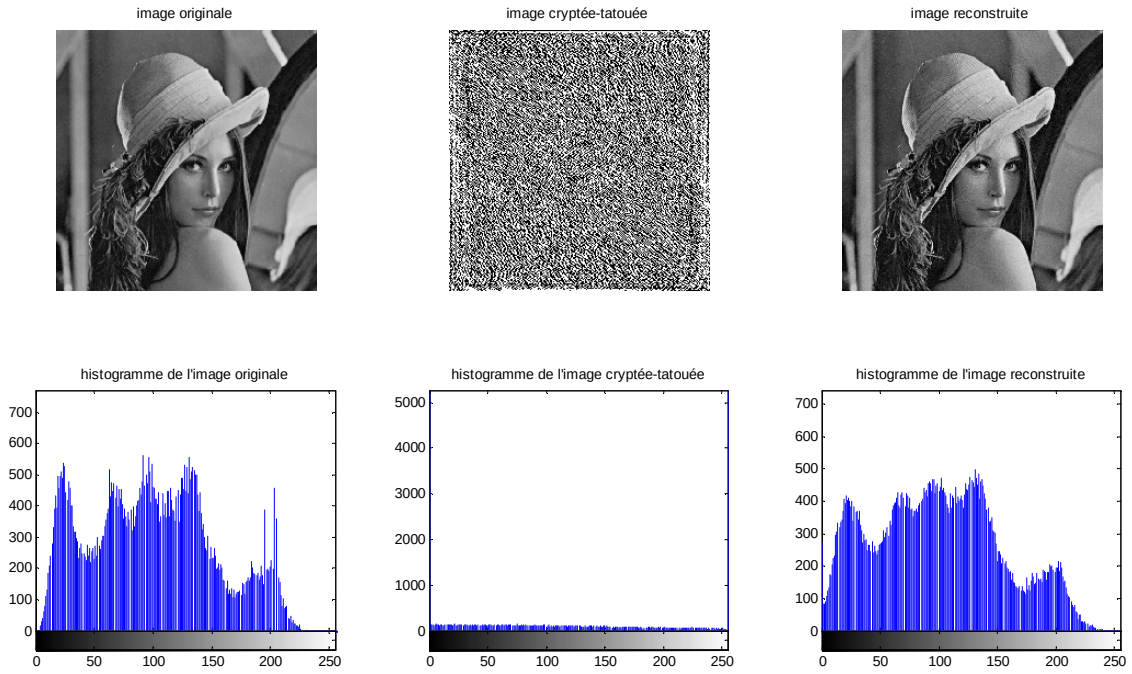


Figure 4.06: *Colonne 1 : image originale et son histogramme, colonne 2 : image cryptée, tatouée et son histogramme, colonne 3 : image reconstruite, décryptée et son histogramme*

Le système de cryptage opère seulement sur une partie de l'information, c'est ce qu'on appelle cryptage sélectif. Nous avons donc choisit une partie de l'image, obtenu après utilisation de la transformée de Fourier, qui représente les coefficients représentatifs de l'information, c'est-à-dire la partie réelle de la transformée.

Cette clé de chiffrement sera encore utilisée pour le déchiffrement au niveau du récepteur. Pour permettre un partage sécurisé de cette clé de session nous avons dissimulé celle-ci dans une autre partie de l'information à envoyer. Pour cela nous avons utilisé une technique de tatouage d'information.

Le choix du type d'insertion de données est basé sur les tatouages robustes face aux compressions et transformation géométrique comme la rotation et la translation. Pour cela nous avons opté sur l'utilisation de la méthode de tatouage additif.

Il est à noter que dans le programme, nous avons ajouté différentes techniques de diffusion et confusion pour rendre l'algorithme difficile à appréhender par un cryptanalyste mais rapide en même temps.

On constate d'après cette figure 4.06 que l'opération engendre une perte d'information équivalente à un pic du rapport signal sur bruit ou $PSNR = 31.3674\ dB$, une erreur quadratique moyenne $MSE = 47.4619$ et une maximum de déviation quadratique $maxerr = 38.9445$.

La méthode utilisée dans cette partie a été appliquée dans la partie LSB de chaque pixel, c'est pour cela que cette méthode est plutôt robuste car nous arrivons encore à extraire les clés de session après une attaque par filtrage médian et après ajout de bruits de type « salt and pepper ». Cette approche ne résiste pas face aux attaques géométriques.

La corrélation entre l'image originale est celle reconstituée est de corr=0.9933, ce qui correspond à un résultat acceptable selon son importance.

Le tableau 4.01 montre l'efficacité du programme tant à l'émission qu'à la réception. L'implémentation d'un tel programme est donc optimisée pour une utilisation sur une plateforme à ressources limitées comme les systèmes embarqués ou les appareils photos.

	Temps d'exécution du programme (s)	Temps processeur utilisé (cputime) (s)
Emission	0.4315	0.4212
Réception	0.0829	0.0936

Tableau 4.01 : *Temps d'exécution des programmes à l'émission et à la réception*

Dans une deuxième approche, représentée par la figure 4.07, l'opération n'est plus faite sur les bits les moins significatifs mais plutôt sur l'ensemble des bits. Par conséquent, nous somme confronté à beaucoup de perte d'informations à la réception du faite que nous utilisons des faibles coefficients de l'image obtenus après la Transformée de Fourier.

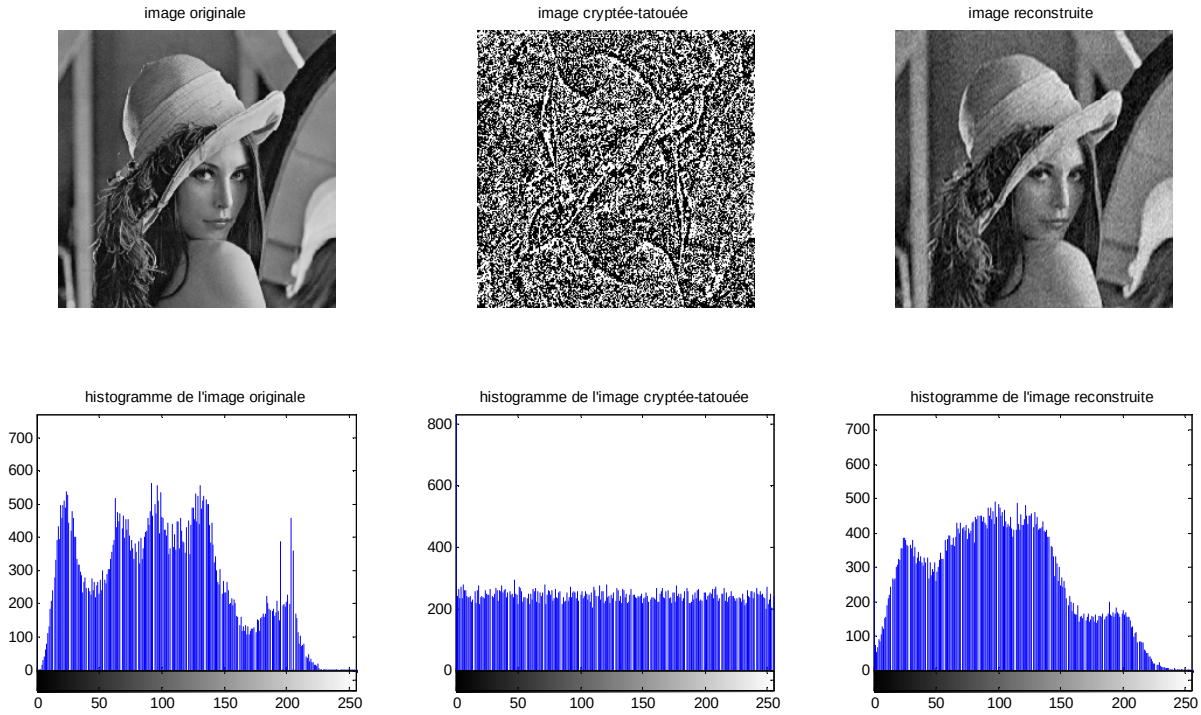


Figure 4.07: *Colonne 1 : image originale et son histogramme, colonne 2 : image cryptée, tatouée et son histogramme, colonne 3 : image reconstruite, décryptée et son histogramme*

Nous avons les résultats suivants :

PSNR (dB)	MSE	Maxerr	Corr
7.1407	1.2561 e+4	1.3641e+3	-0.0028

Tableau 4.02 : *Résultats obtenus*

Il est à noter que cette deuxième méthode n'est pas du tout résistant face aux attaques par bruitage, filtres et transformation géométrique. Après chaque attaque nous n'arrivons pas à avoir les informations tatouées.

	Temps d'exécution du programme (s)	Temps processeur utilisé (cputime) (s)
Emission	0.6518	0.5772
Réception	0.1000	0.0936

Tableau 4.03 *: Temps d'exécution des programmes à l'émission et à la réception*

Ce deuxième cas diffère juste du premier sur la manière de représenter les coefficients de la partie réelle de l'image par rapport aux clés de chiffrements, c'est-à-dire en choisissant d'opérer sur les bits le moins significatifs.

4.3 Système d'Authentification par Sécurisation d'index

Dans cette partie, nous proposons un système de sécurisation de données stockées dans une base de données d'images et un système d'authentification. Ce paragraphe a fait l'objet d'une publication dans une revue internationale, consultable sur www.ijcsn.org.

4.3.1 Indexation

En base de données, un index peut être défini comme étant une structure de données multidimensionnelle qui permet de structurer la base afin de rendre un accès plus efficace aux données. L'index d'une image est, quant à elle, son descripteur. Un descripteur est en fait une caractéristique qui capture certaines propriétés visuelles de toute l'image ou seulement d'une partie de l'image [137], [138].

4.3.1.1 Définition de l'indexation

Autrefois, les images présentes dans les bases données d'images étaient indexées par des mots clefs. Ces mots clefs, appelés également attributs textuels, étaient censés représenter le contenu de l'image, mais étant donné qu'ils devaient être insérés manuellement, l'indexation était assez subjective.

Vu le nombre croissant de ces bases de données sur Internet, ce qui signifie des millions d'images, l'indexation manuelle devient plus chère. Il est donc nécessaire de trouver une méthode qui permet d'indexer automatiquement toutes les images en fonction de leur contenu [137], [138].

L'indexation automatique d'une image consiste à calculer les indices de description de toutes les images d'une base de données d'images données. C'est un processus qui s'effectue hors ligne (*off-line*), ce qui fait que le temps de calcul n'est pas primordial. Ces indices peuvent être extraits à partir de la couleur, de la texture ou d'autres caractéristiques de l'image telles que la transformée de Fourier ou encore la transformée en ondelettes.

4.3.1.2 Indexation manuelle

L'indexation manuelle est l'approche la plus ancienne et la plus répandue de l'indexation d'images. Dans ce cas, les images sont indexées en utilisant des mots-clés. Cette méthode possède cependant plusieurs inconvénients ; dans la plupart des cas, elle est subjective et incomplète, nécessite un effort significatif et coûte cher en temps [139] [140].

4.3.1.3 Indexation automatique

L'indexation automatique, quant à elle, consiste à extraire automatiquement les descripteurs des images qui seront utilisés pour la recherche. Une signature est un vecteur numérique qui représente le contenu visuel de l'image [140].

On distingue plusieurs méthodes d'indexation automatique d'images (figure 4.08). Dans le domaine de luminance, on peut utiliser la couleur, la texture, la forme. Les domaines compressés se divisent en deux sous-catégories, à savoir, les domaines transformés et le domaine spatial. Dans les domaines transformés, les méthodes utilisées pour l'indexation utilisent la transformée de Fourier discrète (TFD), la transformée en cosinus discrète (TCD), la transformée en ondelettes discrète (DWT), la transformée de Karhunen Loève (KLT), Analyse en Composante Principale (ACP) ou encore une Décomposition en Valeurs Singulières (SVD). Dans le domaine spatial, par contre, on peut soit utiliser une méthode fractale, soit effectuer une quantification des vecteurs.

Nous n'allons pas nous étaler sur toutes les méthodes d'indexation, mais nous allons seulement citer quelques cas, à savoir :

- Dans le domaine de luminance : l'indexation par la texture et l'indexation par la couleur, et l'indexation par la forme
- Dans les domaines transformés : l'indexation en utilisant la transformée de Fourier, l'indexation en utilisant la transformée en ondelettes discrètes, l'analyse en composante principale, la SVD et la transformée de Karhunen-Loève

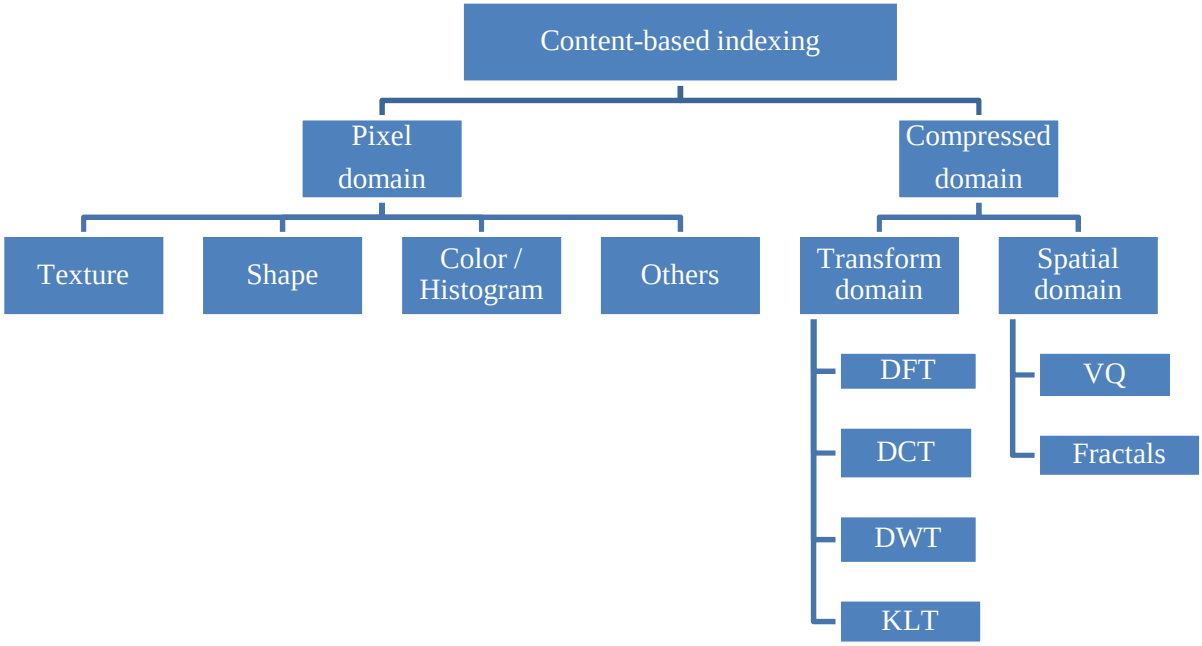


Figure 4.08: *Différents types d'indexation*

4.3.1.4 Indexation dans les domaines transformés

a. Théorie de la valeur singulière

- **Valeurs singulières d'une matrice**

Définition 4.02 :

Nous considérons un système linéaire ayant pour matrice carrée A de dimension n et de rang k. Les valeurs singulières de la matrice A de rang k, notée $\sigma_i(A)$ sont les racines carrées *non nulles* des valeurs propres de AA^T et A^TA.

Soit :

$$\sigma_i(A) = \sqrt{\lambda_i[A.A^T]} = \sqrt{\lambda_i[A^T A]} \tag{4.03}$$

- λ_i est la i^{ème} valeur propre du produit matriciel AA^T ou $A^T A$.

- $\sigma_i(A)$ est la i^{ème} valeur singulière de la matrice A.

- Elles sont réelles, positives ou nulles et sont classées par ordre de grandeur croissante :

$$\sigma_1 \gg \sigma_2 \gg \cdots \gg \sigma_k \; et \; \sigma_{k+1} = \cdots = \sigma_n = 0 \tag{4.04}$$

- La plus grande valeur singulière, appelée aussi norme spectrale, est notée $\bar{\sigma}(A)$ et représente une norme induite sur l'espace des matrices de même dimension que A.

- La plus petite valeur singulière est notée $\underline{\sigma}(A)$.

- **Décomposition d'une matrice carrée en valeurs singulières**

Théorème : Toute matrice carrée $A \in \mathbb{R}^{n\times n}$ peut se décomposer sous la forme :

$$A = U\Sigma V^T \tag{4.05}$$

où U et V sont des matrices unitaires et $\Sigma \in \mathbb{R}^{n\times n}$ est une matrice diagonale de la forme :

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{4.06}$$

Avec $\Sigma_1 = diag(\sigma_1, \dots, \sigma_k)$

Remarque : les colonnes de U sont les vecteurs propres de AA^* et les colonnes de U sont les vecteurs propres de A^*A.

- **Propriétés**

Quelques propriétés des valeurs singulières sont rappelées ci-dessous. Soient :

- $\forall\, \alpha \in C, x \in C^n \; et \; z \in C^n$
- les matrices complexes $A \in C^{n\times n}, A \in B^{n\times n}, E \in C^{n\times n} \; et \; I \in C^{n\times n}$;
- et les normes euclidiennes $\|x\|_2 = \sqrt{x^T x}$ et $\|z\|_2 = \sqrt{z^T z}$.

Propriété 4.01 :

La valeur singulière supérieure est une norme matricielle induite par la norme euclidienne de vecteur appelée aussi norme spectrale :

$$\overline{\sigma}(A) = max \frac{\|Ax\|_2}{\|x\|_2} = \max_{x\to 0} \sqrt{\frac{x^T A^T A x}{x^T x}} \tag{4.07}$$

La valeur singulière supérieure possède donc toutes les propriétés d'une norme induite par une norme vectorielle, en particulier :

$$\overline{\sigma}[\alpha A] = |\alpha| \overline{\sigma}[A] \tag{4.08}$$

$$\overline{\sigma}[A + B] \leq \overline{\sigma}[A] + \overline{\sigma}[B] \tag{4.09}$$

$$\overline{\sigma}[A\, B] \leq \overline{\sigma}[A]\, \overline{\sigma}[B] \tag{4.10}$$

Propriété 4.02 :

La valeur singulière inférieure est une norme matricielle induite par la norme euclidienne de vecteur.

$$\underline{\sigma}(A) = max \frac{\|Az\|_2}{\|z\|_2} = \max_{x\to 0} \sqrt{\frac{z^T A^T A z}{z^T z}} \tag{4.11}$$

La valeur singulière inférieure possède donc toutes les propriétés d'une norme induite par une norme vectorielle, en particulier :

$$\underline{\sigma}[\alpha A] = |\alpha|\underline{\sigma}[A] \tag{4.12}$$

$$\underline{\sigma}[A + B] \leq \underline{\sigma}[A] + \underline{\sigma}[B] \tag{4.13}$$

$$\underline{\sigma}[A\ B] \leq \underline{\sigma}[A]\underline{\sigma}[B] \tag{4.14}$$

Propriété 4.03 :

Si A est inversible :

$$\underline{\sigma}[A] \neq 0 \text{ et } \overline{\sigma}[A] \neq 0 \tag{4.15}$$

$$\underline{\sigma}[A] = \frac{1}{\overline{\sigma}[A^{-1}]} \tag{4.16}$$

$$\overline{\sigma}[A] = \frac{1}{\underline{\sigma}[A^{-1}]} \tag{4.17}$$

Propriété 4.04 :

Soit $\lambda_i(A)$ les valeurs propres de la matrice A. $\forall i = 1, \dots, n$, nous avons :

$$\underline{\sigma}(A) \leq \lambda_i(A) \leq \overline{\sigma}(A) \tag{4.18}$$

Propriété 4.05 :

$$\overline{\sigma}[E] < \underline{\sigma}[A] \Rightarrow \det[A + E] \neq 0 \tag{4.19}$$

$$\det[A + E] = 0 \Rightarrow \overline{\sigma}[E] \geq \underline{\sigma}[A] \tag{4.20}$$

Propriété 4.06 :

$$\forall i = 1, \dots, n \qquad \sigma_i(A) - 1 \leq \sigma_i(I + A) \leq \sigma_i(A) + 1 \tag{4.21}$$

Propriété 4.07 :

Toute matrice complexe $A \in C^{n \times n}$ admet une décomposition en valeurs singulières qui s'écrit :

$$A = V_u \Sigma W_u^T \tag{4.22}$$

avec $\Sigma = diag\{\sigma_1, \dots, \sigma_n\}$

où V_u et W_u sont des matrices unitaires :

$$V_u V_u^T = V_u^T V_u = I, et\ W_u W_u^T = W_u^T W_u = I \tag{4.23}$$

b. Propriétés de la Décomposition de la Valeur Singulière (SVD) d'une image

Les valeurs singulières sont les plus importantes car elles sont uniques, ce sont les plus importants attributs de la matrice image, $\sigma_1, \sigma_2, \dots, \sigma_n$ sont uniques, mais $U\ et\ V$ ne les sont pas [141], [142].

Le rang de la matrice A est égale au nombre de leur valeurs singulières non nuls.

Les Valeurs Singulières représentent l'énergie de l'image, c'est-à-dire que la SVD range le maximum d'énergie de l'image dans un minimum de Valeurs Singulières [142]

Les valeurs singulières d'une image ont une très bonne stabilité, c'est-à-dire, quand une petite perturbation est ajoutée à une image, les valeurs singulières ne change pas significativement.

c. Méthode de la nouvelle approche avec SVD

Premièrement, la méthode d'indexation, c'est-à-dire la phase de l'extraction des informations essentielles de l'image à présenter est divisée en deux parties :

- Dissociation de la taille de l'image originale par trois : Rouge, Vert, Bleu en ne conservant qu'un seul canal qui range le maximum d'énergie de l'image dans les valeurs singulières par ordre décroissant et dans un minimum de valeurs singulières car chaque image est en couleur Rouge, Vert, Bleu.

  Chaque pixel, chaque site (x, y) contient une information couleur sur l'intensité du rouge, l'intensité du vert, l'intensité du bleu.

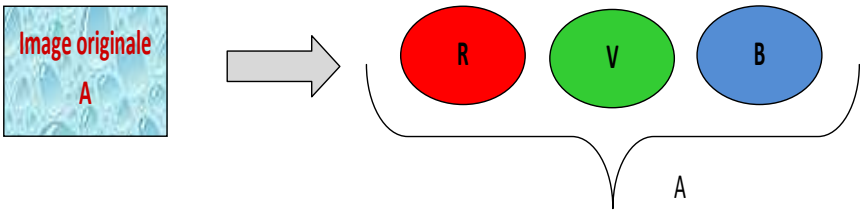


Figure 4.09: *Division des composantes couleurs de l'image*

- Décomposition en SVD de chacune des composantes couleurs de l'image :

$$A = U * S * V^T \tag{4.24}$$

A	=	U	S	V^T
$(m * n)$		$(m * m)$	$(m * n)$	$(n * n)$

$$U = [u_1, u_2, \dots, u_k, u_{k+1}, \dots, u_m] \tag{4.25}$$

$$u_i, \quad i = 1, 2, \dots, m$$

$$u_i^T u_j = \delta_{ij} = \begin{pmatrix} 1 & i = j \\ 0 & i \neq j \end{pmatrix} \tag{4.26}$$

$$V = [v_1, v_2, \dots, v_k, v_{k+1}, \dots, v_n] \tag{4.27}$$

$$v_i, i = 1, 2, \dots, n$$

$$v_i^T v_j = \delta_{ij} = \begin{pmatrix} 1 & i = j \\ 0 & i \neq j \end{pmatrix} \tag{4.28}$$

$$S = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{pmatrix} \tag{4.29}$$

σ_i sont appelés Valeurs Singulières de A.

Pour $i = 1, 2, \dots, n$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0 \tag{4.30}$$

Remarque

Une fois que nous avons pu extraire les informations maximales de l'image, et les avoir stockées dans les minimums de Valeurs Singulières ; nous pouvons appliquer à cette partie divers opérations et traitements pour un usage ultérieur, par exemple la cryptographie.

- **L'analyse en composantes principales**

Le problème consiste à dissocier puis stocker les composantes RVB (rouge, vert, bleu) de l'image dans les première, seconde et troisième composantes principales [139].

- *Calcul des variables et espace d'étude :*

Une série à variables individuelles ; soit l'étude de la variable X, une série de valeurs définies dans $\mathbb{R}$. Nous avons les formules suivantes :

- Moyenne :

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4.31}$$

- Variance :

$$\theta^2 = V(X) = \sum_{i=1}^{N} (x_i - \bar{X})^2 \tag{4.32}$$

- L'écart-type :

$$\sigma = \sqrt{V(X)} \tag{4.33}$$

On a deux séries de valeurs définies dans $\mathbb{R}$:

- Covariance :

$$\theta_{xy} = cov(X,Y) + \sum_{i=1}^{N} (x_i - \bar{X})(y_i - \bar{Y}) \tag{4.34}$$

- Corrélation :

$$cor(X,Y) = \frac{\theta_{xy}}{\theta_x - \theta_y} = \frac{cov(X,Y)}{\sqrt{cov\,(X)}\sqrt{cov(Y)}} \tag{4.35}$$

Matrice des covariances :

Cela revient à calculer la matrice centrée réduite, une matrice carrée, une relation affine.

$$P = \begin{bmatrix} var(R) & cov(R,G) & cov(R,B) \\ cov(G,R) & var(G) & cov(G,B) \\ cov(B,R) & cov(B,G) & var(B) \end{bmatrix} \tag{4.36}$$

Matrice des vecteurs propres :

$$V = \begin{bmatrix} 0{,}614 & 0{,}588 & 0{,}526 \\ -0{,}581 & -0{,}114 & 0{,}806 \\ 05346 & -0{,}801 & 0{,}271 \end{bmatrix} \tag{4.37}$$

Matrices des valeurs propres :

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{4.38}$$

- *Recherche de valeurs propres et vecteurs propres :*

L'image de base est en couleur : rouge, vert, bleu (RVB). Chaque pixel, chaque site (x, y) contient une information couleur sur l'intensité du rouge, du vert, du bleu. Il est donc possible de diviser par trois, la taille d'une image, en ne conservant qu'un seul canal ($(RVB \rightarrow C1)$, $(x, y, z) \rightarrow x'$). On obtient d'une part, une réduction de l'espace couleur et d'autre part la conservation de l'axe permet de conserver le plus d'information, d'avoir 90% de l'information.

$$\frac{\lambda_1}{\sum_{j=1}^{3} \lambda_j} > T\ (0{,}90) \tag{4.39}$$

On définit algébriquement, sur un échantillon de « n » observations sur un vecteur de « p » variables.

$$X = \left(x_1, x_2, \dots, x_p\right) \tag{4.40}$$

La première composante principale de l'échantillon par transformation linéaire:

$$\begin{gathered} Z_1 = a_1^T X = \sum_{i=1}^{p} a_{i1}\, x_i \\ a_1 = (a_{11}, a_{21}, \dots, a_{P1}) \\ var[Z_1] \ \text{Maximum} \end{gathered} \tag{4.41}$$

La $k^{ième}$ composante principale de l'échantillon se déduit par transformation linéaire :

$$Z_k = a_k^T X k = 1, \dots, p \tag{4.42}$$

Où le vecteur (coefficients vectoriels)

$$a_k = (a_{1k}, a_{2k}, \dots, a_{Pk}) \tag{4.43}$$

Est choisi tel que :

$$var[Z_k] \ \text{Maximum} \tag{4.44}$$

Et est sujet à

$$cov[Z_k\,, Z_l] = 0 \text{ pour } \mathrm{k} > l \geq 1 \; et \text{ à } \; a_k^T a_k = 1 \tag{4.45}$$

Pour la matrice de covarianceΛ, les variables $X = \left(x_1, x_2, \ldots, x_p\right)$, a_1 est un vecteur propre de S , correspondant à la valeur propre $\lambda \equiv \lambda_1 \; var[Z_1] = \lambda_1$ avec λ_1, la plus grande valeur propre de S ; a_k est aussi vecteur propre de S. La $k^{ième}$ valeur propre (la plus grande de S) est la variance de la $k^{ième}$ composante principale.

$$\begin{aligned} \Lambda &= A^T S A \\ \Lambda_{ij} &= \lambda_i \delta_{ij} \end{aligned} \tag{4.46}$$

En général, la réduction du nombre de variables utilisées pour décrire un ensemble de données provoque une perte d'information. L'ACP procède de façon à ce que cette perte d'information soit la plus faible possible, selon un sens précis et naturel que l'on donnera au mot "information".

L'Analyse en Composantes Principales peut donc être vue comme une technique de réduction de dimensionnalité. Quant au nombre de composantes principales à utiliser, bien que l'objectif soit en général de n'utiliser qu'un petit nombre de Composantes Principales, l'ACP en construit initialement autant que de variables originales.

Ce n'est que par la suite que l'on décidera du nombre de Composantes à retenir. « Retenir k Composantes Principales » veut dire « remplacer les observations originales par leur projections orthogonales dans le sous-espace à k dimensions défini par les k premières Composantes Principales ».

Comme précédemment, une fois qu'on ait pu extraire les informations maximales de l'image, et les avoir stockées dans les trois composants couleur de l'image, on peut appliquer sur l'une des composantes diverses opérations et traitements pour un usage ultérieur, par exemple la cryptographie.

- **Calcul de la similarité**

Etape 1 :

Au sens d'une métrique donnée, on propose ici, par la distance euclidienne des histogrammes des composantes principales de l'image requête et les images de la base, c'est la phase de la recherche d'image. Ce calcul fournit : les images réponses. C'est-à-dire classer les images.

Etape 2 :

La question qui se pose c'est : Quelle est l'image de la base la plus similaire à la requête ? Le système adresse les meilleures images au sens de la mesure de similarité ;

L'ACP de base suppose que l'on utilise la distance euclidienne, et que tous les individus sont munis du même « poids » unité.

Ces deux hypothèses peuvent être remplacées par des hypothèses plus générales. Selon la similarité visuelle entre deux images, soient $\mathrm{C1}$, $\mathrm{C2}$, $\mathrm{C3}$ et $\mathrm{C1}'$, $\mathrm{C2}'$, $\mathrm{C3}'$ les descripteurs de deux images I et I', notons x, les descripteurs formés de $\mathrm{C1}$, $\mathrm{C2}$, $\mathrm{C3}$ et x' ceux de $\mathrm{C1}'$, $\mathrm{C2}'$, $\mathrm{C3}'$: $x = x(I)$ et $x' = x(I')$

Soit :

$$D = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{4.47}$$

La mesure de similarité revient à calculer $\mathrm{d}\,(\mathrm{x}, \mathrm{x}')$ c'est-à-dire :

$$\mathrm{d}\,(\mathrm{C1}, \mathrm{C1}'), \mathrm{d}\,(\mathrm{C2}, \mathrm{C2}'), \mathrm{d}\,(\mathrm{C3}, \mathrm{C3}')$$

Plus précisément, nous calculons d'abord l'histogramme des composantes principales, une fois ces dernières trouvées, notons ACP1, ACP2, ACP3 ces histogrammes, ACP1s, ACP2s, ACP3s ceux des composantes principales des images dans la base d'image.

La similarité ici retourne la distance euclidienne calculée entre les histogrammes de toutes les ACP concernés.

$$D = \sqrt{\sum_{i=1}^{n} (ACP_i - ACPs_i)^2} \quad (4.48)$$

Une distance **nulle** signifie que les images sont similaires.

4.3.2 Systèmes cryptographiques

Dans cette section, nous allons chiffrer les informations suivant l'algorithme de chiffrement RSA proposé dans le chapitre 3, d'une part.

D'autre part, nous allons appliquer les algorithmes de calcule de points proposés dans le chapitre précédent sur un schéma de chiffrement basé sur les courbes elliptiques de El Gamal.

- Cryptage sur une courbe elliptique : algorithme de El Gamal

Le problème du logarithme discret sur les courbes elliptiques est plus difficile, à taille égale, que le problème du logarithme discret classique. Il est donc tentant de transcrire les cryptosystèmes basés sur le logarithme discret en termes de courbe elliptiques, pour obtenir des cryptosystèmes de sécurité équivalente avec des clés plus petites.

Algorithme 4.02 : Chiffrement

Pour envoyer un message à Bob, Alice procède ainsi :

1) Utilise la clé publique (P, B) de Bob
2) Représenter le message comme un point M
3) Choisir un entier k et calculer $M_1 = kP$
4) Calculer $M_2 = M + kB$
5) Envoyer M_1 et M_2 à Bob

Algorithme 4.03 : Déchiffrement

Bob décrypte le message en calculant

$$M = M_2 - sM_1$$

4.3.3 Approche proposée

L'organigramme des opérations est donné par la figure 4.10 suivante :

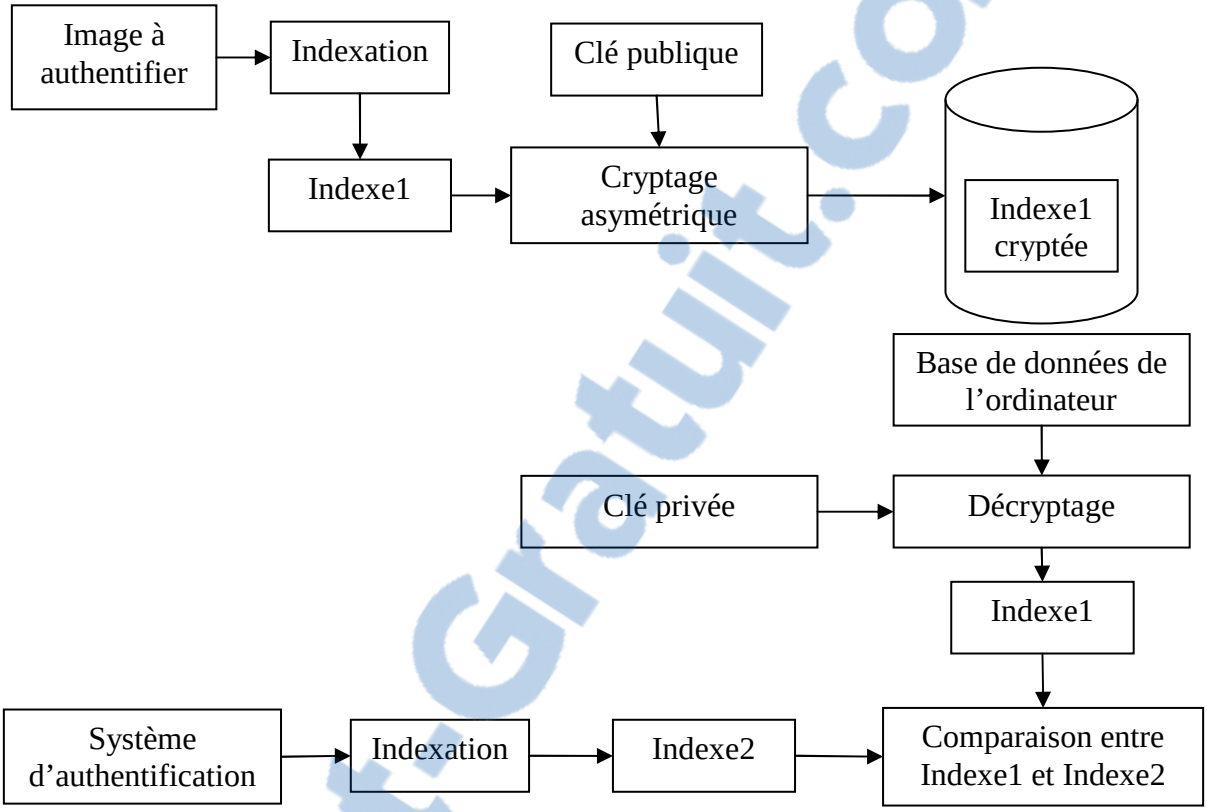


Figure 4.10: *Approche proposée : principe de comparaison d'indexe*

4.3.4 Résultats et interprétations

Dans cette approche, notre but est de tester un nouvel système d'authentification qui consiste à la fois à sécuriser les informations dans une base de données et de ne stocker que les informations représentatives de celles-ci.

La méthode se subdivise en deux parties, à savoir :

- Premièrement, nous allons faire des traitements sur les informations à stocker dans la base ; l'image ou plus précisément les empreintes digitales subissent des opérations d'indexations dans le domaine de la transformée ACP ou SVD. Après décomposition de l'image en trois composantes selon la méthode choisie, nous appliquons un cryptage asymétrique RSA ou un cryptage utilisant les courbes elliptiques sur la

composante présentant l'information représentative de l'image. L'objectif du cryptage est de donner une confidentialité et une intégrité de ces index. Cette partie cryptée sera donc stockée dans la base de données en attendant la comparaison avec l'image requête.

- Deuxièmement, pour s'authentifier, l'utilisateur utilise un dispositif électronique permettant d'obtenir son empreinte digitale. Après prélèvement, cette image requête passe par des opérations d'indexation identique aux traitements de l'image stockées. Pour permettre une authentification, on déchiffre les données dans la base et on les compare avec l'index de l'image requête. La comparaison se fait par calcul de la similarité des deux indexes en calculant sa distance euclidienne.

Les empreintes digitales utilisées dans l'expérience sont :

					
ed1	ed2	ed3	ed4	ed5	ed6
Indexation SVD	Indexation SVD	Indexation SVD	Indexation ACP	Indexation ACP	Indexation ACP

Figure 4.11: *Empreintes digitales utilisées et méthodes d'indexation correspondantes*

Avec l'indexation SVD, nous obtenons trois matrices de telle manière que c'est la matrice S qui est représentative de l'image et c'est celle-ci que nous allons crypter.

De même, avec l'indexation ACP, nous obtenons trois composantes ACP1, ACP2 et ACP3 telles que ACP1 représente l'information représentative de l'image.

Nous avons dans le tableau 4.04 la similarité entre chaque index d'image. Pour comparaison, nous avons utilisé en même temps les deux techniques d'indexations.

	ed1	ed2	ed3	ed4	ed5	ed6
ed1	0	6.7311E+3	1.4921E+3	5.0263E+4	7.0529E+4	8.1675E+4
ed2	6.7311E+3	0	7.5955E+3	4.4304E+4	6.4311E+4	7.7295E+4
ed3	1.4921E+3	7.5955E+3	0	5.0727E+4	7.0966E+4	8.2003E+4
ed4	5.0263E+4	4.4304E+4	5.0727E+4	0	2.6127E+4	5.9028E+4
ed5	7.0529E+4	6.4311E+4	7.0966E+4	2.6127E+4	0	5.6401E+4
ed6	8.1675E+4	7.7295E+4	8.2003E+4	5.9028E+4	5.6401E+4	0

Tableau 4.04 : *Valeurs des similarités entre les index*

Le tableau 4.04 ci-dessus montre l'efficacité de notre approche vu que la probabilité de rencontrer deux empreintes digitales identiques pour deux personnes différentes tend vers la valeur nulle.

Il est à noter que notre approche est invariante face à des transformations géométriques telles que la rotation, le redimensionnement et la translation de l'image requête. Nous pouvons aussi voir le temps d'exécution des deux modes d'indexation dans le tableau suivant :

	Temps d'indexation	Temps de cryptage	Temps de décryptage
Méthode SVD	0.1050	0.0421	0.0442
Méthode ACP	0.0828	0.0346	0.0488

Tableau 4.05 : *Comparaison en temps d'indexation, de cryptage et de décryptage des deux méthodes d'indexation*

En indexant une même image, la méthode d'indexation par ACP représente un large avantage sur la rapidité de calcul par rapport à la méthode d'indexation par SVD du fait que cette dernière effectue des opérations supplémentaires sur l'extraction du niveau d'énergie de l'image. Par contre, La méthode par SVD présente une haute efficacité sur la précision des signatures, c'est-à-dire le maximum d'énergies représentatives de l'information.

4.4 Contribution à la sécurisation des communications sur la VOIP

4.4.1 VOIP

La **voix sur IP**, ou « **VoIP : Voice over Internet Protocol** », est une technique qui permet de communiquer par la voix sur des réseaux compatibles IP, qu'il s'agisse de réseaux privés ou d'Internet, filaire (câble/ADSL/optique) ou non (satellite, wifi, GSM) . Cette technologie est notamment utilisée pour prendre en charge le service de téléphonie sur IP (« ToIP » pour *Telephony over Internet Protocol*) [144].

4.4.1.1 Le transport de la voix

Le transport de la voix sur IP est relativement complexe. La première étape est la numérisation du signal acoustique. C'est le codage. Ensuite, les informations sont découpées en trame pouvant circuler sur un réseau informatique. Divers protocoles peuvent alors être utilisés pour acheminer les informations au(x) destinataire(s). Ainsi le protocole RTCP (Real-time Transport Control Protocol) est utilisé pour contrôler le transport des paquets RTP (Real-time Transport Protocol).

4.4.1.2 Le codage de la voix

La voix produit, lorsqu'on parle, des changements permanents de fréquences. Grâce à ces changements de fréquence, nous pouvons laisser percevoir des émotions et des intonations. Cette multitude de fréquences provoquerait un affichage de plusieurs fréquences sinusoïdales si on les observait sur un oscilloscope.

La voix provoque donc une superposition de signaux sinusoïdaux, c'est-à-dire analogiques. Pour l'envoyer sur un réseau TCP/IP (Numérique), il va falloir convertir ce signal analogique en un signal numérique en format PCM (Pulse Code Modulation) à 64 kbps.

Une fois convertie, il faut compresser la voix ainsi numérisée par le biais d'un codec (Compresseur/Décompresseur) pour l'insérer dans un paquet IP. Le codage doit offrir une qualité de voix la meilleure possible pour un débit et un délai de compression les plus faibles possibles.

Dans la VoIP, Il existe plusieurs techniques de codage, chacune étant mesurée de façon totalement subjective par une masse de population prise au hasard. Elle doit noter chaque codage par un chiffre de 1 à 5 (1 = Insuffisant et 5 = Excellent).

Nom du Codec	Débit du Codec	Note du Codec
G.711 (PCM)	~64 kb/s	4.1
G.723.1	~6,4 kb/s	3.9
G.726	~32 kb/s	3.85
G.729	~8 kb/s	3.92

Tableau 4.06 : *Propriétés des CODEC*

4.4.1.3 Real-time Transport Protocol

RTP ou Real-time Transport Protocol est le protocole basé sur UDP/IP qui permet de transporter la voix sur IP [145].

Le protocole RTP a pour but d'organiser les paquets à l'entrée du réseau et de les contrôler à la sortie. Ceci de façon à reformer les flux avec ses caractéristiques de départ. RTP est un protocole de bout en bout, volontairement incomplet et malléable pour s'adapter aux besoins des applications. Il sera intégré dans le noyau de l'application. Il laisse la responsabilité du contrôle aux équipements d'extrémité. Il est aussi un protocole adapté aux applications présentant des propriétés temps réel.

Il permet ainsi de :

- Mettre en place un séquencement de paquets par une numérotation et ce afin de permettre ainsi la détection des paquets perdus. Ceci est un point primordial dans la reconstitution des données. Mais il faut savoir quand même que la perte d'un paquet n'est pas un gros problème si les paquets ne sont pas perdus en trop grands nombres. Cependant il est très important de savoir quel est le paquet qui a été perdu afin de pouvoir pallier à cette perte.

- Identifier le contenu des données pour leurs associer un transport sécurisé et reconstituer la base de temps des flux (horodatage des paquets : possibilité de resynchronisation des flux par le récepteur)

- L'identification de la source c'est à dire l'identification de l'expéditeur du paquet. Dans un multicast l'identité de la source doit être connue et déterminée.

- Transporter les applications audio et vidéo dans des trames (avec des dimensions qui sont dépendantes des codecs qui effectuent la numérisation). Ces trames sont incluses dans des paquets afin d'être transportées et doivent, de ce fait, être récupérées facilement au moment de la phase de segmentation des paquets afin que l'application soit décodée correctement.

4.4.1.4 Les modes de diffusion [144]

Le terme « VoIP » est en général utilisé pour décrire des communications « point à point ». Pour la diffusion du son ou de vidéos sur IP en multipoints, on parlera plutôt de *streaming* pour une simple diffusion, comme les Web radios par exemple. Le terme multipoints sera réservé à des visioconférences dont le nombre de participants est plus grand que deux.

La voix ou le son sur IP peut se faire en mode *Unicast*, *broadcast* ou *Multicast* sur les réseaux, c'est-à-dire en mode « point à point », en mode « une émission et plusieurs réceptions » (comme un émetteur TV, par exemple) et en mode « une émission pour plusieurs réceptions » (mais le signal n'est routé que s'il y a des récepteurs) comme les radios Web.

Le transport de communication sur IP est très dépendant du délai de latence d'un réseau. Ce délai influe beaucoup sur la qualité psycho-acoustique d'une conversation. Avec l'avènement des réseaux 100 Mbps et xDSL, les temps de latence deviennent acceptables pour une utilisation quotidienne de la voix sur IP.

À l'inverse, les connexions par liaison satellite souffrent d'un temps de latence souvent trop important pour prendre en charge les applications de voix sur IP. En moyenne, le temps de latence sur ce type de liaison est estimé entre 400 et 800 millisecondes. Une connexion filaire (fibre optique ou cuivre) bénéficie d'un temps de latence de 60 à 200 millisecondes.

Plus que la latence, c'est la gigue (jitter) qui pénalise la voix sur IP. En effet, s'il y a des fluctuations du signal en amplitude et fréquence, il faudra un mécanisme de remise en ordre des paquets afin de restituer le message vocal, processus qui se traduira par des blancs et des attentes.

4.4.1.5 Le protocole SIP

SIP, ou Session Initiation Protocol, est un protocole de commande de couche application qui peut établir, modifier et terminer des sessions multimédia (conférences) telles que des communications téléphoniques par l'Internet. SIP peut aussi inviter des participants à des sessions déjà existantes, telles que des conférences en multidiffusion. Des supports peuvent être ajoutés (et retirés) à une session existante. SIP prend en charge de façon transparente la transposition de nom et les services de redirection, ce qui sert de support à la mobilité personnelle [145], les utilisateurs peuvent conserver une identification unique vue de l'extérieur, indépendamment de leur localisation dans le réseau.

SIP prend en charge cinq facettes de l'établissement et de la terminaison de communications multimédia :

- Localisation de l'utilisateur : détermination du système terminal à utiliser pour la communication ;
- Disponibilité de l'utilisateur : détermination de la volonté de l'appelé à s'engager dans une communication ;
- Capacités de l'utilisateur : détermination du support et des paramètres de support à utiliser ;
- Etablissement de session : "sonnerie", établissement des paramètres de session à la fois chez l'appelant et l'appelé ;
- Gestion de session : y compris le transfert et la terminaison des sessions, la modification des paramètres de session, et l'invocation des services.

Les principaux protocoles utilisés pour l'établissement des connexions en voix sur IP sont :SIP, H.323, IAX, Jingle, MGCP, SCCP, UA/NOE, UNISTIM.

4.4.1.6 Perte de paquets (Packet Loss)

Lorsqu'il y a saturation, les mémoires tampons ont besoin de libérer une partie de la bande passante, négligeant ainsi certains paquets. Néanmoins, le trafic VoIP est transmis au dessus de la couche UDP, ce qui implique qu'aucun mécanisme de contrôle de flux ou de retransmission des paquets perdus n'est offert par la couche transport. Cela implique qu'il faut accorder une forte importance aux protocoles RTP et RTCP (Real-Time Transport Control Protocol) qui vont permettre de calculer le taux de perte de paquets, et de réagir en conséquence au niveau de la couche applicative.

4.4.1.7 Qualité de Service (QoS) [144]

a. Latence

La latence est le décalage entre le temps écoulé entre l'envoi d'un paquet et sa réception par le destinataire. Plus le temps de latence est important, plus le transfert est long et sera donc décalé. Pour garantir une communication optimale, la maîtrise du délai de transmission est un point important afin de réduire l'effet d'écho ou la sensation de voix métallique. Le temps de transmission de paquets dans un réseau de type IP dépend de nombreux éléments tels que : le nombre d'équipements actifs traversés dans le réseau, le débit de transit disponible et le délai de propagation de l'information.

b. Variation du délai (gigue)

La gigue est la variation du délai de transmission d'un paquet de bout en bout. Cela correspond à la différence du temps entre le moment où deux paquets auraient dû arriver et le véritable moment de leurs arrivées respectives. La cause de ce problème peut être due à la différence des chemins empruntés par les paquets dans le réseau, une congestion ponctuelle du réseau, ou encore un souci d'encapsulation des paquets IP.

c. Bande passante

C'est le terme utilisé pour évoquer le flux de connexion. C'est une unité souvent prise pour une unité de débit, mais elle ne définit en réalité que la plage de fréquence et le débit en dépend.

4.4.1.8 Sécurité

La sécurité de la téléphonie est souvent restée un sujet à part dans l'entreprise. Désormais, les risques associés aux systèmes téléphoniques peuvent avoir des conséquences graves sur le système d'information et sur l'entreprise. Par ailleurs, la liste des failles historiques de la téléphonie est toujours d'actualité maintenant qu'elle intègre les standards IP [17].

VoIPshield Laboratories, entreprise spécialisée dans la sécurité des systèmes VoIP, a découvert en novembre 2008 une faille de sécurité au sein du protocole RTP qui permettrait de mener des attaques par déni de services sur les utilisateurs de logiciel utilisant le protocole RTP, soit près de 250 millions d'ordinateurs dans le monde.

4.4.1.9 Aspect logiciel

Avec la banalisation des réseaux haut débit, le nombre d'applications possibles a considérablement augmenté. Les applications de VoIP (*Voice over IP*) sont une des nouvelles possibilités offertes.

L'augmentation des débits et les connexions permanentes offrent des possibilités de développement de la voix sur IP (*Internet Protocol*). C'est dans cet intérêt que notre étude se repose.

Le développement de la VoIP a entraîné les concepteurs de plates-formes de programmation à développer des API (*Application Programming Interface*) spécifiques à la voix sur IP. L'intégration de nouveaux besoins dans une plate-forme de développement permet d'attirer les concepteurs de logiciels qui doivent intégrer des fonctions de voix sur IP dans leurs applications. Elles implémentent le protocole SIP.

Les API de VoIP peuvent être utilisées dans de nombreuses applications, la plus simple étant les téléphones logiciels (*softphones*). D'autres applications peuvent intégrer de la VoIP comme besoin secondaire. Par exemple les applications de messagerie instantanée qui intègrent de plus en plus souvent la possibilité de parler directement avec ses contacts ou bien toutes les applications nécessitant une interaction textuelle entre les différentes applications clientes comme les jeux vidéo.

4.4.1.10 PABX et Asterisk

Un PABX est un autocommutateur téléphonique privé (Private Automatic Branch eXchange) [145].

En d'autres termes, il représente l'élément central qui :

1) distribue les appels téléphoniques arrivés,

2) autorise les appels téléphoniques départs,

3) gère les terminaux téléphoniques,

4) gère toutes les autres fonctionnalités ou options.

5) permet les appels entre postes internes sans utiliser le réseau public,

6) programme les droits d’accès au réseau,

7) Gère la ventilation de la facturation par service (taxation).

Un autocommutateur privé possède sa propre intelligence pour faciliter la commutation des appels voix. Cette intelligence est gérée par au moins une unité centrale**(**CPU**),** avec des processeurs d'entrées/sorties qui gèrent les interfaces de lignes et d'équipements de postes, avec également une mémoire vive.

Il existe deux sortes de PABX :

- Les PABX traditionnels (qui peuvent éventuellement migrer partiellement ou totalement en IP),
- Et les PABX-IP ou IPBX ou PBXIP (qui nativement offrent une connectivité IP Ethernet).

Asterisk est un logiciel qui, installé sur un PC, fait office de PABX. C’est un logiciel libre (Open Source), publié sous licence GPL et crée par Mark Spencer de la société Digium.

Il tourne sur Linux, BSD et Mac OS X.

Asterisk offre tous les services de téléphonie « classiques » d'un PBX ainsi que des fonctions avancées :

- Boîte vocale (avis par courriel de réception d'un message vocal, voyant indicateur de message en attente…)
- Conférence téléphonique
- Serveur vocal interactif
- Visiophonie
- Rapport détaillé sur les appels

4.4.2 Approche proposée

Dans le cadre de la sécurisation de la voix dans un réseau VoIP, notre étude est basée sur la configuration suivante :

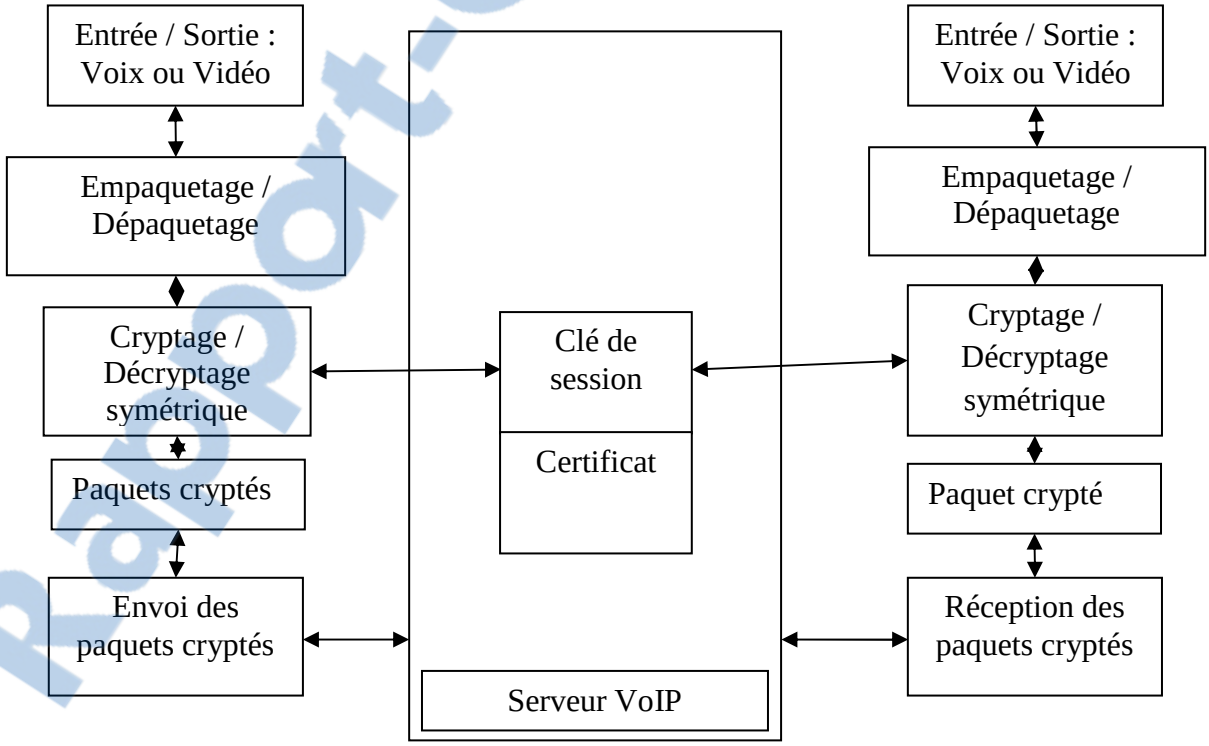


Figure 4.12: *Approche proposée : schéma de cryptage des paquets*

Comme nous l'avons décrit précédemment, VoIP est très assujettie à divers type d'attaques à savoir le capture des paquets, l'écoute clandestine des communications et bien d'autres. Notre contribution est donc de crypter / décrypter les paquets (signalisations et voix, SIP/RTP) transitant à l'entrée / sortie du réseau, comme illustrer sur la figure 4.12.

Avant toute communication, l'émetteur et le récepteur se partagent une clé de session avec le serveur. Cette clé est échangée sur le réseau, dans un paquet SIP de type « MESSAGE », à l'aide du protocole d'échange de clé de Diffie-Hellman, via un canal de transport sécurisé TLS (Transport Layer Secure) [145].

La clé de session servira à crypter et à décrypter les informations en utilisant l'algorithme de chiffrement symétrique de type « one time pad » c'est-à-dire en utilisant des clés différentes à chaque session pour chaque utilisateur, reposant sur le cryptosystème AES.

Les résultats attendus seront le comportement du serveur pour une communication normale et une communication cryptée, suivant le modèle proposé par la figure 4.13.

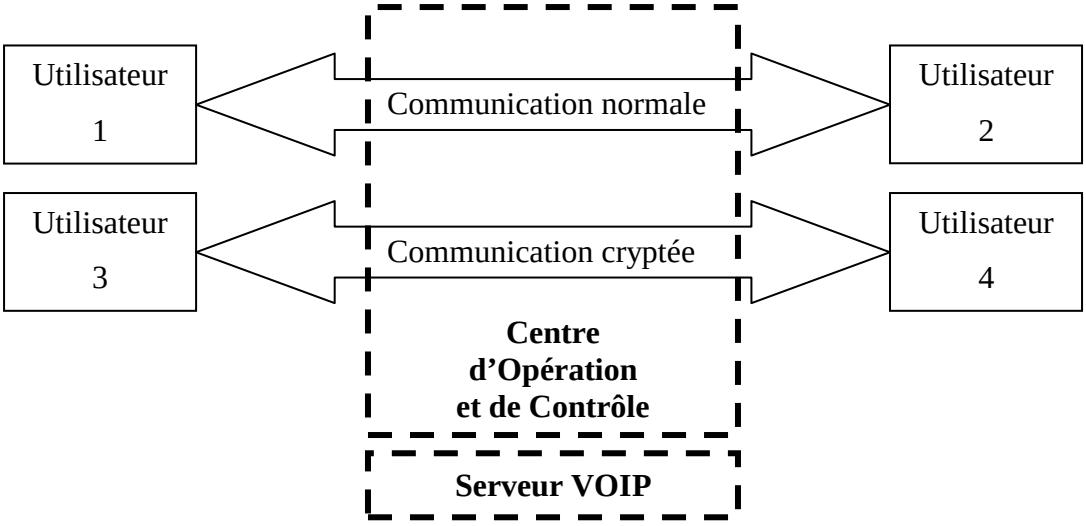


Figure 4.13: *Contrôle du trafic sur le serveur*

4.4.3 Résultats et interprétations

Dans notre recherche, nous avons utilisé cinq machines : quatre pour les clients et une pour le serveur. La configuration du serveur est comme suit : le serveur tourne sur un PC (Personnal Computer) Intel Pentium Core2Duo à 3.2 Ghz, une mémoire RAM de 1 Go et une capacité de stockage de 10 Go. Le système d'exploitation utilisé est Linux avec la

version 6 de la distribution Debian. Les logiciels utilisés sont : Asterisk 1.8 et des softphones comme X-Lite, Mizu Phone, Blink, Phonerlite, Ekiga et Twinkle. Dans toute la simulation nous avons utilisés des utilitaires d'administration de réseau et système comme Wireshark, netstat, top.

4.4.3.1 Premier cas : Communication Normale

Dans cette première simulation nous allons étudier le cas d'une communication de base ou par défaut, qui est adoptée par 80% des usagers de la VoIP. Dans ce contexte, nous verrons successivement :

- Les caractéristiques d'un paquet de signalisation SIP
- Le diagramme des échanges
- Le comportement de serveur

```
INVITE sip:1000@192.168.100.10 SIP/2.0

Via:         SIP/2.0/UDP         192.168.100.12:64312;branch=z9hG4bK-d87543-
093f6c046629653c-1--d87543-;rport

Max-Forwards: 70

Contact: <sip:1001@192.168.100.12:64312>

To: "1000"<sip:1000@192.168.100.10>

From: "1001"<sip:1001@192.168.100.10>;tag=47217725

Call-ID: 5e4d7067a23bc602MjllNzM5NmI2YjdlOGI4YWIyMGRhYWJjZGUxYzc3MzI.

CSeq: 1 INVITE

Allow:  INVITE,  ACK,  CANCEL,  OPTIONS,  BYE,  REFER,  NOTIFY,  MESSAGE,
SUBSCRIBE, INFO

Content-Type: application/sdp

c=IN IP4 192.168.100.12

t=0 0

m=audio 52970 RTP/AVP 107 119 0 98 8 3 101

a=alt:1 4 : EAan+ijN moQBg2NH 192.168.100.12 52970

a=alt:2 3 : 7RRe5NLd zmXw9Ga6 192.168.100.130 52970
```

Figure 4.14: *Extrait de paquet SIP normal de type INVITE*

Nous remarquons une description de session de base ne présentant aucun système de sécurité. La preuve en est que en utilisant l'utilitaire de capture de paquet wireshark, nous pouvons obtenir et décoder les paquets RTP, qui sont responsables du transport des voix, et ainsi nous pouvons écouter clairement les discussions.

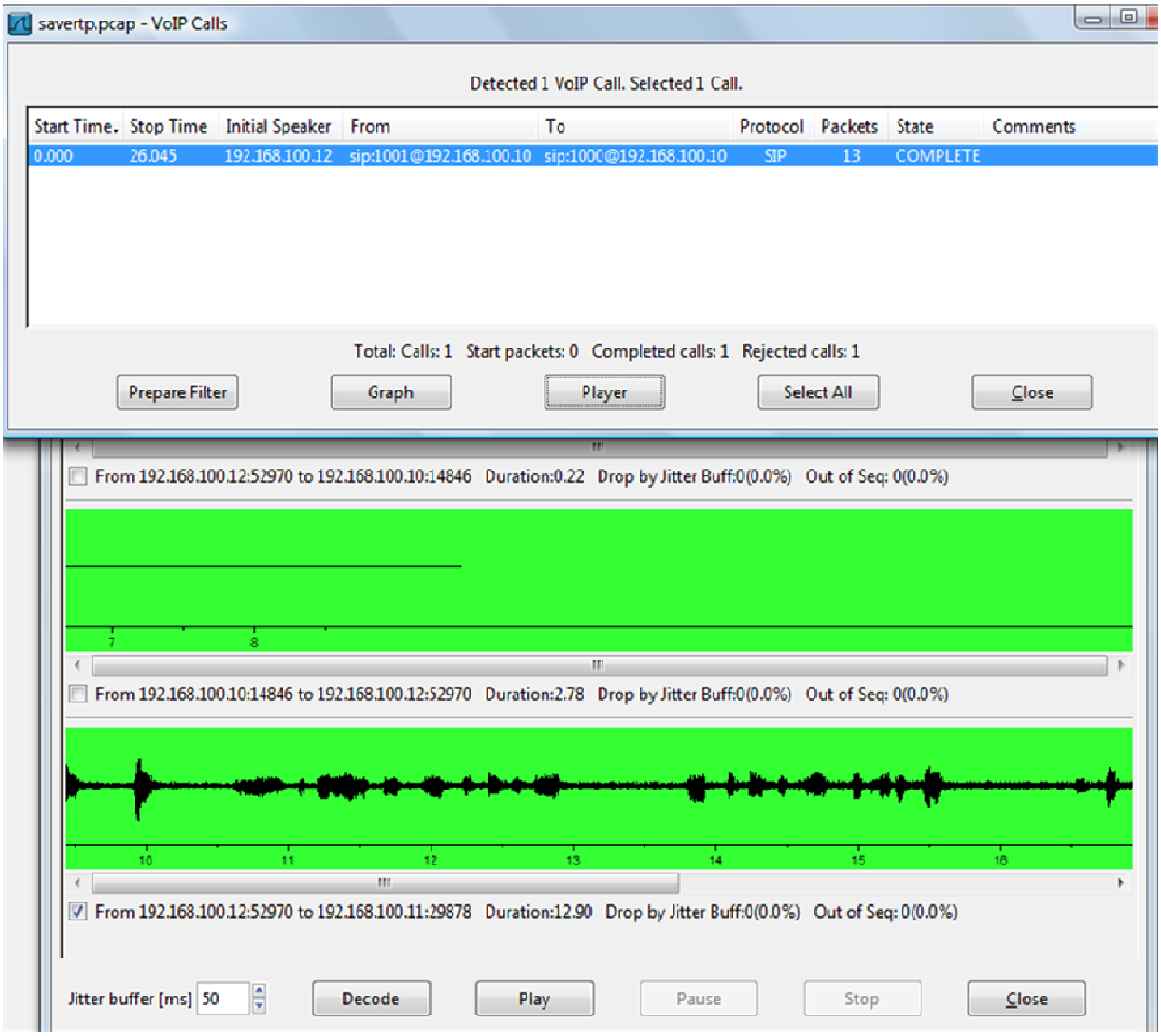


Figure 4.15: *Spectre de la voix Capturée*

La figure 4.15 montre que la voix passe dans le réseau sous forme claire. Ainsi, toute personne situant dans le réseau est capable d'écouter la communication. Ce même scénario a été comparé dans la pratique dans un cyber café en écoutant une discussion faite par un utilisateur utilisant Skype.

Le tableau 4.07 nous donne une récapitulation du comportement du serveur pendant cette communication. Il est à noter que ces valeurs représentent les pics pendants toute la communication.

CPU1	CPU2	RAM	Bande passante
6.0%	8.7%	22.2%	64Kbps

Tableau 4.07 : *Récapitulatif du comportement du serveur*

Comme nous l'avons vu dans le premier cas précédent, les infrastructures VoIP basées sur SIP/RTP ne proposent par défaut aucune confidentialité sur les flux de données voix. Ces flux peuvent être interceptés et décodés par toute personne capable de sniffer en un point du chemin pris par les paquets RTP.

L'absence par défaut de chiffrement est tout à fait justifiée. En effet, le chiffrement peut introduire un overhead (présence de gigue) au niveau des user agents. D'autre part, le chiffrement n'est nécessaire que dans certaines circonstances, pour certains utilisateurs : l'utilisation systématique du chiffrement serait inutile et rendrait plus difficile le diagnostic d'éventuels problèmes. Finalement, le chiffrement n'est pas forcément compatible avec l'ensemble des législations, certaines d'entre elles imposant même un enregistrement des conversations (cas par exemple des réglementations du gendarme américain de la bourse, la SEC, pour les appels passés depuis/vers des salles de trading). Le problème est en fait tout autre. En effet, les problèmes surviennent lorsqu'il est nécessaire de chiffrer : certaines conversations ont effectivement vocation à rester confidentielles, d'où la nécessité du chiffrement.

4.4.3.2 Deuxième cas : Communication cryptée

Le protocole SRTP (Secured RTP) a été développé pour fournir une fonction de chiffrement des flux RTP et ainsi assurer la confidentialité des communications. Ce protocole repose sur un chiffrement unique, l'AES, utilisé dans un mode le transformant en stream cipher (chiffrement par flot). L'implémentation de SRTP traite le chiffrement et le déchiffrement d'un flux SRTP à l'aide d'une même clé, dont les paramètres sont échangés par les terminaux lors de l'établissement de la communication [145].

SRTP ne définit pas de procédure d'échange des paramètres de clé. La méthode la plus répandue aujourd'hui est décrite dans le standard SDES(Security Descriptions, RFC 4568). SDES définit un attribut nommé crypto contenant les informations de clé SRTP et véhiculé dans la partie SDP d'un message SIP. On voit bien qu'il est nécessaire de sécuriser le transfert de (ou des) attribut(s) crypto, sous peine de se faire voler les paramètres de clé SRTP par un tiers. Dans notre étude l'échange des clés se fait à travers les messages SIP sur un canal sécurisé TLS qui utilise à son tour le cryptosystème RSA pour la création du certificat et la négociation des clés de session [145].

Par rapport à la communication normale, nous pouvons constater différentes descriptions. Premièrement, nous remarquons quelques changements au niveau du paquet SIP. En effet, les lignes suivantes ont été ajoutées :

```
SIP/2.0 200 OK

Via:SIP/2.0/UDP
192.168.100.130:5060;branch=z9hG4bK005b05659b6fe2118e9229e502929237;rport=5060;re
ceived=192.168.100.12

From: "2001" <sip:2001@192.168.100.10>;tag=2779306622

To: <sip:2000@192.168.100.10>;tag=80a32b129b6fe211a858c129a9a643ba

.
.
.
t=0 0

m=audio 5062 SRTP/SAVP 8 0 2 3 97 110 111 9 101

a=rtpmap:8 PCMA/8000

a=rtpmap:101 telephone-event/8000

a=fmtp:101 0-16

a=crypto:1 AES_CM_128_HMAC_SHA1_80 Inline:bu3FBm9vGSJGr6eMl4fCy8oZLcJerFn5tg5kMvA

a=ssrc:3671974514

a=sendrecv
```

Figure 4.16: *Extrait d'une description d'un paquet SIP sécurisé*

Les parties surlignées de la figure 4.16 donnent les protocoles et sécurités adoptés.

D'ailleurs, le décodage des paquets SRTP analysés montre une communication totalement cryptées. Dans ce contexte, ces spectres représentent des flux audio assimilés à un bruit.

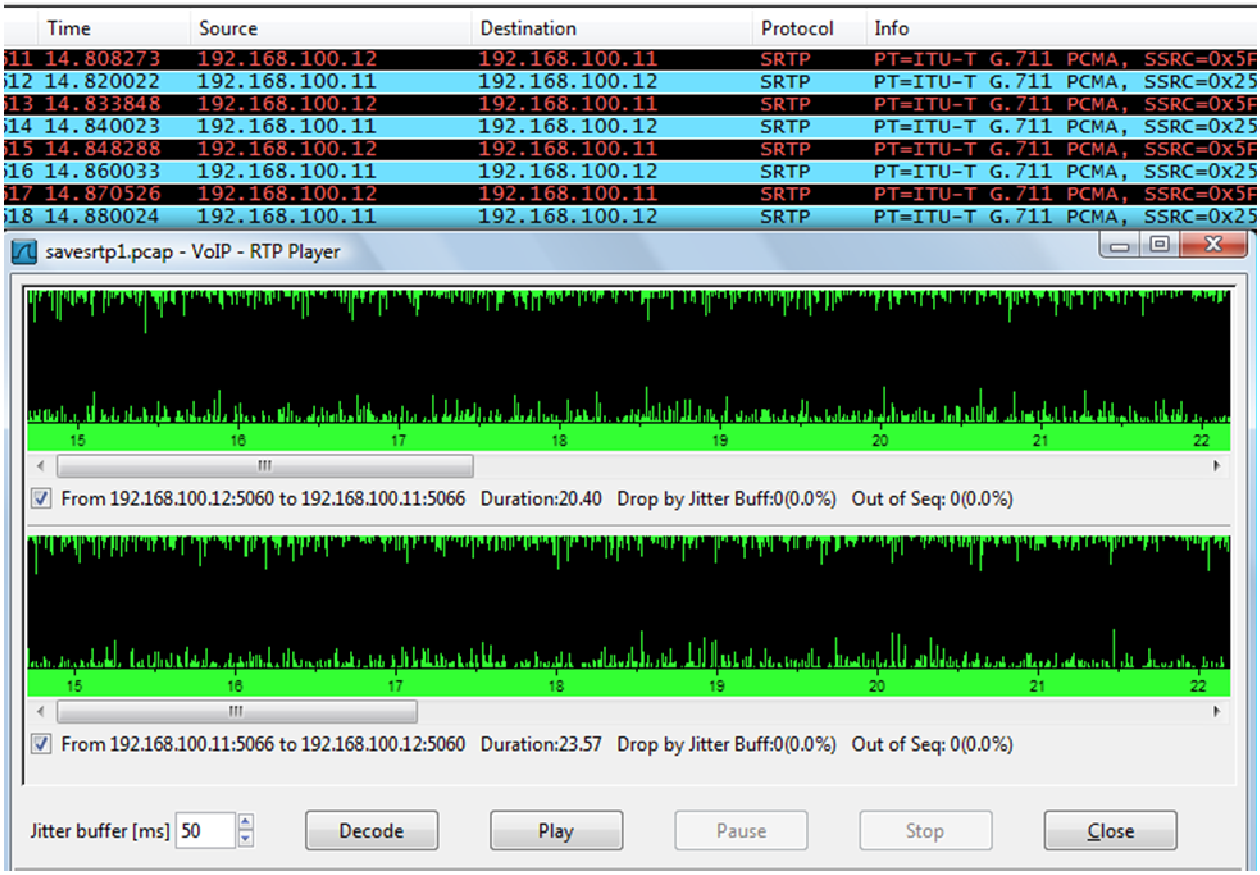


Figure 4.17: *Spectre d'une voix cryptée*

La figure 4.17 nous montre que la communication est belle et bien crypter et le protocole utilisé est SRTP.

Pour le comportement du serveur nous avons le tableau récapitulatif suivant :

CPU1	CPU2	RAM	Bande passante
6.1%	5.9%	23.8%	73Kbps

Tableau 4.08 : *Comportement du serveur pour une communication crypter*

D'après notre analyse, nous constatons une légère différence entre une communication normale et une communication cryptée. Ces différences résident sur le fait qu'une communication cryptée consomme beaucoup plus de ressource du fait que l'implémentation du module de cryptage, tant sur le transport sécurisé des clés que sur le chiffrement des paquets, dans le serveur Asterisk nécessite l'ajout d'une opération d'où la nécessité d'une ressource supplémentaire.

4.5 Conclusion

Dans ce chapitre, nous avons pu simuler les théories ainsi que les approches que nous avons proposées. En effet, nous pouvons constater que même avec le même cryptosystème, nous devrons adopter différents algorithmes suivants l'information, images ou voix, à sécuriser. Cependant, le chiffrement/déchiffrement des données engendre des opérations en plus, qui, par conséquent, demandent un supplément de ressources qu'en fonctionnement normal. Il est primordial de mettre en œuvre et d'implémenter des algorithmes performants, à la fois rapide et peu consommateur de ressources. D'après ces résultats, la modélisation ainsi que les algorithmes cryptographiques, que nous avons proposés, contribuent et apportent une nette amélioration sur la sécurisation des transferts et stockages de données, mais une optimisation est toujours souhaitée.

CONCLUSION

Au cours de cette thèse nous avons étudié trois problématiques liées au transfert et stockage sécurisé d'informations. Le premier problème est celui du transfert sécurisé d'image, le deuxième se rapportant sur l'utilisation de la méthode d'indexation et de la cryptographie pour la sécurisation dans un système d'authentification et en dernier le chiffrement des paquets transitant dans les réseaux VoIP.

Notre première contribution a porté sur la création d'une nouvelle technique qui prend en compte : la compression utilisant la Transformée de Fourier Discrète, permettant ainsi de séparer deux composants représentatifs de l'image : les coefficients représentatifs de l'image ont été cryptés à l'aide de l'algorithme de cryptage symétrique très utilisé pour sa robustesse, et d'autre part, nous avons appliqué la technique d'insertion de données cachées additives pour le partage des clés de session. Nous avons ajouté à ceux-là des algorithmes permettant d'utiliser des techniques de confusion et de diffusion.

L'utilisation de cryptage sélectif avec trois axes de recherche pour la sécurisation du transfert d'images est une stratégie très innovante et originale. Cette stratégie est prometteuse en raison de la croissance exponentielle du trafic des images sur les réseaux numériques. La phase d'analyse des résultats a permis de montrer que les images soumises à cette méthode ont un fort niveau d'intégrité vu la valeur du rapport signal sur bruit, une capacité d'insertion pour l'IDC convenable et robuste aux attaques et un cryptage sélectif satisfaisant dans un environnement à ressources matériels limitées.

Notre deuxième contribution est d'utiliser une technique innovante basée sur l'indexation d'image dans le domaine des transformées. Et c'est sur ces indexes, obtenus avec les méthodes par décomposition en valeur singulière (SVD) et par analyse en composante principale (ACP), que nous avons appliqué le cryptage asymétrique RSA, avec une longueur de clé de 2048 bits et jugé actuellement comme étant le plus robuste, comparé avec le cryptage sur un courbe elliptique ECC. L'index crypté obtenu sera donc stocké et utilisé ultérieurement par le système d'authentification pour comparaison par calcul de la distance eucludienne. D'après les résultats obtenus cette approche est très performante du fait de l'utilisation à la fois d'un système d'indexation rapide et précise et d'un algorithme cryptographique presque inviolable.

Dans la troisième contribution, notre approche consistait à sécuriser les paquets dans le réseau IP en particulier les paquets transportant la voix dans la technologie VoIP. Force est de constater que le module de chiffrement des données dans la voix sur IP n'est pas encore implémenter en totalité dans le serveur vu l'utilisation d'un service temps réel qui exige un délai de traitement minime. Les résultats ont montré qu'on peut bien sécuriser les données aux risques d'une utilisation maximum des ressources, comme le CPU et la mémoire, du serveur et une augmentation du temps de latence du système.

Bien que les approches proposées soient relativement performantes, elles ne sont pas suffisantes pour réaliser un système cryptographiquement sûr, parce que l'objectif a été de montrer l'utilisation conjointe de trois domaines. En effet, l'utilisation de diverses opérations, pour un procédé de sécurisation, dans la première approche, les algorithmes de cryptage asymétrique proposé dans la deuxième et le système de sécurisation utilisé dans la troisième approche ne rendent pas nos méthodes robustes à d'autres types d'attaques.

Perspectives :

L'approche que nous avons proposée nous amène vers les perspectives suivantes :

- *Faire une comparaison de plusieurs techniques de compression-cryptage-tatouage*
- *Elargir les recherches vers d'autres outils mathématiques*
- *Etendre les méthodes sur un flux vidéo*
- *Proposer un modèle mathématique sur la VoIP sécurisée*

ANNEXES

Annexe 1 : *article sur MADA-ETI, ISSN 2220-0673, Vol.1, pp. 1-7, 2013*

Optimisation des algorithmes de calcul cryptographique basés sur les Courbes Elliptiques

***Rakotondraina T.E.*[1], *Randimbindrainibe F.*[2]**

Laboratoire d'Informatique appliquée, Images, Signal, Télécommunication, Automatique et Mathématiques appliquées (LIISTAM)

Département Télécommunication – Ecole Supérieure Polytechnique Antananarivo

Université d'Antananarivo
BP 1500, Ankatso – Antananarivo 101 - Madagascar

[1]*tahina.ezechiel@gmail.com*, [2]*falirandimby@yahoo.fr*

Résumé

Cet article propose des algorithmes optimisés permettant de faire des opérations sur les courbes elliptiques. L'implémentation d'un programme dans un système dépend des ressources physiques et logiques qu'on va lui allouées. Vu que le cryptosystèmebasé sur les courbes elliptiquespermet d'obtenir un niveau de sécurité élevé pour une taille de clé moindre, il est primordial de construire des algorithmes de calcul aussi performant.

Mots clés : Cryptographie, courbe elliptique, algorithme, optimisation

Abstract

This article proposes optimized algorithms for making operations on the elliptic curves. The implementation of a program in a system depends on the physical and logical resources which are allocated to him. Considering the cryptosystem based on the elliptic curves makes it possible to obtain high security level with a less size of key, it is a primary importance to build such powerful calculation algorithms.

Keywords: Cryptography, elliptic curve, algorithm, optimization

1. Introduction

La cryptographie basée sur les courbes elliptiques est née en 1985, découverte indépendamment par Miller V. [7] et Koblitz N. [4].

Notre recherche s'appuie sur les ouvragesElliptic Curves Number Theory and Cryptography [10] et Guide to Elliptic Curve Cryptography [3].

Etudiées depuis des siècles, les courbes elliptiques sont parmi les objets mathématiques lesplus complexes et les plus riches qui soient. Nous verrons une définition générale des

courbeselliptiques ainsi que les opérations qui leurs sont associées, puis nous présentons des algorithmes pour effectuer des calculs sur ces courbes le plus rapidement possible et aussi sur des dispositifs à ressources limitées.

2. Définition d'une courbe elliptique

Soit K un corps, on appelle courbe elliptique sur K une courbe dans le plan projectif$\mathbb{P}^2(K)$, cubique et sans points singuliers, et munie d'un point distingué qui jouera un rôle particulier : élément neutre.Elle est donc définie par un polynôme irréductible homogène en trois variables à coefficient dans K [3]. Par un changement de variables homographique, on peut toujours se ramener à une équation dite de Weierstrass:

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3$$

Avec $a_1, a_2, a_3, a_4, a_6 \in K$.

La courbe elliptique E est l'ensemble des points $(x, y) \in$ K² satisfaisant cette équation et d'un point imaginaire $\mathcal{O}$appelé point à l'infini [10].

2.1 Loi de Groupe

Les applications des courbes elliptiques en cryptographie sont principalement dues à l'existence d'une loi de groupe que nous pouvons définir sur ces dernières. En effet, l'ensemble $E \cup \mathcal{O}$ peut être équipé avec une opération d'additionqui produit un groupe abélien dont l'élément neutre est le point à l'infini$\mathcal{O}$ [10].

En cryptologie, les courbes elliptiques sont utilisées dans le corps $\mathbb{F}_p$ avec p un nombre premier strictement supérieur à 3.
Soit E une courbe elliptique définie sur$\mathbb{F}_p$. L'équation affine de Weierstrass de E peut être simplifiée, pour$K \neq 2,3$, par :

$$y^2 = x^3 + ax + b \qquad (03)$$

La courbe définie par cette formule admet un unique point à l'infini (i.e. avec $z = 0$), de coordonnées (0 :1 :0). C'est en général ce point qui sera distingué. On appelle discriminant de cette courbe l'élément $-16(4a^3 + 27b^2)$ de K.

Le facteur entre parenthèse est le discriminant du polynôme membre de droite :

$$4a^3 + 27b^2 \not\equiv 0(modp) \qquad (04)$$

Soient $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ deux points sur E :

- Le point $(x_1, -y_1)$ est l'opposé du point P et il est noté –P.
- Si $Q \neq P$ et$Q \neq -P$, alors le point $R = P + Q = (x_3, y_3)$ est défini par

$$\begin{cases} x_3 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2 \\ y_3 = y_1 + \left(\frac{y_1 - y_2}{x_1 - x_2}\right)(x_3 - x_1) \end{cases} \qquad (05)$$

- Si $P = Q$, alors le point $2P = (x_3, y_3)$ est défini par :

$$\begin{cases} x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \\ y_3 = x_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_3 - x_1) \end{cases} \quad (06)$$

- Si $x_1 = x_2$ mais $y_1 \neq y_2$, alors $R = \mathcal{O}$
- Si $P = Q$ et $y_1 = 0$, alors $R = \mathcal{O}$

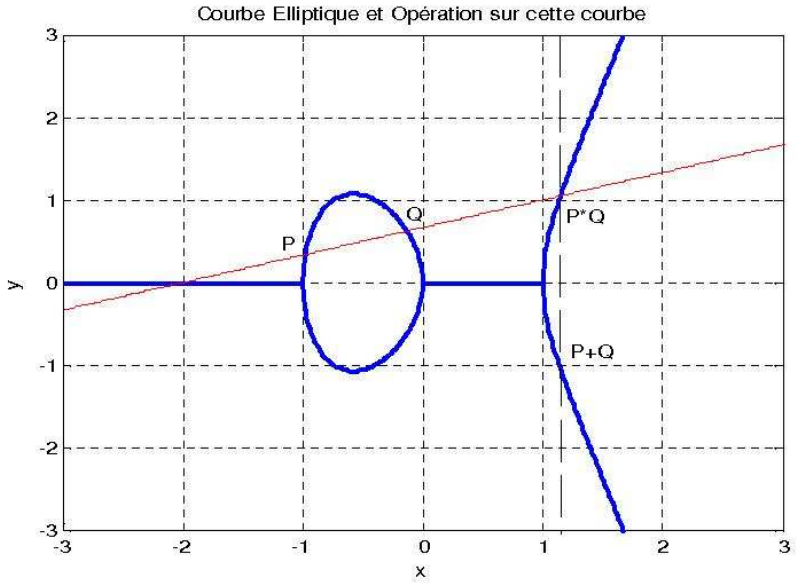


Figure 01 *: Courbe elliptique d'équation* $y^2 = 3x^3 - 3x$

2.2 Remarque

Notons $\mathcal{O}$le point distingué de E. On montre qu'il existe une loi de groupe sur les points de E telle que $\mathcal{O}$soit l'élément neutre et que l'identité $P + Q + R = \mathcal{O}$soit vérifiée pour tout triplet de point alignés. Lorsque la convention $\mathcal{O} = (0:1:0)$ est en vigeur, l'opposé de chaque point de E est son symétrique par rapport à l'axe des abscisses.

La somme de deux points P, Q finis et non opposés est donc le point de coordonnées $(x_3, -y_3) o\grave{u} x_3, y_3$sont donnés par la relation (04). Bien sur, lorsque l'un des points est à l'infini, ou lorsqu' ils sont symétriques l'un de l'autre, on a les identités $P + \mathcal{O} = P$ et $P + (-P) = \mathcal{O}$.

3. Algorithmes de calcul sur les courbes elliptiques

Les algorithmes suivants permettent de calculer le doublement et l'addition.

Pour ce premier algorithme nous allons procéder ainsi :

- Parcourir chaque bit de la clé.
- A chaque étape, doubler une variable qui sert de résultat,
- Initialiser à l'élément neutre,
- Si le bit en cours est à 1, alors ajouter le point P à ce résultat et continuer.
- Retourner la valeur de la variable.

Algorithme 1 :

$Nom : doubleAndAdd$

$Entrée : P, d = (d_{l-1}, \ldots, d_0)_2$

$Sortie : Q = [d]P$

$début$

$T \leftarrow \mathcal{O}$

$Pour\ i \leftarrow l-1 \text{ à } 0\ faire$

$T \leftarrow [2]T$ // Doublement

$si\ d_i = 1\ alors$

$T \leftarrow T + P$ // Addition

$finsi$

$finpour$

$retourner\ T$

fin

L'algorithme 2 ci-après est une modification de l'algorithme *doubleAndAdd* qui permet une plus grande résistance face aux attaques simples par analyse de canaux cachés. Le but de cet algorithme est d'exécuter un doublement et une addition à chaque pas de l'algorithme, et ce, quelque soit la valeur du bit de clé traitée.

Algorithme 2 :

$Nom : doubleAndAddAlw$

$Entrée : P, d = (d_{l-1}, \ldots, d_0)_2$

$Sortie : Q = [d]P$

$Début$

$T_0 \leftarrow \mathcal{O}$

$T_1 \leftarrow \mathcal{O}$

$Pour\ i \leftarrow l-1 \text{ à } 0\ faire$

$T_0 \leftarrow [2]T_0$ // Doublement

$T_1 \leftarrow T_0 + P$ // Addition

$si\ d_i = 1\ alors$

$T_0 \leftarrow T_1$ // Calcul de T_1

$sinon$

$T_0 \leftarrow T_0$

$finsi$

$finpour$

$retourner\ T_0$

fin

Les deux algorithmes précédents sont les plus simples que l'on peut trouver pour calculer $[d]P$ sans effectuer d'additions. Cependant, on voit que le nombre d'additions dépend du nombre de bits à 1 dans la représentation binaire de la clé. Il semble alors naturel de vouloir éclaircir cette représentation, c'est-à-dire de trouver un moyen pour avoir le moins de 1 possible. Pour cela, on va utiliser un algorithme qui ne va plus prendre en entrée la représentation binaire de k, mais une représentation dite *NAF*, pour Non-Adjacent Form. Dans cette représentation, il n'y a pas de chiffres consécutifs différents de 0, mais on introduit des chiffres à -1. Il nous faut les deux algorithmes suivant: un pour calculer cette nouvelle représentation; un autre pour effectuer la multiplication scalaire.

Le premier retourne la représentation du scalaire en base 2 où il y a un écart d'au moins un bit entre deux chiffres non nuls. Une fois cette représentation calculée, on peut effectuer la multiplication scalaire grâce au deuxième algorithme.

Algorithme 3

$Nom : naf$

$Entrée : k \in \mathbb{N}$

$Sortie : NAF\ (k)$

$début$

$i \leftarrow 0$

$tant\ que\ k \geq 1\ faire$

$si\ k\ est\ impaire\ alors$

$k_i \leftarrow 2 - (k\ mod\ 4)$

$k \leftarrow k - k_i$

$sinon$

$k_i = 0$

$k \leftarrow k/2$

$i \leftarrow i + 1$

$finsi$

$retourner\ (k_{i-1}, \ldots, k_1, k_0)$

$fintantque$

fin

Et pour le calcul de kP, nous avons :

Algorithme 4

$Nom : mulNaf$

$Entrée : P, \quad k \in \mathbb{N}$

$Sortie : Q = [k]P$

$début$

$Q \leftarrow \infty$

$pour\ i\ de\ t - 1\ à\ 0\ faire$

$Q \leftarrow 2Q$

$si\ k_i = 1\ alors$

$Q \leftarrow Q + P$

$sinon\ si\ k_i = -1\ alors$

$Q \leftarrow Q - P$

$finsi$

$retourner\ Q$

$finpour$

fin

Nous avons aussi utilisé un algorithme beaucoup plus rapide appelé l'exponentiation de Montgomery. Pour calculer kP, l'exponentiation de Montgomery part du couple $(P, 2P)$ et donne en sortie le couple $(kP, (k + 1)P)$.

Voici son algorithme :

Algorithme 5

$Nom : montgomery$

$Entrée : P, \quad k = (1, k_{n-2}, \ldots, k_0)_2$

$Sortie : Q = [k]P$

$Début$

$P_1 \leftarrow P$

$P_2 \leftarrow 2P$

$Pour\ i \leftarrow n - 2\ à\ 0\ faire$

$P_{k_1 - 1} \leftarrow P_1 + P_2$ // Addition

$P_{k_1} \leftarrow 2P_{k_1}$ // Doublement

$finpour$

$retourner\ P_1$

fin

4. Résultats et interprétations

Pour vérifier la performance des algorithmes, nous avons fait quelques tests par chiffrement d'un fichier texte de 1Ko et nous avons comparé le résultat par rapport à l'algorithme de chiffrement asymétrique RSA, comparé avec le même niveau de sécurité.

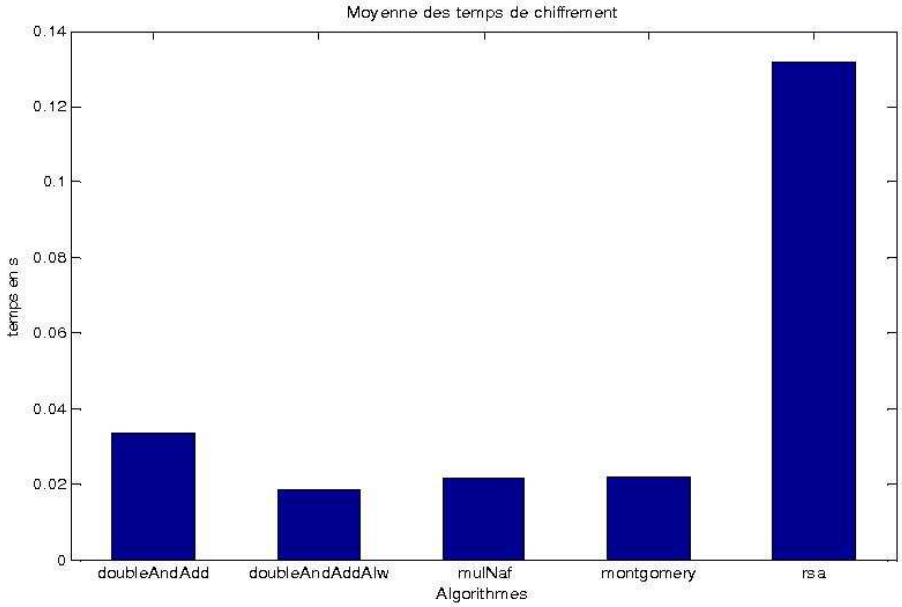


Figure 02 : **Estimation du temps de chiffrement des cinq algorithmes**

La figure 02 montre une grande différence sur ces cryptosystèmes. Les algorithmes utilisés pour le chiffrement sur une courbe elliptique sont très optimisés de l'ordre de quelques millisecondes. Comme les opérations sur les courbes elliptiques ne sont que des successions d'addition et de doublement, ils ne consomment pas et ne nécessitent pas tant de ressources pour effectuer les calculs.

5. Conclusion

Dans notre recherche, nous avons montré que même si les cryptosystèmes basés sur les courbes elliptiques présentent un haut niveau de sécurité, pour une longueur de clé minime, une bonne optimisation des algorithmes les rend encore plus performants. Cependant, nous constatons une facilité d'implémentation de ce programme dans des systèmes présentant des ressources limitées tels que les systèmes embarqués, téléphone et appareils photo.

6. Références

[1] W.Diffie, M.E.Hellman, « *New directions in cryptography* », IEEE Trans. Inform. Theory, vol. 22, n°6, pp. 644-654, 1976.

[2] T.ElGamal, « *A public key cryptosystem and a signature scheme based on discrete logarithms*», IEEE Transactions on Information Theory, vol.31, pp.473-481, 1985.

[3] D. Hankerson, A.Menezes, Vanstone S, « *Guide to Elliptic Curve Cryptography* », Springer, 2004.

[4] N.Koblitz, « *Elliptic curve crypto-systems*», Math. of Comp., vol. 48 n°177, pp. 203-209, 1987.

[5] A.J.Menezes, P.C.Oorschot, S.A.Vanstone, « *Handbook of Applied Cryptography* », CRC Press, 1996.

[6] A.J.Menezes, S.A.Vanstone, « *Elliptic curve cryptosystems and their implementation* ». Journal ofCryptology, vol. 6, pp. 209-224, 1993.

[7] V.S.Miller, « *Use of elliptic curves in cryptography*»,Advances in Cryptology CRYPTO'85, Lecture Notes in Computer Science, pp. 417-426.

[8] R.L.Rivest, A.Shamir, L.Adleman, « *A method for obtaining digital signature and public key cryptosystems* ». Comm. ACM, vol. 21, pp.120-126, Feb 1978.

[9] J.Walter, «*The role of ECDSA in wireless communication (implement-tation and evaluation of ECDSA on constrained devices)* », Los Angeles, 2002.

[10] L.C.Washington, « *Elliptic Curves Number Theory and Cryptography* », Chapman & Hall/CRC, 2003.

Annexe 2 : *article sur IEEE Xplore ISBN: 978-1-4673-1520-3, pp : 48-50, 2012*

Mathematical Modeling of SIP Call

Ravonimanantsoa N.M., Randriamitantsoa A.A, Rakotondraina T.E

Abstract— **The use of VoIP is becoming more common in the world of the Internet**

Modeling of voice over IP at the server is an interesting work for future uses of this technology. The reason for this mathematical modeling is to forecast or predict the levels of bandwidth allocated for a number of user. While using different parameters assumed to be used according to their use in the server ie CPU, NIC, memory. The allocation of new parameters, such as the protocol and are used logicIles recently studied. In this paper, we propose a new mathematical model polynomial models to predict the use of the network interfaces using the SIP protocol. Our method aims to determine the number of clients possible for a mini VoIP server. It has several advantages such as the definition of a minimum required for the deployment of a VoIP family and also the limited number of clients using the server capacity. For our experiment we conducted a simulation of SIP call and deduce the mathematical model by interpolation.

Voip,Sip,Asterisk,polynom

I. INTRODUCTION

This template, the use of the Internet is widely seen as a tool in our daily life, different research has been conducted to improve the user satisfaction on this technology in full bloom.

However, in recent research, it is reported that there is a kind phenomenon of ourselves to the "all IP". This phenomenon marks the beginning of a new era in the field of telecommunications, having crossed for centuries in the resale switched telephony, we are in the era of Telephony IP. This new deal does not mean that the switched telephone networks are more efficient but the reason lies in the idea of cost for infrastructure.

The convergence of traditional telephony to IP telephony is an area of research that deserves many attentions.

Much research has been done on this subject since its appearance [1]. Here we focus on SIP will be the future of IP telephony [2] for its performance.

To conduct our research we have done experiments and simulation.

In this paper, we propose an approach based on experimentation to build a polynomial of degree three (3) for the behavior of the SIP call over a network. We built an experiment with a VoIP server Asterisk running on a Debian 6.0 operating system and two X-lite softphone client that is installed on a Windows XP operating systems to collect data simulation. Using these data, we construct a polynomial model describing the evolution of the amount of captured package. The behavior of each instance during the call is also analyzed.

The motivation of this paper is organized as follows in Section 2. The experimental, results of experiments and observations are described in Section 3. Section 4 analyzes the behavior of our curve. A polynomial model is constructed in Section 5.and finally the conclusion and future work of our research..

II. MOTIVATION

Speech Based Dialog Query System over Asterisk PBX Server[3]

Research entitled "Speech Dialog System based query over Asterisk PBX server" gives insight into the theoretical details and implementation of a speech-based query system for dialogue on the Asterisk server. Asterisk PBX server can be connected to traditional analog phones or PSTN trunk lines (Public Switched Telephone Network) for connection with the PCI Special equipment and VOIP phones using different VoIP protocols such as SIP, IAX and H.323 . This system is designed for queries Railway Enquiry in Indian (Hindi) and English. this document describes how a query system can be deployed Asterisk server to two analog phones and softphones as client interfaces and the results are analyzed recognition accuracy.

From this paper, we have tested the idea of the reality of a call using SIP.

III. THE EXPERIENCE

In this section we will talk briefly about the hardware and software used in our experiment, then in a second part we will talk about the experience itself.

The architecture

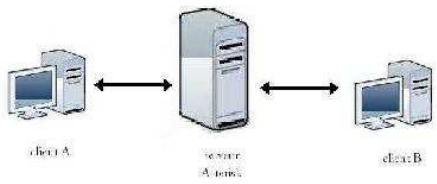


Figure 1. Architecture

Our network architecture is minimized by a server connected to two client machines to make the call (minimum). All this is connected is connected by a switch

Experience

- First, we begin to initiate the call from SIP softphone client windows. Next, we move to a packet capture on our Asterisk server via the command "tcpdump"
- Next, we arrange the packets captured by the command "tcpstat" so it can be traced by our gnuplot graph plotter.
- Finally, we observe the behavior of our graph for a definite time before the call is hung up.

Result

The results give us the following figures

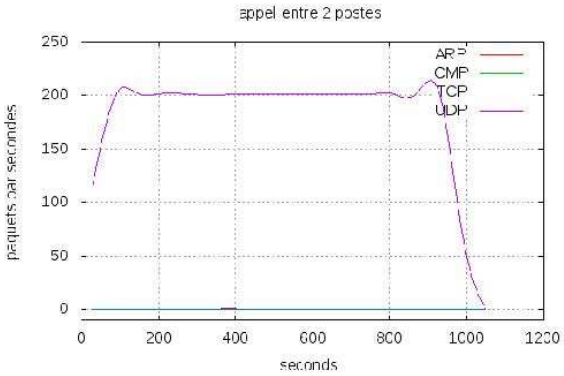


Figure 2. Graphe of SIP call

```
#!/bin/bash
echo ce script permet de parser le flux
tcpdump pour adapter a gnuplot
tcpstat -r traffic_2.dmp -o "%R\t%A\n"
60 > arp_2.data
tcpstat -r traffic_2.dmp -o "%R\t%C\n"
60 > icmp_2.data
tcpstat -r traffic_2.dmp -o "%R\t%T\n"
60 > tcp_2.data
tcpstat -r traffic_2.dmp -o "%R\t%U\n"
60 > udp_2.data
gnuplot gnuplot.script > graphe_2
```

the shell script tcpstat enabled us to plot the graph is below

Figure 3. tcpstat shell

```
Via: SIP/

14:53:15.036004 IP debian.local.sip >
192.168.1.113.21460: SIP, length: 500

E....C..@.-i...o...q..S....>SIP/2.0 100 Trying

Via: SIP/2.0/UDP 192.168.1.1

14:53:15.036535 IP debian.local.sip >
192.168.1.222.44094: SIP, length: 944

E...{...@.wU...o.....>...gINVITE
sip:2222@192.168.1.222:44094;rinstance=0c

14:53:15.036616 IP debian.local.sip >
192.168.1.113.21460: SIP, length: 516

E...D..@.-X...o...q..S....NSIP/2.0 180 Ringing

Via: SIP/2.0/UDP 192.168.1.
```

Figure 4. Tcpdump extract

IV. Analysis of this curve

Based on our findings, we noticed that during the initiation of the call or SIP opens the session with "Guest" we observe a peak in the number of packets per second, this peak is shown at the closing the session can be found or the message "BYE".

So we can conclude that:

- Sends the message invites a lot of packet from the conversation
- Similarly for "bye"
- Finally, our simulation shows that an unanswered call may cause overload our network interface by this peak

V. Polynomial Model

After we obtained our graph was exported in the Matlab software has come for us to deduce a polynomial model with our graph using the function of matlab polyfit. We have chosen three (3) as the degree of our polynomial by the fact that we define three parameters for our research includes: memory, processor and speed of our networks. Thus we have obtained the following result

```
>> plot (x,y)
>> p=polyfit (x,y,3)

p =

  1.0e+004 *

  -0.0000   0.0000  -0.0005   1.0358
```

Figure 5. Matlab script and result

$$P(x)= 1.0e+004(0x^3+0x^2-0.0005x+1.0358) \quad (1)$$

the polynomial will be of this form

$$P(x)= Ax^3+Bx^2+Cx+D) \quad (2)$$

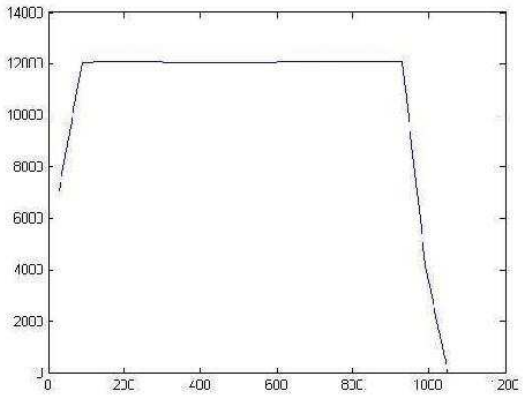


Figure 6. Matlab curv

With

A=parameter of memory

B=parameter of processor

C=Parameter of network

And D the constant of our sip status

VI. CONCLUSION AND FUTURE WORK

In this paper it is shown that the introduction to the call for a SIP is a peak which is not significant but may cause a malfunction of the interface resale after several attempts if n 'is not picked up at the Asterisk server.

Experimental results show that the communication itself represents a number of this package increases with number of calls. Our simulation and our experience leads us to conclude that establish a SIP session is a certain level of packet-level network interface and a number of session afraid to challenge the level of our communication network interface. Finally, we conclude that we can model our call by determining the constants A, B, C and D to see what type of computer for any number of client for the Asterisk server. We anticipate that the design of such a model requires the simulation with several poste. Diverses methods can lead us in this endeavor to improve the efficiency of our model..

REFERENCES

[1] Saurabh Goel,Muhua Bhattacharya,speech based dialog query system over Asterisk PBX Server,ICSPS 2010

[2] H. Abdelnur, R. State, I. Chrisment, and C. Popi, "Assessing the security of VoIP Services," in Integrated Network anagement, IM 2007. 10th IFIP/IEEE International Symposium on Integrated Network Management, Munich, Germany, 21-25. IEEE, May 2007, pp. 373–382.

[3] I E. Chen, "Detecting dos attacks on sip systems," VoIP Management and Security, 2006. 1st IEEE Workshop on, pp. 53–58, April 2006..

[4] D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis,S. Gritzalis, K. Ehlert, and D. Sisalem, "Survey of security vulnerabilities in session initiation protocol," Communications Surveys & Tutorials, IEEE, vol. 8, no. 3, pp. 68–81, Qtr. 2006..

[5] A. Habib, M. M. Hefeeda, and B. K. Bhargava, "Detecting service violations and dos attacks," in In Proceedings of 2003 Internet Society Symposium on Network and Distributed System Security (NDSS03, 2003, pp. 177–189.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[6] P. montoro, E. casilari, A comparative stutdy of VoIP standarts with asterisk,Fourth International conference on Digital Télécommunications 2009

[7] Ravonimanantsoa N M.V,Randriamitantsoa P.A "Comparison Of The Consumption Of Resources Between HTTP And SIP" Advanced Engineering Forum Volume 1 p330

Ravonimanantsoa N.Manda-vy

Student at University of Antananarivo and have thesis in computer application in telecommunication, born may 23, 1978 in Antsirabe (Madagascar),have a diploma computer engineer in 2008 at University of Fianarantsoa(Madagascar)and mathematic degree in 2002 at University of Antananarivo(Madagascar.

He has two paper published in international journal and teach at Ecole Supérieur Polytechnique d'Antananarivo

Annexe 3 : *article sur MADA-ETI, ISSN 2220-0673, Vol.1, pp.8-12, 2013*

Transfert sécurisé d'images dans le domaine de la TFD

***Rakotondraina T. E.*[1], *Ramafiarisona H. M.*[2]**

Laboratoire d'Informatique appliquée, Images, Signal, Télécommunication, Automatique et Mathématiques appliquées (LIISTAM)

Département Télécommunication – Ecole Supérieure Polytechnique Antananarivo

Université d'Antananarivo

BP 1500, Ankatso – Antananarivo 101 - Madagascar

[1]*tahina.ezechiel@gmail.com*, *2mhramafiarisona@yahoo.fr*

Résumé

Cet article présente une nouvelle approche pour un transfert sécurisé d'images. Elle comporte trois traitements à savoir : une compression basée sur la Transformée de Fourier Discrète ou TFD, une opération de cryptage symétrique et une technique d'Insertion de Données Cachées ou IDC pour le transport des informations sensibles.

1. Introduction

Notre recherche est basée sur la combinaison de trois méthodes de traitement d'information.

Premièrement, l'information à transférer, c'est-à-dire l'image, subit un codage source qui est une compression du signal utilisée dans le but d'éliminer toutes redondances et d'optimiser la puissance de calcul. Pour cela nous avons utilisé une méthode par transformée qui est la Transformée de Fourier sur une information discrète (TFD). Le choix de ce mode est que, d'une part, la TFD représente les coefficients de l'image sous forme complexe, ce qui multiplie le choix de l'utilisation de ces coefficients et d'autre part, elle simplifie la représentation matricielle de l'image et diminue ainsi le nombre de calcul et les manipulations à faire.

Deuxièmement, nous avons mis au point un algorithme de génération de clé aléatoire qui serait en mesure de donner des clés de session servant à crypter l'information. L'algorithme utilisé est l'algorithme de cryptage symétrique AES, connu, utilisé et implémenté dans divers systèmes informatiques, du fait de sa rapidité et sa robustesse face aux différents types d'attaques connues. Le système de cryptage opère seulement sur une partie de l'information, c'est ce qu'on appelle cryptage sélectif. Nous avons donc choisit une partie de

la représentation de l'image, obtenu après utilisation de la transformée de Fourier, qui représente les coefficients représentatifs de l'information, c'est-à-dire la partie réelle de la transformée.

Troisièmement, pour le transport des clés de session, qui vont être utilisées pour le déchiffrement, on insère celles-ci dans l'autre partie des coefficients qui est la partie imaginaire. Pour cela nous avons utilisé une technique de tatouage additif pour sa résistance envers les types d'attaques comme l'ajout de bruit et le filtrage.

Nous présentons deux techniques qui diffèrent l'une de l'autre par leur robustesse face aux attaques.

2. Approche proposée

A l'émission, nous avons le schéma des opérations suivantes :

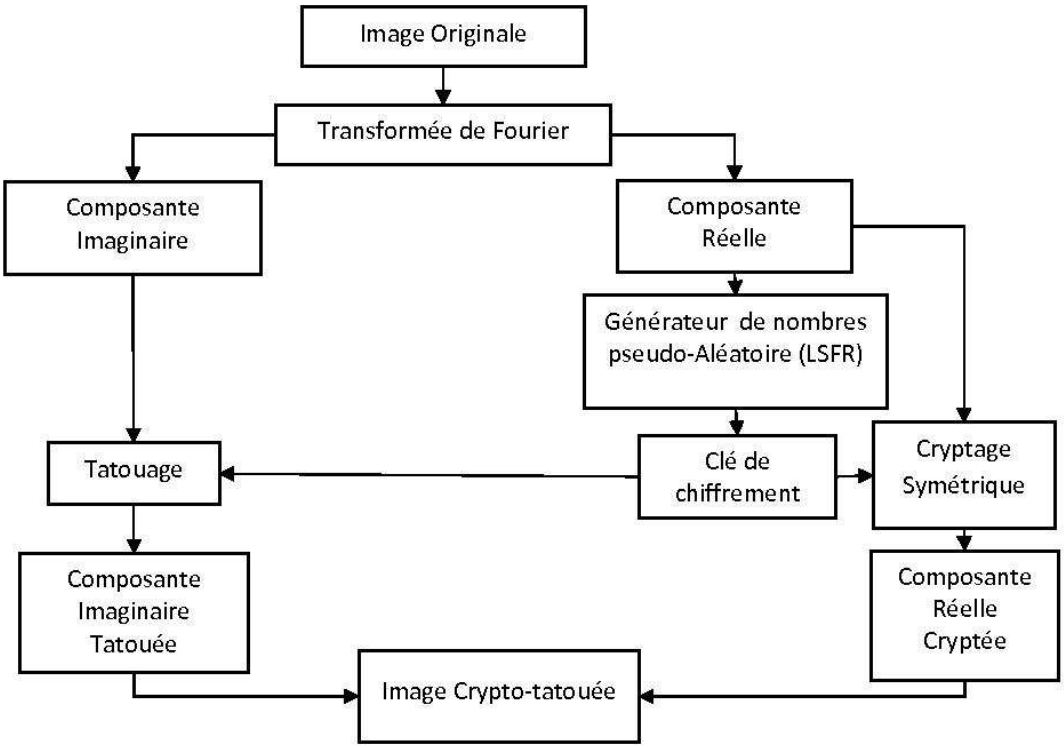


Figure 01 : *schéma à l'émission de l'approche proposée*

A la réception, la restitution de l'information se fait par une série d'opération inverse à celle proposée.

4.1.1 *Résultats et interprétation*

Comme le chiffrement utilisé une méthode symétrique, la clé de chiffrement sera encore utilisée pour le déchiffrement au niveau du récepteur. Pour permettre un partage sécurisé de cette clé de session nous avons dissimulé celle-ci dans une autre partie de l'information à envoyer. Pour cela nous avons utilisé une technique de tatouage d'information. Le choix du type d'insertion de données est basé sur les tatouages robustes face aux compressions et transformation géométrique comme la rotation et la translation.

Pour cela nous avons opté sur l'utilisation de la méthode de tatouage additif.

Il est à noter que dans le programme, nous avons ajouté différentes techniques de diffusion et confusion pour rendre l'algorithme difficile à appréhender par un cryptanalyste, mais rapide en même temps. Les séries de test sont faites sur une machine Intel Pentium Dual Core de 2,2 Ghz avec 3072 Mo de mémoire RAM.

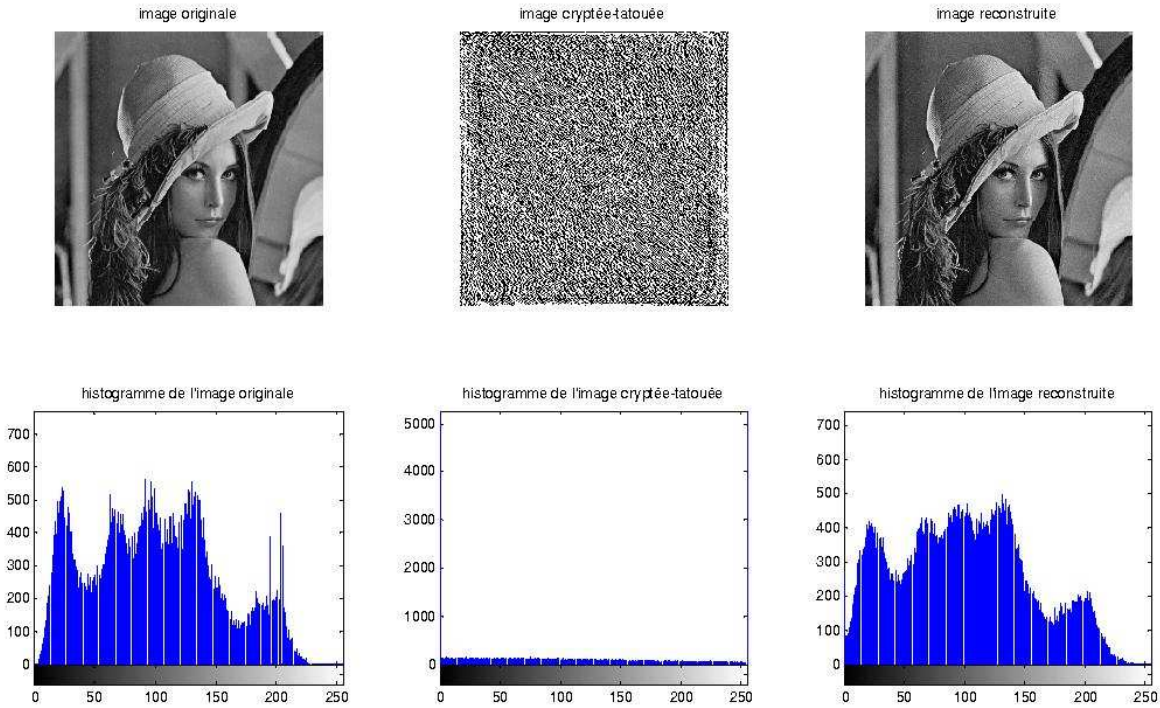


Figure 02 : *Colonne 1 : image originale et son histogramme, colonne 2 : image cryptée, tatouée et son histogramme, colonne 3 : image reconstruite, décryptée et son histogramme*

On constate d'après la figure que l'opération engendre une perte d'information équivalente à un pic du rapport signal sur bruit ou $PSNR = 31.3674\ dB$, une erreur quadratique moyenne

$MSE = 47.4619$ et une maximum de déviation quadratique $maxerr = 38.9445$. La méthode utilisée dans cette partie a été appliquée dans la partie LSB de chaque pixel, c'est pour cela que cette méthode est plutôt robuste car nous arrivons encore à extraire les clés de session après une attaque par filtrage médian et après ajout de bruits de type « salt and pepper ». Cette approche ne résiste pas face aux attaques géométriques.

La corrélation entre l'image originale est celle reconstituée est de $corr = 0.9933$, ce qui correspond à un résultat acceptable selon son importance. Le tableau 01 montre l'efficacité du programme tant à l'émission qu'à la réception. L'implémentation d'un tel programme est donc optimisée pour une utilisation sur une plateforme à ressources limitées comme les systèmes embarqués ou les appareils photos.

	Temps d'exécution	Temps processeur utilisé
Emission	0.4315 s	0.4212 s
Réception	0.0829 s	0.0936 s

Tableau 01 : *Temps d'exécution des programmes à l'émission et à la réception*

Dans une deuxième approche, représentée par la figure 03, l'opération n'est plus faite sur les bits les moins significatifs mais plutôt sur l'ensemble des bits. Par conséquent, nous somme confronté à beaucoup de perte d'informations à la réception du faite que nous utilisons des faibles coefficients de l'image obtenus après la Transformée de Fourier.

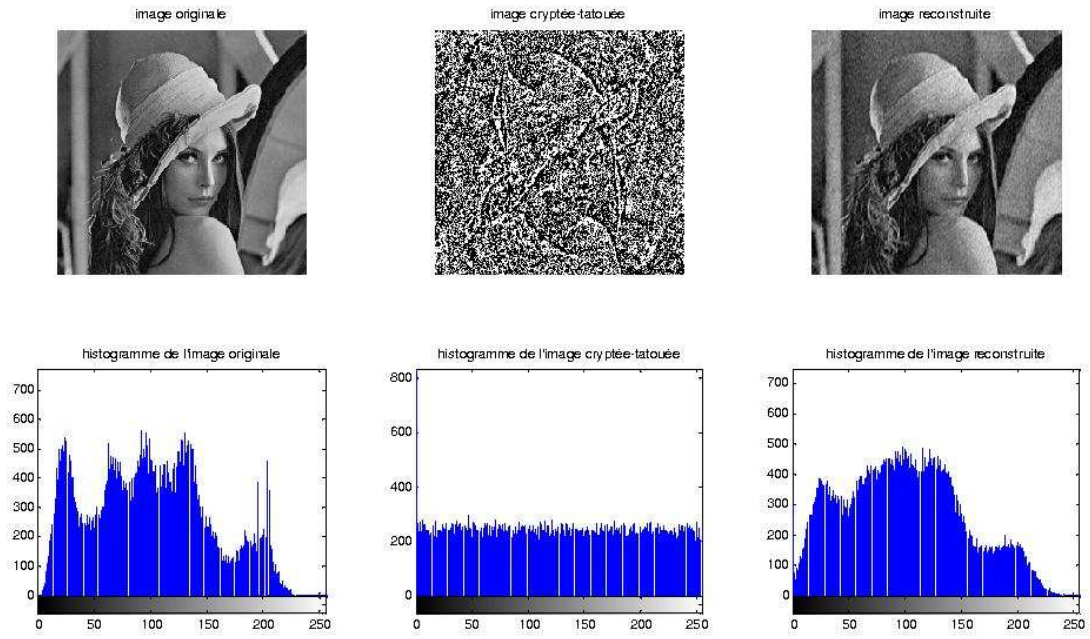


Figure 03 : *Colonne 1 : image originale et son histogramme, colonne 2 : image cryptée, tatouée et son histogramme, colonne 3 : image reconstruite, décryptée et son histogramme*

Nous avons les résultats suivants :

PSNR (dB)	MSE	Maxerr	Corr
7.1407	1.2561 e+4	1.3641e+3	-0.0028

Tableau 02 : *Résultats obtenus*

Il est à noter que cette deuxième méthode n'est pas du tout résistant face aux attaques par bruitage, filtrage et transformation géométrique. Après chaque attaque nous n'arrivons pas à avoir les informations tatouées.

	Temps d'exécution	Temps processeur utilisé
Emission	0.6518 s	0.5772 s
Réception	0.1000 s	0.0936 s

Tableau 03 : *Temps d'exécution des programmes à l'émission et à la réception*

Ce deuxième cas diffère juste du premier sur la manière de représenter les coefficients de la partie réelle de l'image par rapport aux clés de chiffrements, c'est-à-dire en choisissant d'opérer sur les bits le moins significatifs.

3. Conclusion

Notre approche se base sur l'utilisation de la représentation des coefficients d'image sous forme complexe suite à une transformation par TFD. On a pu voir qu'il est avantageux d'utiliser un traitement sélectif de l'image surtout pour la manipulation et la représentation de la matrice. On optimise ainsi la rapidité en temps de traitement et permettre une parallélisation de l'opération de cryptage et de tatouage. Cette approche est bien adaptée sur des environnements à faibles ressources matérielles et d'espace mémoire. Pour la robustesse, la combinaison du cryptage AES et tatouage additif est plus qu'avantageux. AES est souvent recommandé pour un cryptage symétrique et le tatouage additif très utilisé, tous les deux sont réputés pour leur résistance face aux types d'attaques très connues et courantes.

4. Référence bibliographique

[1] P. Duhamel and M. Vetterli, « *Fast Fourier transforms: a tutorial review and a state of the art* » Signal Processing, pp. 259–299, 1990.

[2] J. D. Gibson, *"Handbook of Image and Video processing"*, Academic Press, 2000

[3] Denning, E. Dorothy, *"Cryptography and data security"*, Addison Wesley, Purdue University, 1982.

[4] H. B. Razafindradina, P. A. Randriamitantsoa, « *Tatouage robuste et aveugle dans le domaine des valeurs singulières* », Laboratoire LASM, pp. 5-7, janvier 2010.

[5] J. Marconi, M. Rodrigues, « *Transfert sécurisé d'images par combinaison de techniques de compression, cryptage et marquage* », Thèse Doctorat, Université Montpellier II, 2006.

[6] W. Bender, D. Gruhl, N. Morimoto, *"Techniques for data hiding"*, IBM Systems Journal, pp.133-336, 1996.

Annexe 4 : *article sur International Journal of Computer Science and Network (IJCSN), ISSN 2277-5420, vol.2, 2013*

Authentication System Securing Index of Image using SVD and ECC

[1] Tahina Ezéchiel Rakotondraina, [2] Paul Auguste Randriamitantsoa, [3]Dr. Henri Bruno Razafindradina

1&2 Department of Telecommunication, High School Polytechnic of Antananarivo, University of Antananarivo
Antananarivo, Ankatso BP 1500, Madagascar
tahina.ezechiel@gmail.com, rpauguste@gmail.com

2 Department of Telecommunication, High Institute of Technology
Diego Suarez, Madagascar
hbrazafindradina@gmail.com

Abstract

This paper presents a new approach to securing information stored in a database. It has three components including: an operation for indexing images using Singular Value Decomposition (SVD), which will constitute the reference images, asymmetric encryption operation using Elliptic Curve Cryptosystem (ECC), aiming to make confidential these reference images stored and a technique for comparing these images to a query image using the Euclidian Distance.

Keywords: *Image Indexation, Cryptography, Euclidian Distance, Secure Database, Fingerprinting*

1. Introduction

In this approach, our goal is to test a new authentication system, at the same time we secure the information in a database by only storing the information representing. This way gives us more free memory at the Data base table.
The method is divided into two parts, namely:

First, we will make treatments on information stored in the database, the information or specifically the fingerprinting image is undergoing an indexing operation in the transformed domain using Singular Value Decomposition (SVD). This operation aims to decompose the image into three matrixes which represent each other a specific detail.

Second, after decomposing the image into three components according to the method chosen, we apply asymmetric encryption using Elliptic Curve Cryptosystem (ECC), on the index of the image with the information representative. The objective of encryption is to provide confidentiality and integrity of the index. This part encrypted will be stored in the database until the comparison with the query image.

Third, to authenticate, the user uses electronic or optical device to obtain a fingerprint. After collection, the query image undergoes an identical indexing operations treatment like the stored images. To enable authentication, we decrypt the data in the database and compared with the index of the query image. The comparison is performed by calculating the similarity of the two indexes by using the Euclidean distance.

2. Proposed approach

The flow diagram is given by the following figure:

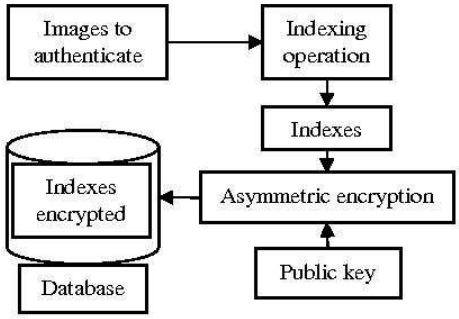


Fig. 1 Securing Scheme

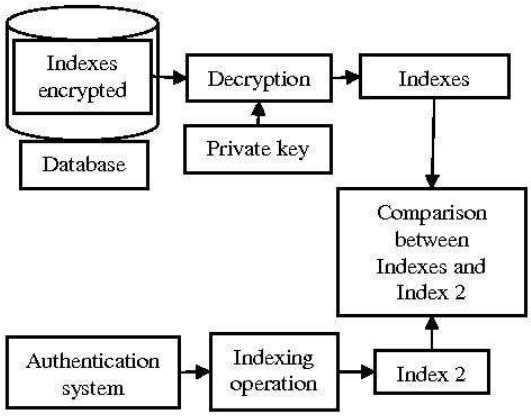
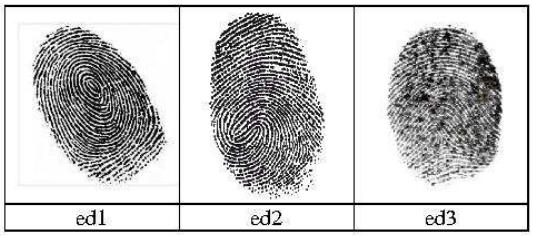


Fig. 2 Comparison Scheme

3. Results and interpretations

With SVD indexing method, we obtain three matrixes: $A = U * S * V^T$, such that the matrix S represent more the image and what is it that we encrypt.

Fingerprints used in the experiment are:



ed1 ed2 ed3

ed4 ed5 ed6

Fig. 3 Fingerprints use

We have the following table (Table 1) which shows the similarity between the query images and index images stored in the database (after decryption). Series of test are carried out on an Intel Pentium Dual Core 2.2 GHz with 3072 MB of RAM.

Table 1: Index similarity values

	ed1	*ed2*	*ed3*	*ed4*	*ed5*	*ed6*
ed1	0	6.7311 e+3	1.4921 e+3	1.0350 e+3	952.6909	358.9680
ed2	6.7311 e+3	0	7.5955 e+3	6.9186 e+3	6.9467 e+3	6.8422 e+3
ed3	1.4921 e+3	7.5955 e+3	0	2.1550 e+3	2.1039 e+3	1.5101 e+3
ed4	1.0350 e+3	6.9186 e+3	2.1550 e+3	0	650.5721	933.5138
ed5	952.6909	6.9467 e+3	2.1039 e+3	650.5721	0	973.9959
ed6	358.9680	6.8422 e+3	1.5101 e+3	933.5138	973.9959	0

Table 1 above shows the effectiveness of our approach because the probability of encountering two identical fingerprints of two different people tends to zero. It should be noted that our approach is invariant against geometric transformations such as rotation, scaling and translation of the query image because we used the values of the histogram of the indexed image. We can also see the execution time (in seconds) of the program: the indexing operation, encryption and decryption in the following table:

Table 02: computation speed of the indexing, encryption, decryption programs

Indexing time	*Encryption time*	*Decryption time*
0.0828 [s]	0.0346 [s]	0.0488 [s]

Comparing our indexing method by another method such as Principal Component Analysis (PCA), PCA has a large advantage on the computation speed compared to the indexing method by SVD, in the fact of, SVD uses additional operations when extracting the energy level of the image. But, the SVD method has high efficiency on the accuracy of signatures, that is to say the maximum energy representing information.

4. Conclusion

Our approach is based on how to represent the index of a fingerprint by the SVD indexing method. The problem of limited storage capacity is determined from the fact that only the representative part of the image is stored but not the whole picture. We also used a very powerful cryptosystem, which is asymmetric cryptography based on elliptic curves (ECC). It allows a level of security similar to the RSA cryptosystem, which is still the most used, using an encryption key of 1024 bits, against key seven times shorter in ECC. Finally, the evaluation is done by calculating the similarity between the query image and reference images decrypted.

Acknowledgments

Authors thank French Cooperation through the MADES project for its Sponsor and financial support.

References

[1] R. Norcen, M. Podesser, and A. Pommer, "Confidential Storage and Transmission of Medical Image Data", Computers in Biology and Medicine, 2003, pp. 277-292.

[2] S. Nandagopalan, B. S. Adiga., and N. Deepak, "A universal model for Content-Based Image Retrieval", World Academy of Science, Engineering and Technology, 2008.

[3] N. G Rao, V. V. Kumar, and V. V. Krishna, "Texture Based Image Indexing and Retrieval", International Journal of Computer Science and Network Security, Vol. 9, No. 5, May 2009.

[4] G. Zeng, "Face recognition with singular value decomposition », CISSE proceeding, 2006.

[5] L. Washington. Elliptic Curves, Numbers Theory and Cryptography, Chapmann and al, 2003.

Tahina E. Rakotondraina was born in Antsirabe, Madagascar on 1984. He received his M.S. degrees in Information Theory Student in 2010 at University of Antananarivo (Madagascar). He works as a Teacher assistant and a Ph.D. student at High School Polytechnic of Antananarivo. His currents research interests include Cryptography, multimedia, Information Hiding, VOIP. He is a co-author of two papers published in international journal.

Henri B. Razafindradina was born in Fianarantsoa, Madagascar, on 1978. He received, respectively, his M.S degree and Ph.D. in Computer Science and Information Engineering in 2005 and 2008. He served since 2010 as a professor at High Institute of Technology Diego Suarez, became an assistant lecturer in 2011. His current research interests include Images compression, multimedia, computer vision, information Hiding.

Annexe 5

Théorie de la complexité

A5.1 Présentation

La théorie de la complexité s'attache à connaître la difficulté (ou la complexité) d'une réponse par algorithme à un problème, dit algorithmique, posé de façon mathématique. Pour pouvoir la définir, il faut présenter ces trois concepts que sont les problèmes algorithmiques, les réponses algorithmiques aux problèmes, la complexité des problèmes algorithmiques.

A5.2 Problème algorithmique

Un problème algorithmique est un problème posé de façon mathématique, c'est-à-dire qu'il est énoncé rigoureusement dans le langage des mathématiques – le mieux étant d'utiliser le calcul des prédicats. Il comprend des *hypothèses*, des *données* et une *question*. On distingue :

- les **problèmes de décision** : ils posent une question dont la réponse est *oui* ou *non* ;

- les **problèmes d'existence ou de recherche d'une solution** : ils comportent une question ou plutôt une injonction de la forme *« trouver un élément tel que …»* dont la réponse consiste à fournir un tel élément.

A5.3 Réponse algorithmique

Dans chaque catégorie de problèmes ci-dessus, on dit qu'un problème a une réponse algorithmique si sa réponse peut être fournie par un algorithme. Un problème est *décidable* s'il s'agit d'un problème de décision – donc d'un problème dont la réponse est soit *oui* soit *non* et si sa réponse peut être fournie par un algorithme. Symétriquement, un problème est *calculable* s'il s'agit d'un problème d'existence et si l'élément calculé peut être fourni par un algorithme. La théorie de la complexité ne couvre que les problèmes décidables ou calculables et cherche à évaluer les ressources temps et espace mémoire mobilisées pour obtenir algorithmiquement la réponse.

A5.4 Complexité d'un problème algorithmique

La théorie de la complexité vise à savoir si la réponse à un problème peut être donnée très efficacement, efficacement ou au contraire être inatteignable en pratique (et en théorie), avec des niveaux intermédiaires de difficulté entre les deux extrêmes ; pour cela, elle se fonde sur une estimation – théorique – des temps de calcul et des besoins en mémoire informatique. Dans le but de mieux comprendre comment les problèmes se placent les uns par rapport aux autres, la théorie de la complexité établit des hiérarchies de difficultés entre les problèmes algorithmiques, dont les niveaux sont appelés des *« classes de complexité »*. Ces hiérarchies comportent des ramifications, suivant que l'on considère des calculs déterministes – l'état suivant du calcul est « déterminé » par l'état courant – ou non déterministes.

A5.5 Modèle de calcul

L'analyse de la complexité est étroitement associée à un **modèle de calcul**. L'un des modèles de calcul les plus utilisés est celui des machines abstraites dans la lignée du modèle proposé par Alan Turing en 1936.

Les deux modèles les plus utilisés en théorie de la complexité sont :

- La machine de Turing ;
- La machine RAM (Random Access Machine).

Dans ces deux modèles de calcul, un calcul est constitué d'étapes élémentaires ; à chacune de ces étapes, pour un état donné de la mémoire de la machine, une action élémentaire est choisie dans un ensemble d'actions possibles. Les *machines déterministes* sont telles que chaque action possible est unique, c'est-à-dire que l'action à effectuer est dictée de façon unique par l'état courant de celle-ci.

S'il peut y avoir plusieurs choix possibles d'actions à effectuer, la *machine* est dite *non déterministe*. Il peut sembler naturel de penser que les machines de Turing non déterministes sont plus puissantes que les machines de Turing déterministes, autrement dit qu'elles peuvent résoudre en un temps donné des problèmes que les machines déterministes ne savent pas résoudre dans le même temps.

A5.6 Complexité en temps et en espace

Sans nuire à la généralité, on peut supposer que les problèmes que nous considérons n'ont qu'une donnée. Cette donnée a une taille qui est un nombre entier naturel. La façon dont cette taille est mesurée joue un rôle crucial dans l'évaluation de la complexité de l'algorithme.

Ainsi si la donnée est elle-même un nombre entier naturel, sa taille peut être appréciée de plusieurs façons : on peut dire que la taille de l'entier p vaut p, mais on peut aussi dire qu'elle vaut $log(p)$ parce que l'entier a été représenté en numération binaire ou décimale ce qui raccourcit la représentation des nombres, ainsi *1 024* peut être représenté avec seulement onze chiffres binaires et quatre chiffres décimaux et donc sa taille est *11* ou *4* et non pas de l'ordre de *1 000*.

Le but de la complexité est de donner une évaluation du temps de calcul ou de l'espace de calcul nécessaire en fonction de cette taille, qui sera notée *n*. L'évaluation des ressources requises permet de répartir les problèmes dans des classes de complexité.

Pour les machines déterministes, on définit la classe $TIME(t(n))$ des problèmes qui peuvent être résolus en temps $t(n)$. C'est-à-dire pour lesquels il existe au moins un algorithme sur une machine déterministe résolvant le problème en temps $t(n)$. Le temps est le nombre de transitions sur machine de Turing ou le nombre d'opérations sur machine RAM, mais en fait ce temps n'est pas une fonction précise, mais c'est un ordre de grandeur, on parle aussi d'évaluation asymptotique, ainsi pour un temps qui s'évalue par un polynôme ce qui compte c'est le degré du polynôme, si ce degré est *2*, on dira que l'ordre de grandeur est en $O(n^2)$, que la complexité est *quadratique* et que le problème appartient à la classe $TIME(n^2)$.

Notation	Type de complexité
$O(1)$	complexité constante (indépendante de la taille de la donnée)
$O(log(n))$	complexité logarithmique
$O(n)$	complexité linéaire

$O(n.log(n))$	complexité quasi-linéaire
$O(n^2)$	complexité quadratique
$O(n^3)$	complexité cubique
$O(n^p)$	complexité polynomiale
$O(n^{log(n)})$	complexité quasi-polynomiale
$O(2^n)$	complexité exponentielle
$O(n!)$	complexité factorielle

Tableau A5.1 : *Types de complexité*

A5.7 Les quatre familles de classes de complexité en temps et en espace

Suivant qu'il s'agit de temps et d'espace, de machines déterministes ou non déterministes, on distingue quatre classes de complexité.

$TIME(t(n))$ est la classe des problèmes de décision qui peuvent être résolus en temps de l'ordre de grandeur de $t(n)$ sur une machine déterministe.

$NTIME(t(n))$ est la classe des problèmes de décision qui peuvent être résolus en temps de l'ordre de grandeur de $t(n)$ sur une machine non déterministe.

$SPACE(s(n))$ est la classe des problèmes de décision qui requièrent pour être résolus un espace de l'ordre de grandeur de $s(n)$ sur une machine déterministe.

$NSPACE(s(n))$ est la classe des problèmes de décision qui requièrent pour être résolus un espace de l'ordre de grandeur de $s(n)$ sur une machine non déterministe.

A5.8 Classes de complexité

Dans ce qui suit nous allons définir quelques classes de complexité parmi les plus étudiées en une liste qui va de la complexité la plus basse complexité à la complexité la plus haute. Il faut cependant avoir à l'esprit que ces classes ne sont pas totalement ordonnées.

Commençons par la classe constituée des problèmes les plus simples, à savoir ceux dont la réponse peut être donnée en *temps constant*. Par exemple, la question de savoir si un nombre entier est positif peut être résolue sans vraiment calculer, donc en un temps indépendant de la taille du nombre entier, c'est la plus basse des classes de problèmes.

La classe des *problèmes linéaires* est celle qui contient les problèmes qui peuvent être décidés en un temps qui est une fonction linéaire de la taille de la donnée. Il s'agit des problèmes qui sont en $O(n)$.

Souvent au lieu de dire « un problème est dans la classe *C* » on dit plus simplement « le problème est dans *C* ».

A5.8.1 Classes L et NL

Un problème de décision qui peut être résolu par un algorithme déterministe en espace *logarithmique* par rapport à la taille de l'instance est dans *L*.

Avec les notations introduites plus haut, $L = SPACE(log(n))$. La classe *NL* s'apparente à la classe L mais sur une machine non déterministe ($NL = NSPACE(log(n))$.

Par exemple savoir si un élément appartient à un tableau trié peut se faire en espace logarithmique.

A5.8.2 Classe P

Un problème de décision est dans *P* s'il peut être décidé sur une machine déterministe en temps *polynomial* par rapport à la taille de la donnée. On qualifie alors le problème de polynomial, c'est un problème de complexité $O(n^k)$ pour un certain *k*.

Un exemple de problème polynomial est celui de la connexité dans un graphe. Étant donné un graphe à s sommets (on considère que la taille de la donnée, donc du graphe est son nombre de sommets), il s'agit de savoir si toutes les paires de sommets sont reliées par un chemin.

Un algorithme de parcours en profondeur construit un arbre couvrant du graphe à partir d'un sommet. Si cet arbre contient tous les sommets du graphe, alors le graphe est connexe.

Le temps nécessaire pour construire cet arbre est en au plus $c.s^2$ (où c est une constante), donc le problème est dans la classe *P*.

On admet, en général, que les problèmes dans *P* sont ceux qui sont facilement solubles

A5.8.3 Classe NP et classe Co-NP (complémentaire de NP)

La classe *NP* des problèmes ***N*on-déterministes *P*olynomiaux** réunit les problèmes de décision qui peuvent être décidés sur une machine non déterministe en temps polynomial.

De façon équivalente, c'est la classe des problèmes qui admettent un algorithme polynomial capable de tester la validité d'une solution du problème, on dit aussi capable de construire un certificat. Intuitivement, les problèmes dans *NP* sont les problèmes qui peuvent être résolus en énumérant l'ensemble des solutions possibles et en les testant à l'aide d'un algorithme polynomial.

Par exemple, la recherche de cycle Hamiltonien dans un graphe peut se faire à l'aide de deux algorithmes :

- le premier engendre l'ensemble des cycles (en temps exponentiel, classe *EXPTIME*, voir ci-dessous) ;
- le second teste les solutions (en temps polynomial).

Ce problème est donc de la classe *NP*.

La classe duale de la classe *NP*, quand la réponse est *non*, est appelée *Co-NP*.

A5.8.4 Classe PSPACE

La classe $PSPACE$ est celle des problèmes décidables par une machine déterministe en espace polynomial par rapport à la taille de sa donnée. On peut aussi définir la classe $NSPACE$ ou $NPSPACE$ des problèmes décidables par une machine non déterministe en espace polynomial par rapport à la taille de sa donnée. Par le théorème de Savitch, on a $PSPACE = NPSPACE$, c'est pourquoi on ne rencontre guère les notations $NSPACE$ ni $NPSPACE$.

A5.8.5 Classe EXPTIME

La classe $EXPTIME$ rassemble les problèmes décidables par un algorithme déterministe en temps exponentiel par rapport à la taille de son instance.

A5.8.6 Classe NC (Nick's Class)

La classe **NC** est la classe des problèmes qui peuvent être résolus en temps **poly-logarithmique** (c'est à dire résolus plus rapidement qu'il ne faut de temps pour lire séquentiellement leurs entrées) sur une **machine parallèle** ayant un nombre **polynomial** (c'est à dire raisonnable) de processeurs.

Intuitivement, un problème est dans **NC** s'il existe un algorithme pour le résoudre qui peut être parallélisé et qui gagne à l'être. C'est à dire, si la version parallèle de l'algorithme (s'exécutant sur plusieurs processeurs) est significativement plus efficace que la version séquentielle.

Par définition, **NC** est un sous ensemble de la classe $\boldsymbol{P}$ ($\boldsymbol{NC} \subseteq \boldsymbol{P}$) car une machine parallèle peut être simulée par une machine séquentielle.

Mais on ne sait pas si $\boldsymbol{P} \subseteq \boldsymbol{NC}$ (et donc si $\boldsymbol{NC} = \boldsymbol{P}$). On conjecture que non, en supposant qu'il existe dans **P** des problèmes dont les solutions sont intrinsèquement non parallélisables.

A5.9 Problèmes C-complets ou C-difficiles

Soit **C** une classe de complexité (comme **P**, **NP**, etc.). On dit qu'un problème est **C-complet** ou **C-difficile** si ce problème est au moins aussi difficile que tous les problèmes dans **C**. Formellement on définit une notion de réduction : soient Π et Π'deux problèmes ; une réduction de Π' à Π est un algorithme (ou une machine) d'une complexité qu'on sait être inférieure à celle de la classe **C** transformant toute instance de Π' en une instance de Π.

Ainsi, si l'on a un algorithme pour résoudre Π, on sait aussi résoudre Π', mais de plus, si la complexité de Π est au moins celle de la classe **C**, on peut dire que Π est donc au moins aussi difficile à résoudre que Π'.

Π est alors **C-complet** ou **C-difficile** si pour tout problème Π' de **C**, Π' se réduit à Π.

Pour les problèmes **NP-complet** ou **NP-difficile** on s'autorise uniquement des réductions dans P, c'est-à-dire que l'algorithme qui calcule le passage d'une instance de Π' à une instance de Π est polynomial. Quand on parle de problèmes **P-complet** ou **P-difficile** on s'autorise uniquement des réductions dans $LOGSPACE$.

On qualifie de **NP-complet** les problèmes décisionnels, c'est-à-dire que la réponse est de type binaire (oui/non, vrai/faux, 1/0,...). On qualifie de **NP-difficile** les problèmes d'optimisation, c'est-à-dire que la réponse est de type numérique. A un problème d'optimisation **NP-difficile**, est associé un problème de décision **NP-complet**, mais dire qu'un problème **NP-difficile** est aussi **NP-complet** est un abus de langage car il n'y a pas de comparaison possible.

A5.10 Réduction de problèmes et problèmes NP-complets

Pour montrer qu'un problème Π est **C**-difficile pour une classe **C** donnée, il y a deux façons de procéder : ou bien montrer que tout problème de **C** se réduit à Π, ou bien montrer qu'*un* problème **C**-difficile se réduit à Π. Par exemple, démontrons que le problème de la recherche de circuit hamiltonien dans un graphe orienté est NP-Complet.

Le problème est dans *NP*, car on peut trouver un algorithme pour le résoudre à l'aide d'une machine non déterministe, par exemple en engendrant (de façon non déterministe) un circuit, puis en testant au final s'il est hamiltonien.

On sait que le problème de la recherche d'un cycle hamiltonien dans un graphe non orienté est **NP-difficile**. Or un graphe non orienté peut se transformer en un graphe orienté en créant deux arcs opposés pour chaque arête.

Il est donc possible de ramener le problème connu, NP-difficile, à savoir chercher un cycle hamiltonien dans un graphe non orienté, au problème que nous voulons classer, à savoir chercher un circuit hamiltonien dans un graphe orienté.

Le problème de l'existence d'un circuit hamiltonien est donc NP-difficile.

A5.11 Le problème ouvert P = NP

On a clairement $\boldsymbol{P \subseteq NP}$ car un algorithme déterministe est un algorithme non déterministe particulier, ce qui, dit en mots plus simples, signifie que si une solution peut être calculée en temps polynomial, alors elle peut être vérifiée en temps polynomial. En revanche, la réciproque : $\boldsymbol{P \subseteq NP}$, qui est la véritable difficulté de l'égalité $P = NP$ est un problème ouvert central d'informatique théorique. Il a été posé en 1970 indépendamment par Stephen Cook et Leonid Levin. La plupart des spécialistes conjecturent que les problèmes NP-complets ne sont pas solubles en un temps polynomial. À partir de là, plusieurs approches ont été tentées.

Des algorithmes d'approximation (ou heuristiques) permettent de trouver des solutions approchées de l'optimum en un temps raisonnable pour un certain nombre de programmes. Dans le cas d'un problème d'optimisation on trouve généralement une réponse correcte, sans savoir s'il s'agit de la meilleure solution.

Des algorithmes stochastiques : en utilisant des nombres aléatoires on peut «forcer» un algorithme à ne pas utiliser les cas les moins favorables, l'utilisateur devant préciser une probabilité maximale admise que l'algorithme fournisse un résultat erroné.

Des algorithmes par séparation et évaluation permettent de trouver la ou les solutions exactes. Le temps de calcul n'est bien sûr pas borné polynomialement mais, pour certaines classes de problèmes, il peut rester modéré pour des instances relativement grandes.

L'approche de la complexité paramétrée consiste à identifier un paramètre qui, dans le cas où il reste petit, permet de résoudre rapidement le problème. En particulier les algorithmes FPT en un paramètre *k* permettent de résoudre un problème en temps proportionnel au produit d'une fonction quelconque de *k* et d'un polynôme en la taille de l'instance, ce qui fournit un algorithme polynomial quand *k* est fixé.

On peut restreindre la classe des problèmes considérés à une sous-classe suffisante, mais plus facile à résoudre.

BIBLIOGRAPHIE

[1] A.J. Menezes, P.C. Oorschot, S.A. Vanstone, *« Handbook of Applied Cryptography »*, CRC Press, 1996.

[2] W. Diffie, M. E. Hellman, « *New directions in cryptography* », IEEE Trans. Inform. Theory, vol. 22, n°6, pp. 644-654, 1976.

[3] N. Koblitz, *« A Course in Number Therory and Cryptography »*. Springer-Verlag, 1987.

[4] C.E. Shannon, *« A mathematical theory of communication »*, Bell System Technical Journal, 1948.

[5] B. A. Robert, *"Information Theory"*, Urbana, Illinois, 1965

[6] J. H. Marcia, *"Error Control Coding"*, Pearson Prentice Hall, 2004

[7] H. Y. C. Louis, S. Yeneng, *"Coding Theory And Cryptology"*, World Scientific Publishing, 2002

[8] L. B. Norman, *"Codes: An Introduction to Information Communication and Cryptography"*, Springer, 2008

[9] W. Dominic, *" Codes and Cryptography"*, Clarendon Press-Oxford, 1988

[10] J. Hoffstein, J. Pipher, J. H. Silverman, *"An Introduction to Mathematical Cryptography"*, Springer,2000

[11] M. E. Martínez, C. Munuera, *"Advances In Algebraic Geometry Codes"*, World Scientific, 2008

[12] T. Shaska, W. C. Huffman , D. Joyner, V. Ustimenko , *"Advances In Coding Theory And Cryptography"*, World Scientific, 2007

[13] F. Arnault, *« Théorie des nombres et cryptographie »*, Cours Université de Limoges, mai 2002.

[14] P. Rouchon, *« Mathématiques Discrètes »*, Ecole Nationale Supérieure de Paris, Nov 2003.

[15] N. Koblitz, *"Algebraic Aspects of Cryptography"*, Springer, 1999

[16] G. Chassé, *« Cryptographie Mathématique Algorithme »*, Ecole des Mines de Nantes, 2002.

[17] F. Leprévost, *« Les standard cryptographique du XXIè siècle et IEEE-1363 »*, Cours Université Paris 6, Juil 2000.

[18] B. Brauer, G. Rozenberg, A. Salomaa, *"Complexity Theory and Cryptology"*, Springer, 1998

[19] A Myasnikov, V. Shpilrain, A. Ushakov, *"Group-based Cryptography"*, Springer, 2008

[20] A. B. Johannes, "*Introduction to cryptography"*, Springer, 2000

[21] F. Bergeron, A. Goupil, *« La cryptographie de l'Antiquité à l'Internet »*, Université du Quebec , Montreal, Fév 2006.

[22] R. A. Mollin, *" An introduction to cryptography Second Edition"*, Chapman and Hall/CRC, 2007

[23] S. Vaudenay, *"A Classical Introduction To Cryptography Applications For Communications Security"*, Springer, 2006

[24] P. Garrett, *"Making, Breaking Codes : An Introduction to Cryptology"*, Prentice Hall, 2001

[25] P. Perret, S. Richard, *« La cryptographie »*, Clusir, Fév 2007.

[26] G. Labouret, *« Introduction à la cryptographie » Cabinet Hervé Schauer Consultants, 2001*

[27] T. Baigneres, P. Junod, L. Yi, J. Monnerat, S. Vaudenay, *"A Classical Introduction to Cryptography-Exercise Book"*, Springer, 2006

[28] M. Wenbo, *"Modern Cryptography: Theory and Practice"*, Prentice Hall, 2003

[29] S. Douglas, *"Cryptography: Theory and Practice"*, CRC Press, 1995

[30] F. Dramaix, D. Broek, V. Wens, *« La cryptographie quantique »*, Printemps des sciences, 2003.

[31] B. Schneier, *" Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C"*, John Wiley & Sons, 1996

[32] L.S. Van, M. Malengreaux, *« Les mystères de la cryptographie »* Collège du Sartay, Embourg, 2005.

[33] Denning, E. Dorothy, "*Cryptography and data security"*, Addison Wesley, Purdue University, 1982

[34] R.L. Rivest, A. Shamir, L. Adleman, *« Cryptographic communications system and method »*, U.S. Patent, Sept 1983.

[35] J.L. Massey, J.K. Omura, *« Method and apparatus for maintaining the privacy of digital messages conveyed by public transmission »*, U.S. Patent, Jan 1986.

[36] R.L. Rivest, *« The MD5 message-digest algorithm »*, Internet Request for Comments 1321, RFC 1321, Avril 1992.

[37] R.L. Rivest, A. Shamir, L. Adleman, *« A method for obtaining digital signature and public key cryptosystems »*. Comm. ACM, vol. 21, pp.120-126, Feb 1978.

[38] G. Tsudik, *« Message authentication with one-way hash functions »*, Computer Communication Review, 1992.

[39] W. Diffie, P. C. V. Oorschot, M.J. Wiener, *« Authentication and authenticated key exchanges »*, Designs, Codes and Cryptography, 1992.

[40] T. El Gamal, *« A public key cryptosystem and a signature scheme based on discrete logarithms »*, IEEE Transactions on Information Theory, vol.31, 1985.

[41] H. N. Jason, "Cryptanalysis of RSA and Its Variants", Chapman and Hall, 2010

[42] R.J. McEliece, *« A public-key cryptosystem based on algebraic coding theory »*, Jet Propulsion Laboratory, Pasadena, California, 1978.

[43] D. Hankerson, A. Menezes, S. Vanstone, *« Guide to Elliptic Curve Cryptography »*, Springer, 2004.

[44] L.C. Washington, *« Elliptic Curves Number Theory and Cryptography »*, Chapman & Hall/CRC, 2003.

[45] N. Koblitz, *« Introduction to Elliptic Curves and Modular Forms »*, Springer-Verlag, GTM 97, 1987.

[46] G. Mercier, C. Roux, G. Martineau, « *Technologies du Multimédia »*, cours ENST Bretagne, dpt ITI, France, 2003

[47] S. Znaty, J. L. Dauphin, "*IP Multimedia Subsystem : Principes et Architecture*", EFORT, 2007

[48] J. Vriendt, G. Vinagre, A. V. Ewijk, *"Multimedia broadcast and multicast services in 3G mobile networks"*, Alcatel Telecommunications Revew, 2004

[49] J. Marconi, M. Rodrigues, *« Transfert sécurisé d'images par combinaison de techniques de compression, cryptage et marquage »*, Thèse Doctorat, Université Montpellier II, 2006

[50] J. D. Gibson, *"Handbook of Image and Video processing"*, Academic Press, 2000

[51] P. Dwayne, *"Image Processing in C-Second Edition"*, R&D Publications, 2000

[52] I. E. G. Richardson, *"H.264 and MPEG-4 Video Compression- Video Coding for Next-generation Multimedia"*, Wiley, 2003

[53] F. Luxereau, *"Compression du signal audio-visuel"*, Dunod, Paris, 2008.

[54] A. Gersho, R. M. Gray, *"Vector quantization and signal compression"*, Kluwer, Boston, 1992.

[55] S. S. Maniccam, N. G. Bourbakis, *"Lossless image compression and encryption using SCAN"*, Pattern Recognition, pp.1229-1245, 2001.

[56] Z. Wang, A. C. Bovik, L. Lu, *"Why is image quality assessment so difficult ?"*, Conf. on Acoustics, Speech and Signal Processing, vol 4, page 3313-3316, 2002.

[57] A. Ninassi, O. Le Meur, Le Callet P., *"Task impact on the visual attention in subjective image quality assessment"*, In EUSIPCO-06, Florence, Italy, 2006.

[58] A. B. Watson, G. Y. Yang, J. A. Solomon, *"Visibility of wavelet quantization noise"*, IEEE Trans. Image Proc, 6(8), pp.1164-1175, 1997.

[59] B. Newman, *"Secrets of German Espionage"*, Robert Hale Ltd, London, 1940.

[60] D. Kahn, *"The Histories - Terpsichore - Polymnia. J.M"*, Dent & Sons Ltd, London England, 1992.

[61] D. Sieberg, *"Bin Laden exploits technology to suit his needs"*, Technical report, CNN, New York, USA, September 2001.

[62] S. Katzenbeisser, F. A. P. Petitcolas, *"Information Hiding Techniques for Steganography and Digital Watermarking"*. Artech House, London, 2000.

[63] W. Bender, D. Gruhl, N. Morimoto, *"Techniques for data hiding"*, IBM Systems Journal, pp.131-336, 1996.

[64] N. F. Johnson, S. Jajodia, *"Exploring Steganography : Seeing the Unseen"*, IEEE Computer, pp.26-34, 1998.

[65] J. M. Rodrigues, J. R. Rios, W. Puech, *"SSB-4 System of Steganography using bit 4"*, In WIAMIS04, Portugal, 2004.

[66] J. Fridrich, *"A New Steganographic Method for Palette-Based Images"*, In PICS, Georgia, USA, 1999.

[67] H. Niimi, M. Noda, E. Kawaguchi, *"Luminance Quasi-Preserving Color Quantization for Digital Steganography to Palette-Based Images"*, In ICPR02, vol 1, pp.251-254, 2002.

[68] M. Whelan, F. Balado, C. Guenole, *"Iterative Estimation of Amplitude Scaling on Distortion Compensated Dither Modulation"*, In Security, Steganography and Watermarking of Multimedia Contents VII, San Jose, CA, USA, 2005.

[69] J. L. Toutant, W. Puech, C. Fiorio, *« Amélioration de l'invisibilité par adaptation de la quantification aux données à insérer »*, In GRETSI'05, 2005.

[70] C. C. Chang, C. Tung-Shou, C. Lou-Zo, *"A steganographic method based upon JPEG and quantization table modification"*, Inf. Sci. Inf. Comput. Sci., 141(1-2), pp.123-138, 2002.

[71] C. C. Thien, J.C. Lin, *"A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function"*, Pattern Recognition, pp.2875-2881, 2003.

[72] K. Solanki, N. Jacobsen, S. Chandrasekaran, *"High Volume Data Hiding in Images : Introducing Perceptual Criteria Into Quantization"*, In ICASSP02, 2002.

[73] H. Noda, J. Spaulding, M. N. Shirazi, *" Application of bit-plane decomposition steganography to JPEG2000 encoded images"*, SPLetters, pp.410-413, 2002.

[74] M. Kurosaki, K. Munadi, H. Kiya, *"Error correction using data hiding technique for JPEG2000 images"*, In ICIP03, pp.473-476, 2003.

[75] H. Alasady, M. Ibnkahla, *"A Simple Data Pre-Distortion Technique for Satellite Communications : Design and Implementation on Altera DSP Board"*, Technical report, GSPx'04, 2004.

[76] S. Katzenbeisser, F. A. P. Petitcolas, *"Information Hiding Techniques for Steganography and Digital Watermarking"*, Artech House, London, 2000.

[77] K. Solanki, K. Sullivan, *"Statistical Restoration for Robust and Secure Steganography"*, IEEE International Conference on Image Processing, 2005.

[78] N. Provos, *"Defending Against Statistical Steganalysis"*, In USENIX Security Symposium, pp.323-335, 2001.

[79] S. Dumitrescu, X. Wu, *"LSB steganalysis based on high-order statistics"*, In MM&Sec '05 : Proceedings of the 7th workshop on Multimedia and security, pp.25-32, New York, 2005.

[80] F. Davoine, S. Pateux, *«Tatouage de documents audiovisuels numériques »*, Hermès Science Publications, Lavoisier, France, 2003.

[81] N. Nikolaidis, I. Pitas, *"Digital Image Watermarking : An Overview"*, In ICMCS, vol 1, pp. 1-6, 1999.

[82] G. S. El-Taweel, H. M. Onsi, M. Samy, *"Secure and Non-Blind Watermarking Scheme for Color Images"*, ICGST International Journal on Graphics, Vision and Image Processing, SI1, 2005.

[83] R. Safabakhsh, S. Zaboli, A. Tabibiazar, *"Digital watermarking on still images using wavelet transform"*, In IEEE International Conference on Information Technology, vol 1, pp. 671-675, 2004.

[84] P. W. Wong, N. Memon, *"Secret and public key image watermarking schemes for image authentication and ownership verification"*, IEEE Transactions on Image Processing, pp.1593-1601, 2001.

[85] T. Y. Kim, K. Taejeong, C. Hyuk, *"Correlation-based asymmetric watermark detector"*, In IEEE - Inter. Conf. on Information Technology Coding and Computing, pp.564-568, Las Vegas, 2003.

[86] Y. H. M. Chen, *"A fragile watermark error detection scheme for wireless video communications"*, IEEE Transactions on Multimedia, pp.201-211, 2005.

[87] C. T. Li, *"Digital fragile watermarking scheme for authentication of JPEG images"*, IEEE Proceedings Vision, Image and Signal Processing, 2004.

[88] A. Ramalingam, S. Krishnan, *"Robust image watermarking using a chirp detection-based technique"*, IEEE Proceedings Vision, Image and Signal Processing, pp.771-778, December 2005.

[89] J. Delhumeau, T. Furon, N. Hurley, *"Improved Polynomial Detectors for Side-Informed Watermarking"*, In Proc. of Security and Watermarking of Multimedia Contents V, SPIE Electronic Imaging, Santa Clara, CA, USA, 2003.

[90] P. M. Saraju, N. Ranganathan, K. N. Ravi, *"VLSI Implementation of Visible Watermarking for a Secure Digital Still Camera Design"*, In Proceedings. 17th International Conference on VLSI Design, pp.1063-1068, 2004.

[91] H. Yongjian, J. Huang, K. Sam, *"Image Fusion Based Visible Watermarking Using Dual-Tree Complex Wavelet Transform"*, In IWDW, pp.86-100, 2003.

[92] J. Fridrich, G. Dur, *"Invertible authentication"*, In SPIE Security and watermarking of multimedia contents, San Jose, California, 2001.

[93] L. G. Gaëtan, S. Pateux, *"Wide Spread Spectrum Watermarking with Side Information and Interference Cancellation"*, In Proceedings of SPIE, Santa Clara, CA, U.S.A, January 2003.

[94] P. Bas, B. Roue, J. M. Chassery, *« Tatouage d'images couleur additif : vers la sélection d'un espace d'insertion optimal »*, In Coresa03, vol 1, 2003.

[95] M. Barni, F. Bartolini, *"Watermarking Systems Engineering, Signal Processing and Communication Series"*, Marcel Dekker Inc., New York, USA, 2004.

[96] J. Eggers, R. Buml, R. Tzschoppe, *"Scalar Costa Scheme for Information Embedding"*. IEEE Transactions on Signal Processing, 2003.

[97] I. Cox, M. Miller, J. Bloom, *"Digital watermarking"*, Morgan Kaufmann Publishers Inc., San Fransisco, USA, 2002.

[98] H. Joumaa, F. Davoine, *« Tatouage substitutif d'images intégrant un masque de pondération visuelle »*, In CORESA, Lyon, France, 2003.

[99] F. Davoine, S. Pateux, *« Tatouage de documents audiovisuels numériques »*, Hermès Science Publications, Lavoisier, France, 2003.

[100] A. Parisis, P. Carré, F. Maloigne, *« Watermarking et couleur : étude de différents espaces de représentation couleur »*, In Coresa01, Dijon, France, 2001.

[101] P. Bas, B. Roue, J.M. Chassery, *« Tatouage d'images couleur additif : vers la sélection d'un espace d'insertion optimal »*, In Coresa03, Lyon, France, 2003.

[102] F. Hartung, B. Girod, *"Watermarking of Uncompressed and Compressed Video"*, Signal Processing, pp.283-333, 1998.

[103] W. Bender, D. Gruhl, N. Morimoto, *"Techniques for data hiding"*, IBM Systems Journal, pp.133-336, 1996.

[104] D. Kundur, D. Hatzinakos, *"A Robust Digital Image Watermarking Scheme Using the Wavelet Based Fusion"*, In IEEE-ICIP'97, vol 1, pp.544-547, 1997.

[105] X. Xia, C. Boncelet, G. Arce, *"A Multiresolution Watermark for Digital Images"*, In IEEE-ICIP'97, vol 1, pp.548-551, Santa Barbara, USA, 1997.

[106] P. Bas, *« Méthodes de tatouage d'images fondées sur le contenu »*, PhD thesis, Institut National Polytechnique de Grenoble, 2000.

[107] D. Upman, *"Jpeg-Jsteg Computer Software"*, Technical report, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.

[108] L. Guillemot, Moureaux J-M., *« Tatouage d'images : une nouvelle approche basée sur une méthode de compression »*, In CORESA, pp.253-256, Lyon, France, 2003.

[109] D. Delanay, B. Macq, *"Generalized 2-D cyclic patterns for secret watermark generation"*, In IEEE ICIP, volume 2, pp.77-80, 2000.

[110] M. Kutter, *"Watermarking resisting to translation, rotation and scaling"*, In SPIE, Multimedia systems and applications, vol 3528, pp.423-431, 1999.

[111] F. Davoine, P. Bas, P. Hebert, *« Watermarking et Résistance aux déformations géométriques »*, In Coresa, 1999.

[112] P. Amat, W. Puech, *« Transfert sécurisée d'une ROI sans perte par une méthode d'insertion de données cachées robuste à la compression JPEG »*, In GRETSI, Louvain-La-Neuve, Belgique, 2005.

[113] G. Lo-Varco, W. Puech, M. Dumas, *"Content Based Watermarking for Securing Color Images"*, Journal of Imaging Science and Technology, pp.450-458, 2005.

[114] A. Manoury, *« Tatouage d'images numériques par paquets d'ondelettes »*, Université de Nantes, Page 35, Décembre 2001

[115] J. Ruanaidh, G. Csurka, F. Deguillaume, T. Pun, *« Tatouage d'images basé sur le Transformée de Fourier Discrète »*

[116] J. L. Dugelay, S. ROCHE, *« Introduction au tatouage d'images »*, Department of Multimedia Communications, EURECOM, pp.4-13, 2002.

[117] B. Martin, "Codage, Cryptologie et Applications", Presses Polytechniques et Universitaires Ramandes, 2003

[118] M. Wenbo, *"Modern Cryptography: Theory and Practice"*, Prentice Hall, 2003

[119] L.C. Washington, *« Elliptic Curves Number Theory and Cryptography »*, Chapman & Hall/CRC, 2003.

[120] D. Hankerson, A. Menezes, S. Vanstone, *« Guide to Elliptic Curve Cryptography »*, Springer, 2004.

[121] J. Walter, *«The role of ECDSA in wireless communication (implementation and evaluation of ECDSA on constrained devices) »*, Los Angeles, 2002.

[122] S. Goel,M. Bhattacharya, *«speech based dialog query system over Asterisk PBX Server»*, ICSPS, 2010

[123] H. Abdelnur, R. State, I. Chrisment, C. Popi, *«Assessing the security of VoIP Services*" in Integrated Network anagement, IM 2007. 10thIFIP/IEEE International Symposium on Integrated NetworkManagement, Munich, Germany, 21-25. IEEE, May 2007, pp. 373–382.

[124] P.Montoro, E. Casilari, *"A comparative study of VoIP Standards with Asterisk"*, Forth international conference On Digital telecommunication, 2009.

[125] V. N. G. J. Soares, P. A. C. Neves, and J. J. P. Rodrigues, "*Past, Present and Future of IP Telephony"*, Proc.of Communication Theory, Reliability, and Quality of Service (CTRQ'08), Bucharest (Romania), Jul.2008, pp.19-24.

[126] A. Gorti, *"A fault tolerant VoIP implementation based on open standards"*, Proc.of Sixth European Dependable Computing Conference (EDCC'06), Coimbra (Portugal), Oct.2006, pp.35-38.

[127] M.Z.Alam, S.Bose, M.M.Rahman, and M.A.Al-Mumin, *"Small Office PBX Using Voice Over Internet Protocol (VOIP)"*,Proc.of IEEE The 9th International Conference on Advanced Communication Technology (ICACT 2007), Vol.3, Gangwon-Do(Korea), Feb.2007, pp.1618-1622.

[128] E. Casilari, H. Montes, F. Sandoval, *"Modelling of Voice Traffic Over IP Networks"*, CSNDSP 2002, Network Communications K1.5, July 2002.

[129] R. Norcen, M. Podesser, A. Pommer, "*Confidential Storage and Transmission of Medical Image Data*", Computers in Biology and Medicine, pp. 277-292, 2003.

[130] J. Marconi, M. Rodrigues, *« Transfert sécurisé d'images par combinaison de techniques de compression, cryptage et marquage »*, Thèse Doctorat, Université Montpellier II, 2006.

[131] C. Gasquet, P. Witomski, « *Analyse de Fourier et applications* », Dunod, 1996.

[132] P. Duhamel and M. Vetterli, « *Fast Fourier transforms: a tutorial review and a state of the art* » Signal Processing, pp. 259–299, 1990.

[133] F. Deguillaume, S. Voloshynovskiy, S. Pereira, M. Madueno, *« Filigranage d'images digitales »*, Bulletin SEV/VSE 9/01, 2001.

[134] C. Cavaro-Menard « *Codage, compression et échange d'images* », DEA signaux et images en biologie et médecine, 2002.

[135] S.-M. Frédéric, « *Rapport pour l'Etudes d'Approfondissement : Stéganographie vs Tatouage* », Université Paris II, 2005.

[136] M. Vissac, J.-L. Dugelay, *« Un panorama sur l'indexation d'images fixes »*, Institut Eurocom, Dept.of Multimedia Communication, 2004.

[137] H. L. T. Nguyen, *« Recherche d'image basée sur le contenu sémantique »*, Rapport de TIPE, Institut de la Francophonie pour l'Informatique, Hanoi, Vietnam, Juillet 2005.

[138] A. Manzanera, *« Indexation d'images »*, Cours Master IA & D, ENSTA, Université Pierre et Marie Curie, 2004.

[139] V. Gouet-Brunet, *« Base de données multimédia, Introduction à la recherche par contenu visuel dans les banques d'images »*, cours CNAM, 2007.

[140] A. Shahbahrami, D. Borodin, B. Juurlink, *« Comparison between Color and Texture features for image retrieval »*, Computer Engineering Laboratory, Delft University of Technology, The Netherlands and Department of Computer Engineering, University of Guilan, Rasht, Iran, 2005.

[141] G. Zeng, « *Face recognition with singular value decomposition* », CISSE proceeding, 2006.

[142] H. B. Razafindradina, P. A. Randriamitantsoa, « *Tatouage robuste et aveugle dans le domaine des valeurs singulières* », Laboratoire LASM, pp. 5-7, janvier 2010.

[143] H. L. T. Nguyen, *« Recherche d'image basée sur le contenu sémantique »*, Rapport de TIPE, Institut de la Francophonie pour l'Informatique, Hanoi, Vietnam, Juillet 2005.

[144] J. Rosenberg, A. Johnston, M. Handley, *"SIP : Protocole d'initialisation de session"*, RFC 3261, 2002.

[145] D. Endler, M. Collier *"Voice Over IP Security Secrets & Solutions"*, McGraw-Hill/Osborne, 2007.

RENSEIGNEMENT SUR L'AUTEUR

Nom : RAKOTONDRAINA

Prénoms : Tahina Ezéchiel

Adresse : Lot AKTIC 76 Vontovorona
102 Antananarivo
Madagascar

Téléphone : +261 34 19 082 00

E-mail : tahina.ezechiel@gmail.com

Titre du mémoire :

MODELISATION ET ALGORITHMES DE TRANSFERT SECURISE D'INFORMATION

Nombre de page : 222

Nombre de tableaux : 20

Nombre de figures : 49

Mots clés : Cryptographie, Codage, Tatouage, VoIP, Sécurité, Image, SRTP, Voix

Directeur de mémoire: RANDRIAMITANTSOA Paul Auguste

Téléphone: +261 34 10 342 58

Mail : rpauguste@gmail.com

RESUME

Dans ce livre, nous contribuons à une mise en application de la théorie de l'information, du fait que nous avons associé la mathématique appliquée à la communication, le codage des informations multimédia, l'insertion de données cachées, la cryptographie et la VoIP. La philosophie que nous avons adopté, dans cette thèse, consiste à développer tous ces domaines dans le but de sécuriser le transfert des informations images et voix. L'objectif de notre recherche est donc de combiner ces méthodes, de modéliser et de proposer des nouvelles approches. Nous avons testé, ainsi, nos propositions sur un transfert sécurisé d'image, sur un système d'authentification et sur la sécurisation d'un appel dans un réseau VoIP.

ABSTRACT

In this book, we contribute to an implementation of the information theory, owing to the fact that we associated the mathematics applied to the communication, coding multimedia information, the insertion of hidden data, cryptography and VoIP. The philosophy that we adopted, in this thesis, consists to developing all these fields with an aim of making safe a transfer of information like images and voice. The objective of our research is thus to combine these methods, to model and propose new approaches. We have to test our proposals for a secure image transfer, on an authentication system and for securing calls in a VoIP network.