République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Larbi Tébessi - Tébessa
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département : Mathématiques et Informatique

Mémoire de fin d'études
Pour l'obtention du diplôme de MASTER 2

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Réseaux et Sécurité Informatique

Thème

Cryptage d'images pour une sécurité élevée de transmission sur le réseau

Présenté Par :
TOLBA Bouzid

Devant le jury :

Mr LAIMECHE Lakhdar	*MCA*	*Université Larbi Tébessi*	*Président*
Mr MERAOUMIA Abdallah	*MCA*	*Université Larbi Tébessi*	*Examinateur*
Mr ZEGGARI Ahmed	*MCB*	*Université Larbi Tébessi*	*Encadreur*

Date de soutenance : *15/09/2020*

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Larbi Tébessi - Tébessa
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département : Mathématiques et Informatique

Mémoire de fin d'études
Pour l'obtention du diplôme de MASTER 2

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Réseaux et Sécurité Informatique

Thème

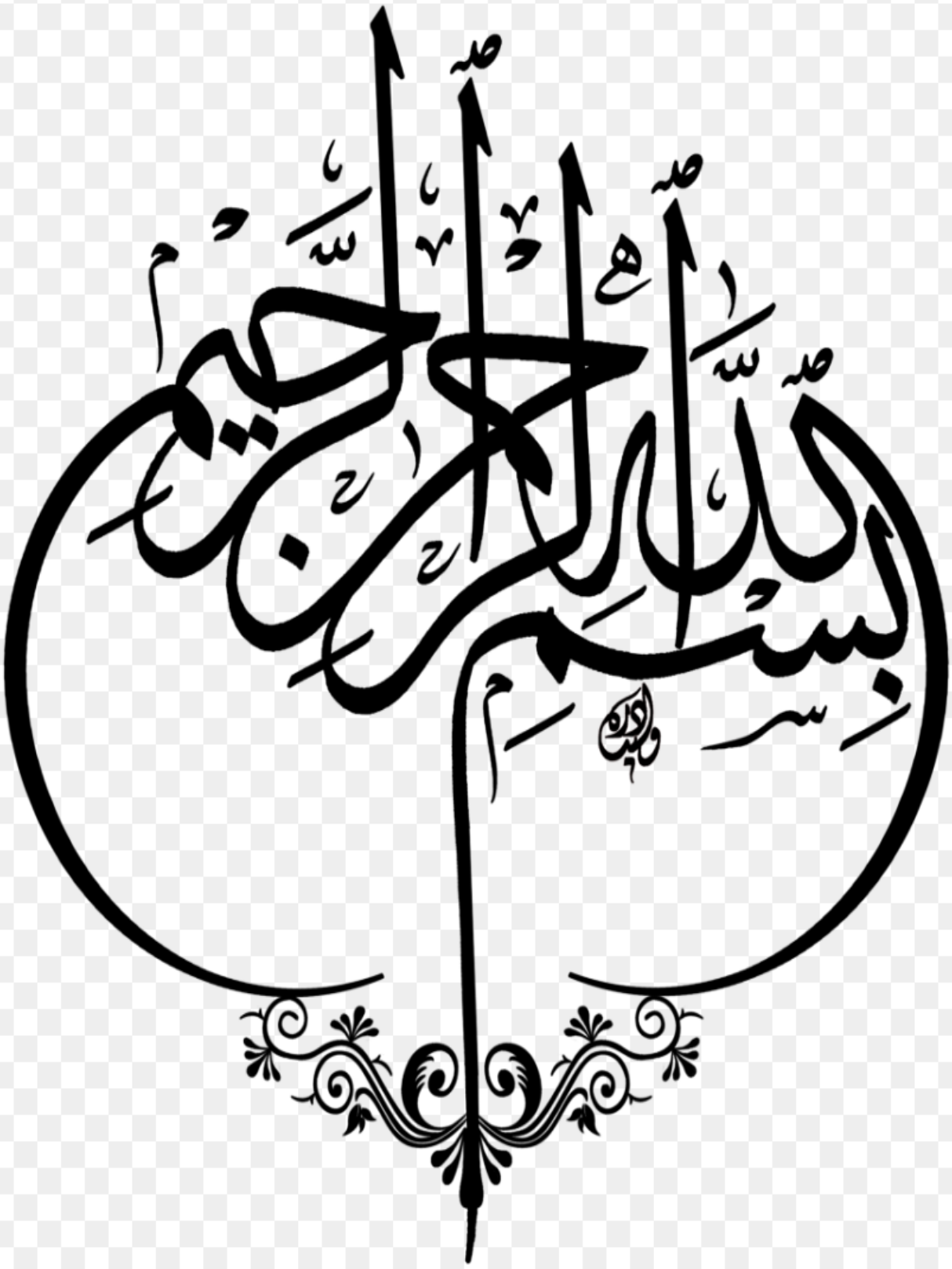
Cryptage d'images pour une sécurité élevée de transmission sur le réseau

Présenté Par :
TOLBA Bouzid

Devant le jury :

Mr LAIMECHE Lakhdar	*MCA*	*Université Larbi Tébessi*	*Président*
Mr MERAOUMIA Abdallah	*MCA*	*Université Larbi Tébessi*	*Examinateur*
Mr ZEGGARI Ahmed	*MCB*	*Université Larbi Tébessi*	*Encadreur*

Date de soutenance : *15/09/2020*


بسم الله الرحمن الرحيم

Remerciements

Tout d'abord, nous remercions le Dieu, notre créateur de nos avoir donné la force, la volonté et le courage afin d'accomplir ce travail modeste.

Ce mémoire achève mes études de master à l'université de Tébessa après avoir obtenu un diplôme d'ingénieur d'état en informatique à l'université d'Oum El Bouagui en juin 2008, cela représente pour moi l'occasion d'exprimer ma reconnaissance pour tous les personnes qui m'ont aidé à arriver jusqu'à cette étape.

Je souhaite exprimer ma vive gratitude au Mr **Zeggari Ahmed** pour m'avoir assuré un cadre de travail excellent et pour sa disponibilité, sa patience et sa grande expérience.

Je tiens également à remercier messieurs les membres de jury pour l'honneur qu'ils m'ont fait en acceptant de siéger à mon soutenance.

Je remercie l'ensemble des enseignants qui ont contribué à mon formation, ainsi que mon formidable promotion 2020 du département de mathématique et d'informatique, tout particulièrement aux étudiants de l'option RSI.

Finalement, je tiens à exprimer mon profonde gratitude à mon famille qui m'a toujours soutenu et à tout ce qui participe de réaliser ce mémoire.

Dédicaces

Je dédie ce modeste travail à :

A mes très chers parents, pour leurs assistantes, conseils, patience, soutien et sacrifices.

A mes très chers frères, et ma chère sœur.

A tous ceux que j'aime et qui m'aiment.

Je dédie également ce travail à ma grande famille TOLBA.

A toutes mes amies de l'université de Tébessa et de l'université d'Oum El Bouagui.

A tous ceux qui sont proches de mon cœur et qui m'encouragent à donner le meilleur en moi.

Bouzid

Résumé

Le développement rapide dans le traitement d'image numérique et les technologies de communication de réseau ont rendu la vie facile et en même temps a ajouté la complexité au niveau de sécurité surtout avec le développement des technologies de piratage numérique. Il est nécessaire de protéger l'information d'image envoyée de l'utilisation illégale.

Cet mémoire propose une technique de sécurité pour les images confidentielles qui est la combinaison de deux techniques, la première est la compression d'image qui est basée sur la transformation en ondelettes qui compressera l'image confidentielle et réduira sa taille sans dégrader la qualité à un niveau inacceptable, ce qui permet diminuer également le temps nécessaire pour que les images soient cryptées et transmise sur le réseau, la deuxième est la cryptographie basée sur une clé symétrique qui va crypter l'image confidentielle, qu'assure une sécurité élevée de transmission sur le réseau.

Mots clés : Sécurité, Cryptographie, Clé symétrique, Cryptage, Décryptage, Transformation en ondelettes, Compression d'image.

Abstract

The rapid development in digital image processing and network communication technologies have made life easy and at the same time added complexity to the level of security especially with the development of digital hacking technologies. It is necessary to protect the sent image information from illegal use.

This thesis proposes a security technique for confidential images which is the combination of two techniques, the first is image compression which is based on the transformation into wavelets which will compress the confidential image and reduce its size without degrading the quality an unacceptable level, which also makes it possible to reduce the time necessary for the images to be encrypted and transmitted over the network, the second is cryptography based on a symmetric key which will encrypt the confidential image, which ensures high transmission security on the network.

Keywords: Security, Cryptography, Symmetric Key, Encryption, Decryption, Wavelet Transform, Image Compression.

ملخص

أدى التطور السريع في معالجة الصور الرقمية وتقنيات الاتصال الشبكي إلى تسهيل الحياة وفي نفس الوقت أضاف تعقيدًا إلى مستوى الأمان خاصةً مع تطوير تقنيات القرصنة الرقمية. من الضروري حماية معلومات الصورة المرسلة من الاستخدام غير القانوني.

تقترح هذه الرسالة تقنية لتأمين الصور السرية وهي مزيج من تقنيتين ، الأولى هي ضغط الصورة الذي يعتمد على التحويل بالموجات والتي تعمل على ضغط الصورة السرية وتقليل حجمها دون التقليل من الجودة إلى مستوى غير مقبول، ما يسمح أيضًا بتقليل الوقت اللازم لتشفير الصور وإرسالها عبر الشبكة، والثاني هو التشفير المستند إلى مفتاح متماثل يقوم بتشفير الصورة السرية، مما يضمن أمان إرسال عاليًا على الشبكة.

الكلمات المفتاحية: الأمان ، التشفير ، المفتاح المتماثل ، التشفير ، فك التشفير ، التحويل بالموجات ، ضغط الصور.

Table des matières

Liste des figures:

Liste des tableaux

Liste des acronymes et abréviations

DWT	Discrete Wavelet Transform
CWT	Continues Wavelet Transform
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
RLE	Run Length Encoding
LZW	Lempel Ziv Welch
JPEG	Joint Photographic Expert Group
RGB	Red, green, blue
DCT	Discrete Cosine Transform
RC4	Rivest Cipher 4
DES	Data Encryption Standard
AES	Advanced Encryption Standard
3DES	Triple Data Encryption Standard
RSA	Rivest, Shamir, Adleman
GUI	Graphical User Interface
GUIDE	Graphical User Interface Développent Environnent

Introduction général

Le trafic des images numériques augmente rapidement sur les réseaux. La protection des données numériques, et en particulier les images, devient importante pour de nombreuses raisons telles que la confidentialité et l'intégrité. Actuellement, la façon la plus répandue de répondre au problème de la confidentialité est le cryptage. Dans notre travail, nous présentons une technique de sécurité pour les images confidentielles qui est la combinaison de deux techniques, la première est la compression d'image qui est basée sur la transformation en ondelettes, la deuxième est la cryptographie basée sur une clé symétrique, qu'utilise deux opérations logiques, l'une est «XOR» et la seconde est une opération de «décalage circulaire» .

Le but de la compression est de minimiser voire supprimer la redondance de l'information dans une image, et donc réduire la taille en octets d'un fichier graphique sans dégrader la qualité de l'image à un niveau inacceptable. Cela diminuer également le temps nécessaire pour que les images soient cryptées et transmise sur le réseau.

Les techniques de compression de données peuvent être divisées en deux grandes familles: la première est la compression de données avec perte, qu'est concède une certaine perte de précision en échange d'une compression considérablement accrue. Elle s'avère efficace lorsqu'elle est appliquée aux images graphiques et à la voix numérisée. La deuxième est la compression sans perte, qui consiste en ces techniques garantissant de générer une copie exacte du flux de données d'entrée après un cycle de compression. Elle s'agit utilisé lors du stockage des enregistrements de base de données, des feuilles de calcul ou des fichiers de traitement de texte.

Ce mémoire est divisé en trois chapitres selon l'organisation suivante :

Le premier chapitre: dans ce chapitre nous donnons une définition simple de l'image numérique et de certaines de ses propriétés, ainsi que le concept de compression et qu'il existe deux grandes familles pour classer les différents algorithmes de compression de données (compression de données avec et sans perte). Nous expliquons certains des algorithmes de codage les plus populaires pour chaque famille : le codage de Shannon-Fano, Huffman, Run length encoding (RLE) et Lempel-Ziv-Welch (LZW) pour les algorithmes de compression sans perte, alors que nous avons expliqué deux types de méthode de compression avec perte la compression JPEG et la transformation en ondelettes.

Le deuxième chapitre: Ce chapitre est entièrement consacré aux différentes méthodes de chiffrement. Le début, comme d'habitude, a été de définir le cryptage et de donner des concepts à divers termes liés à cette science. Ensuite, les différentes méthodes de chiffrement sont divisées en plusieurs classifications: selon la clé (le cryptage symétrique, asymétrique et hybride), selon la technique (chiffrement par blocs et par flots), selon le domaine de cryptage (cryptage d'images dans le domaine spatial et fréquentiel) et selon le pourcentage des données cryptées (cryptage total et cryptage partiel ou sélectif). Dans le dernier, nous avons expliqué la cryptographie basée sur une clé symétrique.

Le troisième chapitre: au début, nous avons expliqué les algorithmes de cryptage et de décryptage de la cryptographie basée sur une clé symétrique. Nous avons parlé de l'environnement de travail utilisé pour implémenter notre application, que ce soit l'environnement matériel (les caractéristiques de l'ordinateur utilisé) ou l'environnement logiciel (le langage de programmation Matlab avec la version R2019a). Ensuite, nous avons expliqué les différentes étapes utilisées dans notre application, en commençant par le choix de l'image secrète jusqu'à la décompression, en passant par le processus de compression, de cryptage et de décryptage. Nous avons également fait une traduction de l'interface graphique de notre application de compression et de cryptage d'image, et présenté le rôle de chaque bouton et axes ...etc., en plus d'afficher les résultats obtenus.

Chapitre I: Introduction à la compression

Chapitre I

Introduction à la compression

I.1. Introduction :

Le développement rapide de l'internet et de communication a grandement facilité la vie des gens et a également conduit à une augmentation explosive de la quantité de données transmises par les réseaux, et les images sont aujourd'hui des documents très importants.

Avec l'émergence des photographies numériques et d'autres images couleur complexes, qui peuvent générer des fichiers de très grande taille, des problèmes d'espace de stockage et la nécessité de transmettre rapidement des données d'image sur les réseaux et sur Internet sont survenus avec elle.

Qu'ont donc conduit au développement d'une gamme de techniques de compression d'images, afin de réduire la taille physique des fichiers sans dégrader la qualité de l'image à un niveau inacceptable.

I.2. Généra1ttés et Notions de Base sur l'image

I.2.1. Définition de l'image numérique:

Une image numérique ou fixe est une représentation binaire d'informations visuelles, telles que des dessins, des images, des graphiques, des logos ou des images vidéo individuelles. Les images numériques peuvent être enregistrées électroniquement sur n'importe quel périphérique de stockage [**1**].

I.2.2. Caractéristiques d'une image numérique

L'image numérique est un ensemble structuré d'informations caractérisé par les paramètres suivants:

Pixel: est l'unité de base d'une image. Il est un point carré et l'ensemble de ces points constitue l'image.

Dimension: C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels) [2]. Pour calculer la taille d'une image numérique, il suffit de multiplier le nombre de pixels sur la hauteur par le nombre de pixels sur la largeur de l'image.

Résolution: c'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution [2], la résolution d'une image numérique s'exprime en« ppp » (pixels par pouce) ou en anglais « ppi » (pixels per inch), et la résolution d'impression d'une imprimante ou de capture d'un scanner s'exprime en « ppp » (points par pouce) ou en anglais « dpi » (dots per inch), exemple : Résolution = 50 pixels / 5" = 10 ppp.

Bruit: Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [2].

Histogramme: L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant [2].

Contours et textures: Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différente [2].

Luminance: C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par [2]:

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

Contraste: C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si *L1* et *L2* sont les degrés de luminosité respectivement de deux zones voisines *A1* et *A2* d'une image, le contraste *C* est défini par le rapport [2]:

$$C = \frac{L1 - L2}{L1 + L2}$$

Images à niveaux de gris: Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [2].

Images en couleurs: Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.) [2].

I.3. Définition et types de compression:

I.3.1. Définition :

La compression consiste à réduire la taille physique de blocs d'informations. Elle est très utile pour plusieurs applications informatiques. Les différents algorithmes de compression sont basés sur 3 critères :

– **Le taux de compression :** c'est le rapport de la taille du fichier initial sur la taille du fichier compressé.

$$\tau c = \frac{\text{Nombre de bits de l'image originale}}{\text{Nombre de bits de l'image compressée}}$$

– **La qualité de compression :** sans ou avec pertes (avec le pourcentage de perte).

– **La vitesse** de compression et de décompression.

Un compresseur utilise un algorithme qui sert à optimiser les données en fonction du type de données à compresser ; un décompresseur est donc nécessaire pour reconstruire les données grâce à l'algorithme dual de celui utilisé pour la compression. La méthode de compression dépend du type de données à compresser car une image ou un fichier audio ne représentent pas le même type de données.

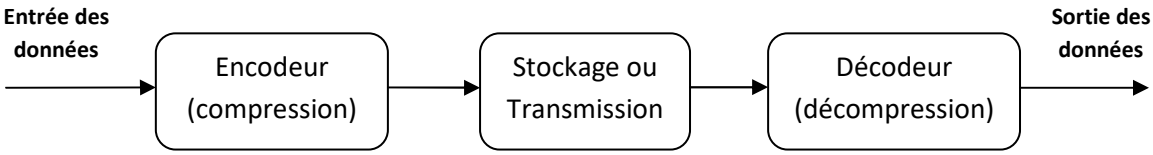


Figure I.1: Un schéma général de compression des données.

I.3.2. types de compression :

I.3.2.1. La compression sans pertes :

En compression sans perte, les données d'origine peuvent être récupérées exactement à partir des données compressées, c'est-à-dire que l'image d'origine et l'image construite sont identiques ou proches d'être identiques.

Dans le contexte de la compression sans pertes, la méthode prend en entrée une série de bits **X** qu'elle transforme en une nouvelle série de bits **Y** plus court que **X**. La série de bits **Y** est

transmise ou stockée pour usage ultérieur. Lorsque l'on veut récupérer les données, On prend **Y** et on applique la méthode de compression inverse [3].

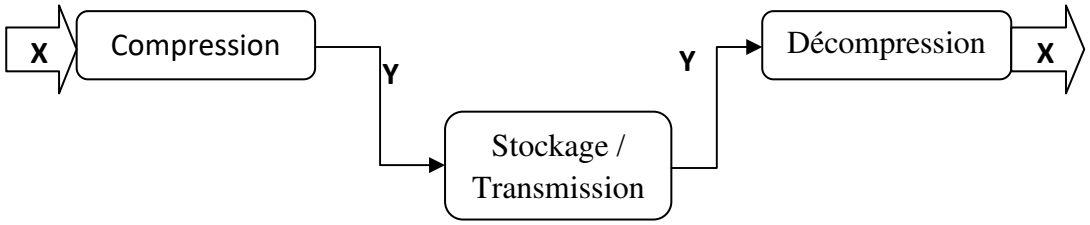


Figure I.2: Compression sans pertes.

I.3.2.1.1. Le codage de Shannon-Fano :

L'algorithme pour produire un code de Shannon-Fano pour un alphabet ∑ est le suivant [4]:

1- Trier les symboles de l'alphabet ∑ en ordre décroissant de fréquence d'apparition (du plus fréquent au moins fréquent).
2- Diviser ∑ en deux sous-ensembles ∑**1** et ∑**2** en respectant les deux contraintes suivantes :
   a) ∑**1** doit contenir les n plus fréquents symboles de ∑ et ∑**2** doit contenir les **|∑| - n** moins fréquents symboles de ∑.
   b) La différence entre la somme des fréquences d'apparition des symboles de l'ensemble ∑**1** et la somme des fréquences d'apparition des symboles de l'ensemble ∑**2** doit être la plus petite possible. Formellement, nous cherchons les ensembles **∑1 = {s1, s2, … , sj-1}** et **∑2 = {sj, sj+1, … , sN}** tels que :

   ou N = |∑| et **p(si)** est la fréquence d'apparition de **si**.

3- Attribuer aux symboles dans ∑**1** le bit **0** comme premier bit de leurs mots de code et attribuer aux symboles dans ∑**2** le bit **1**.
4- Répéter l'algorithme récursivement, à partir de l'étape 2, sur chacun des ensembles jusqu'à ce qu'il ne reste qu'un symbole dans chaque ensemble. A chaque fois qu'un ensemble est divisé, les symboles appartenant à cet ensemble se voient attribuer un bit supplémentaire à leurs mots de code.

Exemple: Pour illustrer cet algorithme, nous allons coder la phrase suivante:

"INTERNATIONAL CONFERENCE ON OPERATOR THEORY "

Symbole	A	C	E	F	H	I	L	N	O	P	R	T	Y	Spa.
Fréquence	3	2	6	1	1	2	1	6	6	1	5	4	1	4

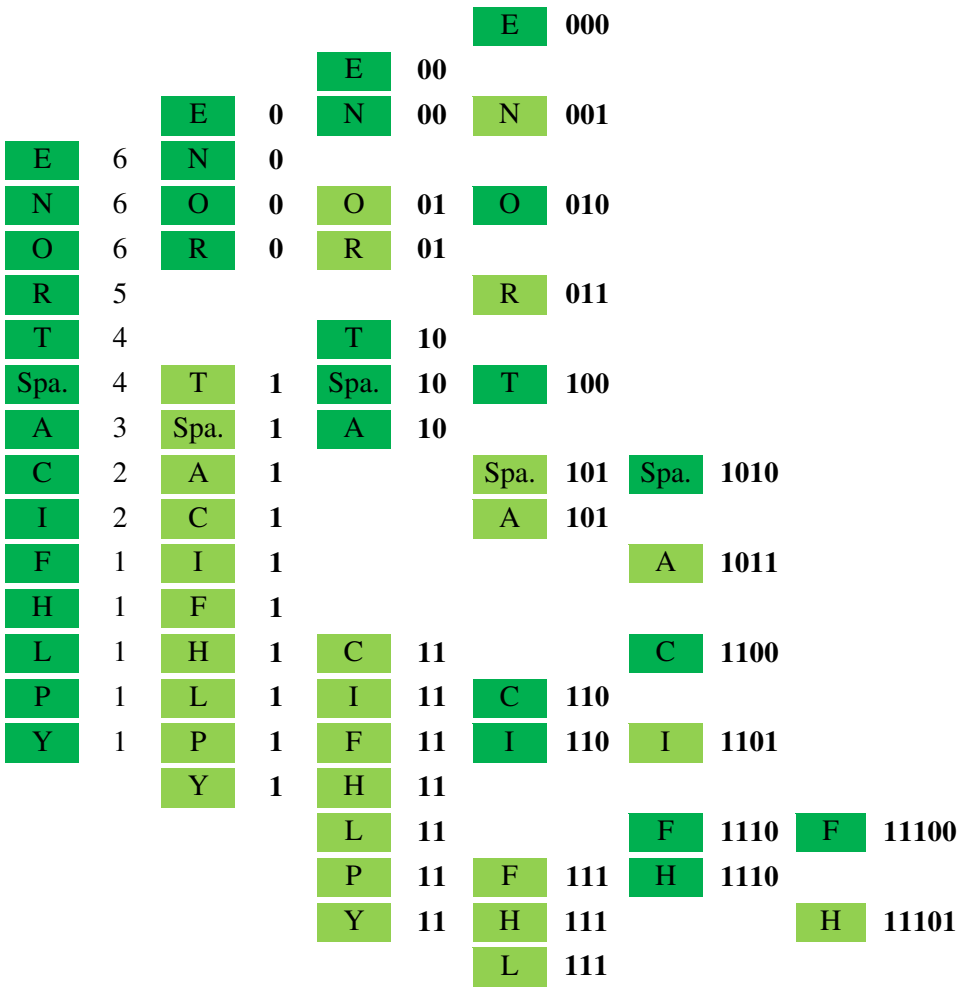
Etape 1 : On commence par le calculer des probabilités à partir des fréquences d'apparition on obtient le tableau suivant:

Symbole	A	C	E	F	H	I	L	N	O	P	R	T	Y	Spa.
Fréquence	3	2	6	1	1	2	1	6	6	1	5	4	1	4
Prob. (%)	7	5	14	2	2	5	2	14	14	2	12	9	2	9

Après le calcul des probabilités on classe les symboles dans un ordre décroissant comme suit:

Symbole	E	N	O	R	T	Spa.	A	C	I	F	H	L	P	Y
Fréquence	6	6	6	5	4	4	3	2	2	1	1	1	1	1

Application successive des étapes 2,3 et 4, nous obtenons le résultat suivant:



P	111	L	1111	L	11110		
Y	111	P	1111				
		Y	1111	P	11111	P	111110
				Y	11111		
						Y	111111

Le résultat de codage est illustrer dans le tableau suivant où l est la longueur de code obtenu.

Symbole	Code	/
A	**1011**	4
C	**1100**	4
E	**000**	3
F	**11100**	5
H	**11101**	5
I	**1101**	4
L	**11110**	5
N	**001**	3
O	**010**	3
P	**111110**	6
R	**011**	3
T	**100**	3
Y	**111111**	6
Spa.	**1010**	4

I.3.2.1.2. Le codage de Huffman :

En informatique et en théorie de l'information, le codage de Huffman est un algorithme de codage entropique utilisé pour la compression de données sans perte [5]. Le codage de Huffman est basé sur la fréquence d'occurrence d'un élément de données. Le principe est d'utiliser un nombre de bits plus faible pour coder les données les plus fréquentes [6]. La base de ce codage est un arbre de code selon Huffman, qui attribue des mots de code courts aux symboles fréquemment utilisés et des mots de code longs aux symboles rarement utilisés. L'algorithme de construction de l'encodage suit cet algorithme chaque symbole est une feuille et une racine. L'organigramme de l'algorithme de Huffman est illustré à la figure I.3 [7].

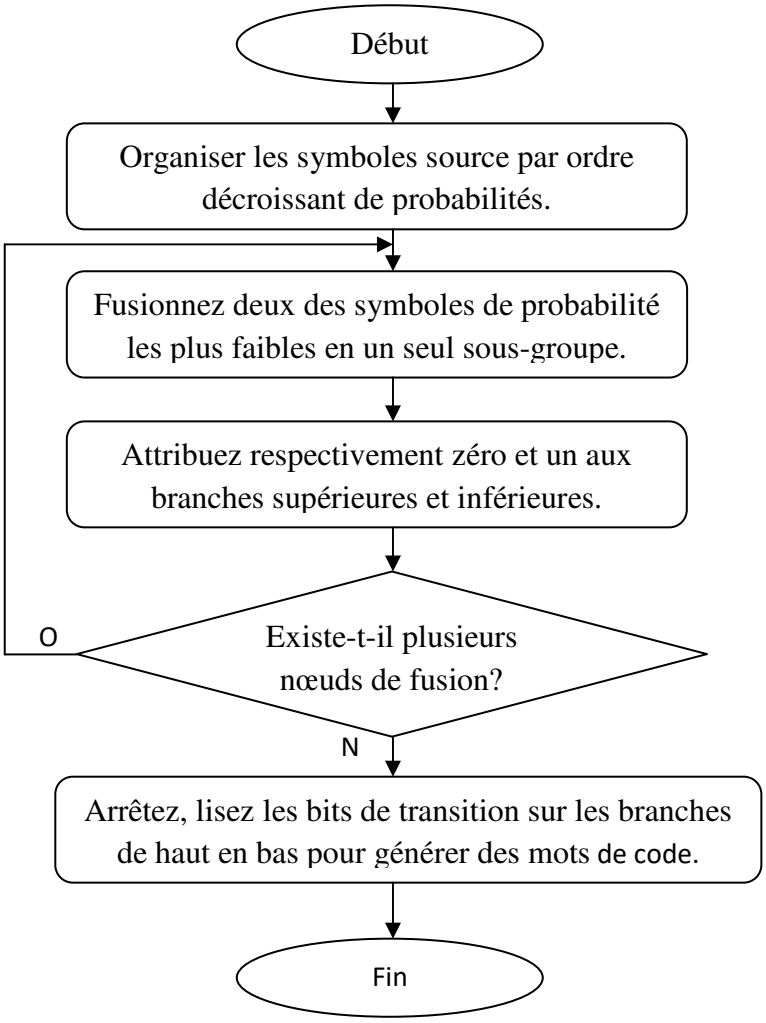
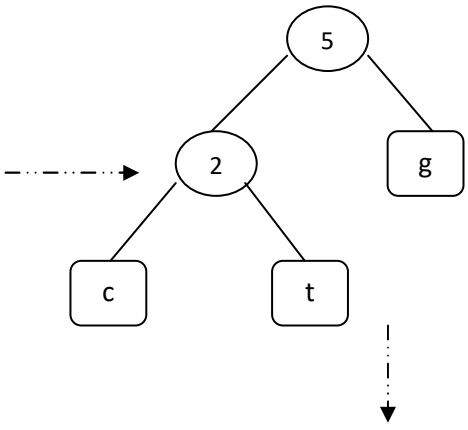


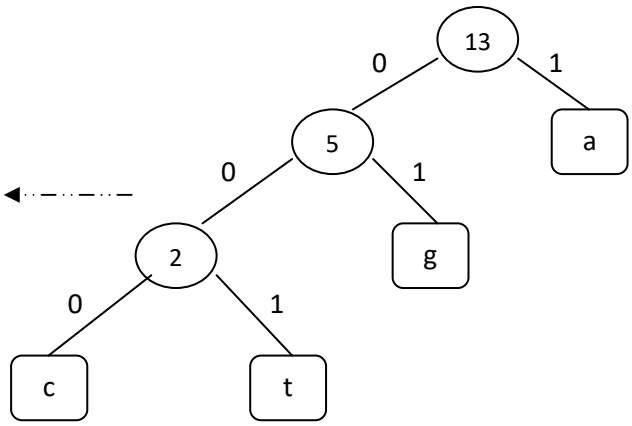
Figure I.3: Organigramme de l'algorithme de Huffman.

Exemple: pour illustrer l'algorithme de Huffman, nous allons coder la phrase suivante: **'cagataaagagaa'**

Symboles	fréquence
c	1
t	1
g	3
a	8



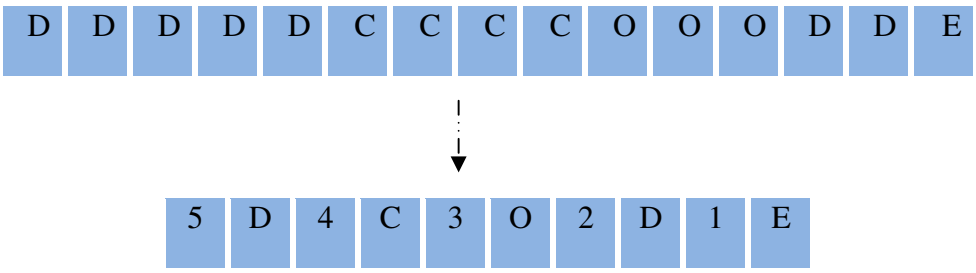
Symboles	code	fréquence
c	000	1
t	001	1
g	01	3
a	1	8



I.3.2.1.3. Le codage de Run length encoding (RLE) :

Run Length Encoding (RLE) est peut-être la technique de compression la plus simple couramment utilisée. Les algorithmes RLE sont sans perte et fonctionnent en recherchant des séries de bits, octets ou pixels de la même valeur et en codant la longueur et la valeur de la série. En tant que tel, RLE obtient les meilleurs résultats avec des images contenant de grandes zones de couleurs contiguës, et en particulier des images monochromes. Les images couleur complexes, telles que les photographies, ne se compressent pas bien - dans certains cas, RLE peut en fait augmenter la taille du fichier [8].

Exemple : pour illustrer l'algorithme RLE, nous allons coder la phrase suivante: **'DDDDDCCCCOOODDE'**



I.3.2.1.4. L'algorithme Lempel-Ziv-Welch (LZW) :

Utilise des techniques adaptatives et basées sur des dictionnaires. Le prédécesseur de LZW est LZ77 et LZ78 développé par Jacob Ziv et Abraham Lempel en 1977 et 1978. Terry Welch a développé la technique en 1984. LZW largement utilisé sous UNIX, GIF pour les modems [9].

La compression LZW est l'une des techniques du dictionnaire adaptatif. Le dictionnaire est créé lors de l'encodage des données. L'encodage peut donc être effectué à la volée. Le dictionnaire n'a pas besoin d'être transmis. Le dictionnaire peut être créé à la réception à la volée. La compression LZW fonctionne mieux lorsqu'elle est appliquée sur des images monochromes et des fichiers texte contenant du texte / des motifs répétitifs [10].

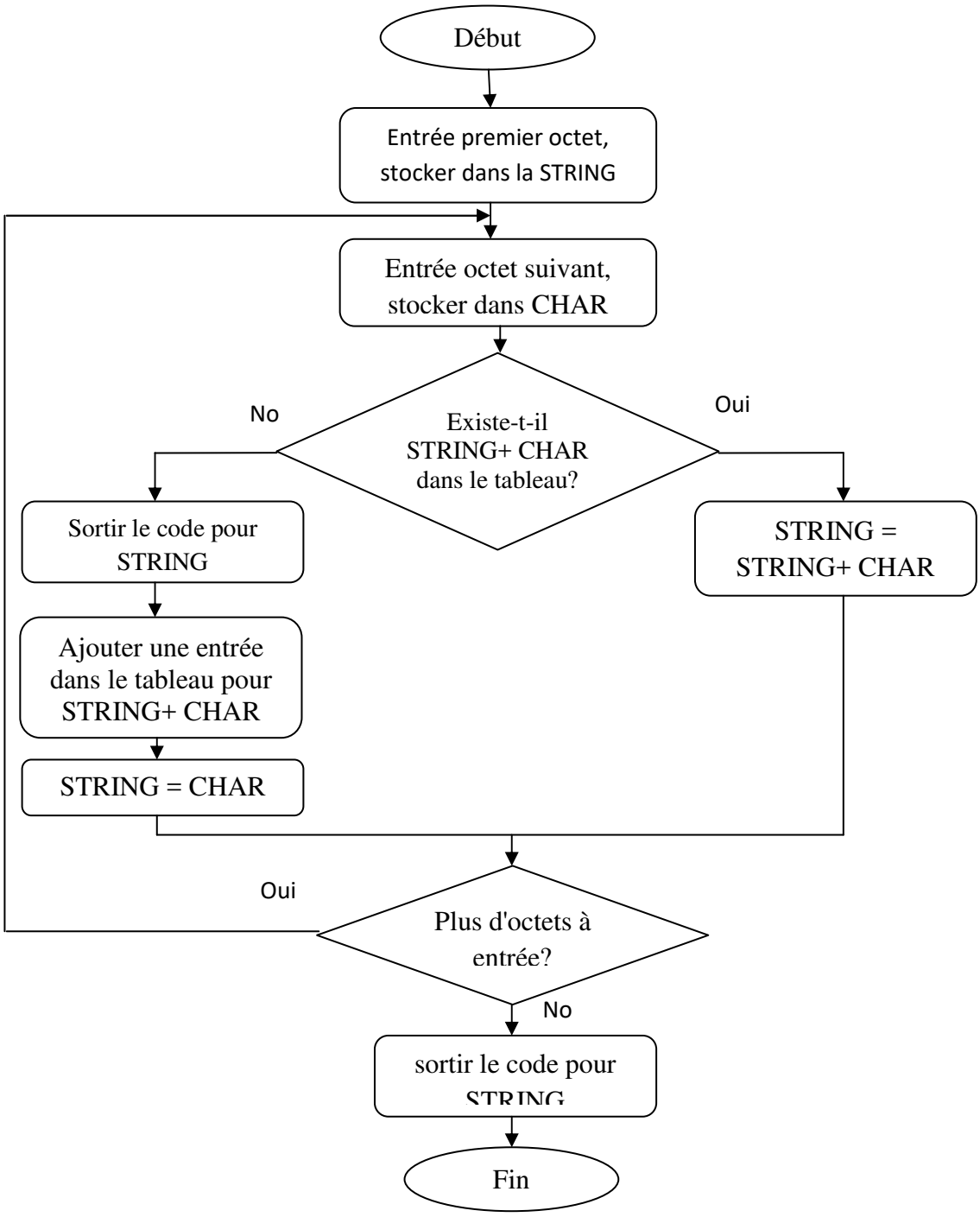


Figure I.4: Organigramme de l'algorithme de Lempel-Ziv-Welch (LZW).

Exemple : pour illustrer l'algorithme LZW, nous allons coder la phrase suivante: '**ananas** '

Fenêtre	Sortie	Ajout dict.	
		phrase	code
a n a n a s			
a n a n a s	Code (a)	an	256
a n a n a s			
a n a n a s	Code (n)	na	257
a n a n a s			
a n a n a s	256	ana	258
a n a n a s			
a n a n a s	Code (a)	as	259
a n a n a s	Code (s)		

I.3.2.2. La compression avec pertes :

Compression avec perte dans laquelle l'image d'origine a perdu certaines informations lorsqu'elle est récupérée à partir d'une image compressée, elle donne un meilleur taux de compression par rapport à la compression sans perte. Ici, on compresse une série de bits **X** pour obtenir la série **Y**, qui est transmise ou stockée. Lorsqu'on décompresse **Y** on retrouve **Z**, qui est potentiellement différent de **X**. Si **Z** diffère de **X**, il faudra alors que **Z≈X**, selon une mesure approprié [11].

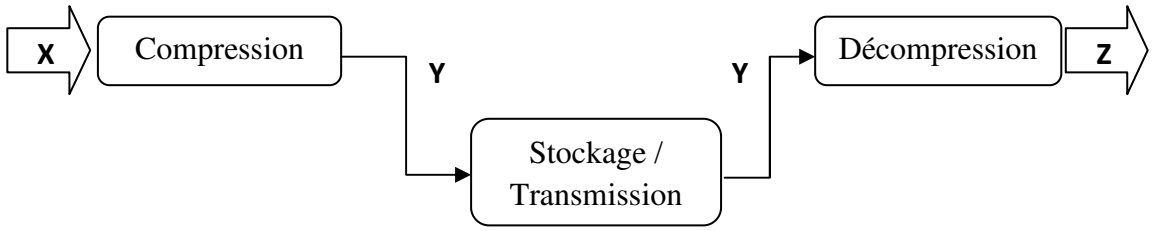


Figure I.5: Compression avec pertes.

I.3.2.2.1. La compression JPEG :

JPEG est l'abréviation de (Joint-Photographic-Expert-Group), qui représente la comite d'experte qui édite des normes de compression pour l'image fixe, et qui a réussi entre l'année 1991 et 1992 de développer un algorithme de compression d'image qui propose deux modes de

compression, le premier avec perte et le second sans perte. Cet algorithme reste jusqu'a nos jours une norme standard pour le codage d'image surtout celles prisent avec une appareille photo [12].

Le principe de cette norme repose sur le principe que l'il humain est moins sensible aux changements des couleurs brusques qu'aux changements de couleurs lentes.

C'est pour cette raison que l'algorithme fait d'abord une transformation de couleur de RGB vers YCbCr.

La compression d'une image au format JPEG suit un certain nombre d'étapes visant à réduire la place occupée à l'aide de diverses méthodes. Ces étapes sont illustrées dans le schéma ci-dessous [13] :

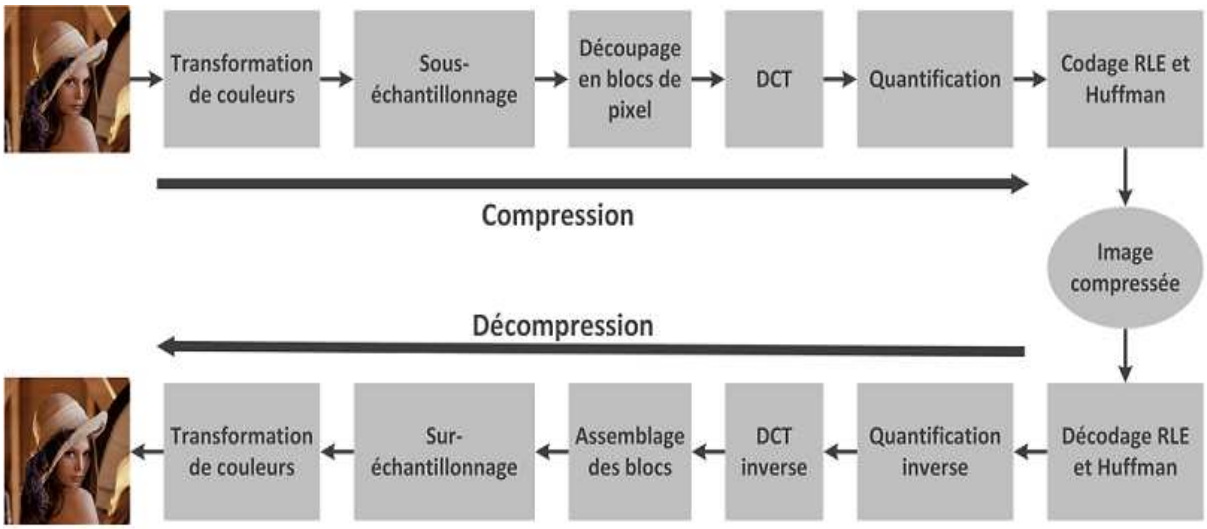


Figure I.6: Les étapes de la Compression & décompression JPEG.

Dans ce qui suit nous présentons les étapes de déroulement de l'algorithme JPEG :

L'étape 1 : transformation de l'image de format RGB vers YCbCr Dans cette phase l'algorithme applique une fonction de transformation pour extraire les couleurs lentes de l'image.

La formule de passage de RGB ou RVB (rouge vert bleu) vers YCbCr est la suivante [14]:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.144 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \times \begin{pmatrix} R \\ V \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

La formule inverse :

$$\begin{pmatrix} R \\ V \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.402 \\ 1 & -0.344 & -0.714 \\ 1 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} Y \\ (Cb-128) \\ (Cr-128) \end{pmatrix}$$

Ou :

- *Y* est la luminance.
- les deux chrominances *Cb, Cr.*

L'étape 2 : pour faciliter les calculs, une division pour chaque bloc *(Y, Cb, Cr)* doit être appliquée pour chaque sous-bloc *Y, Cr, Cb*, chaque bloc contient 8*8 case figure I.7.

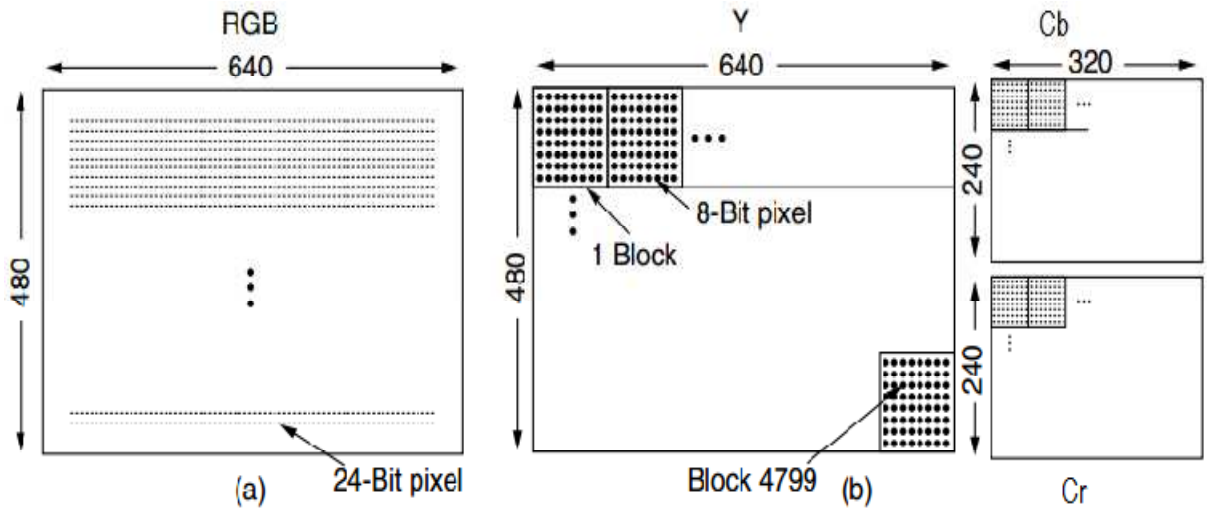


Figure I.7: (a) Données d'entrée RGB 640 × 480. (b) Après la préparation du bloc [15].

L'étape 3 : la transformée DCT (Discrete Cosine Transform), en français transformée en cosinus discrète), est une transformation numérique qui est appliquée séparément à chaque bloc de 8 × 8 pixels de chaque composante (YCbCr) ou (YUV) selon les équations 3 et 4. La DCT converti des blocs d'échantillons en blocs de coefficients de fréquences, elle est de même nature que la transformée de Fourier.

La transformée DCT s'exprime par [15]:

$$F(u,v) = \frac{1}{4} C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}\right]$$

Et la transformée DCT inverse s'exprime par [15]:

$$f(x,y) = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v) * \cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right]$$

Ou : $\boldsymbol{C(u), C(v) = \frac{1}{\sqrt{2}}}$ $\boldsymbol{pour}\ u, v = 0;$

$\boldsymbol{C(u), C(v) = 1}$ $\boldsymbol{autrement}$

$\boldsymbol{F(u,v)}$**:** représente la valeur de la DCT au point de coordonnées *(u, v)* dans le bloc résultat de 8×8 pixels.

$\boldsymbol{f(x,y)}$**:** représente la valeur du pixel de coordonnées *(i, j)* dans le bloc de l'image originale de 8×8 pixels.

L'étape 4 : la quantification s'applique à chaque bloc obtenu à partir du DCT. La quantification consiste à diviser le bloc de 8*8 case par une matrice de quantification déjà choisis par le codeur, le but est d'atténuer les hautes fréquences auquel l'œil est très peu sensible. C'est à cette phase qu'on perd l'information d'origine, car la quantification est une fonction à sens unique (on ne peut pas revenir en arrière) [16].

Généralement la matrice choisis par le codeur est de cette forme la matrice choisis par le codeur est de cette forme [17]:

$$\begin{pmatrix} A & B & B & C & C & D & D & D \\ B & B & C & C & D & D & D & D \\ B & C & C & D & D & D & D & D \\ C & C & D & D & D & D & D & D \\ C & D & D & D & D & D & D & D \\ D & D & D & D & D & D & D & D \\ D & D & D & D & D & D & D & D \\ D & D & D & D & D & D & D & D \end{pmatrix}$$

Figure I.8: modèle de matrice de quantification.

Ou : *A, B, D* sont en ordre croissant, avec ce genre de matrice on veut préserver les pixels qui se situent dans le coté haut à gauche.

- La formule de la quantification est la suivante:

$$F'(u,v) = ent\,[F(u,v)/q(u,v)$$

- La formule inverse (dequantifacation):

$$F''(u,v) = F'(u,v) * q(u,v)$$

Avec :

$\boldsymbol{F'(u,v)}$ est la matrice quantifiée, $\boldsymbol{F(u,v)}$ est la matrice des coefficients à quantifier, $\boldsymbol{q(u,v)}$ est la table de quantification et $\boldsymbol{F''(u,v)}$ représente la matrice dequantifiée.

L'étape 5 : linéarité les 64 éléments et applique un codage de longueur d'exécution à la liste. La numérisation du bloc de gauche à droite, puis de haut en bas ne concentrera pas les zéros ensemble à la fin de la séquence, donc un motif de numérisation en Zig-Zag (balayage depuis les basses fréquences jusqu'aux hautes fréquences) est utilisé comme indiqué sur la figure I.9.

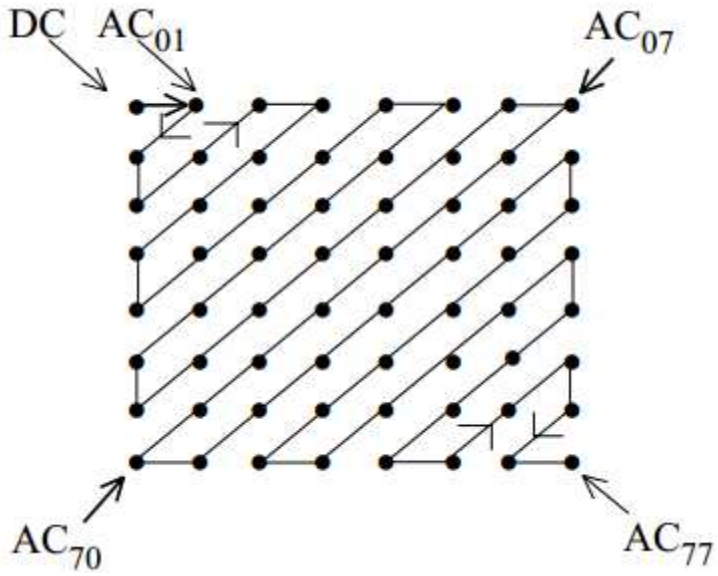


Figure I.9: numérisation en Séquence Zig-Zag [15].

D'une part, les suites de valeurs nulles sont simplement codées en donnant le nombre de 0 successifs. D'autre part, les valeurs non nulles seront codées en utilisant RLE, et Huffman consécutivement.

Le schéma général de cet algorithme est illustré dans la figure I.10 [18]:

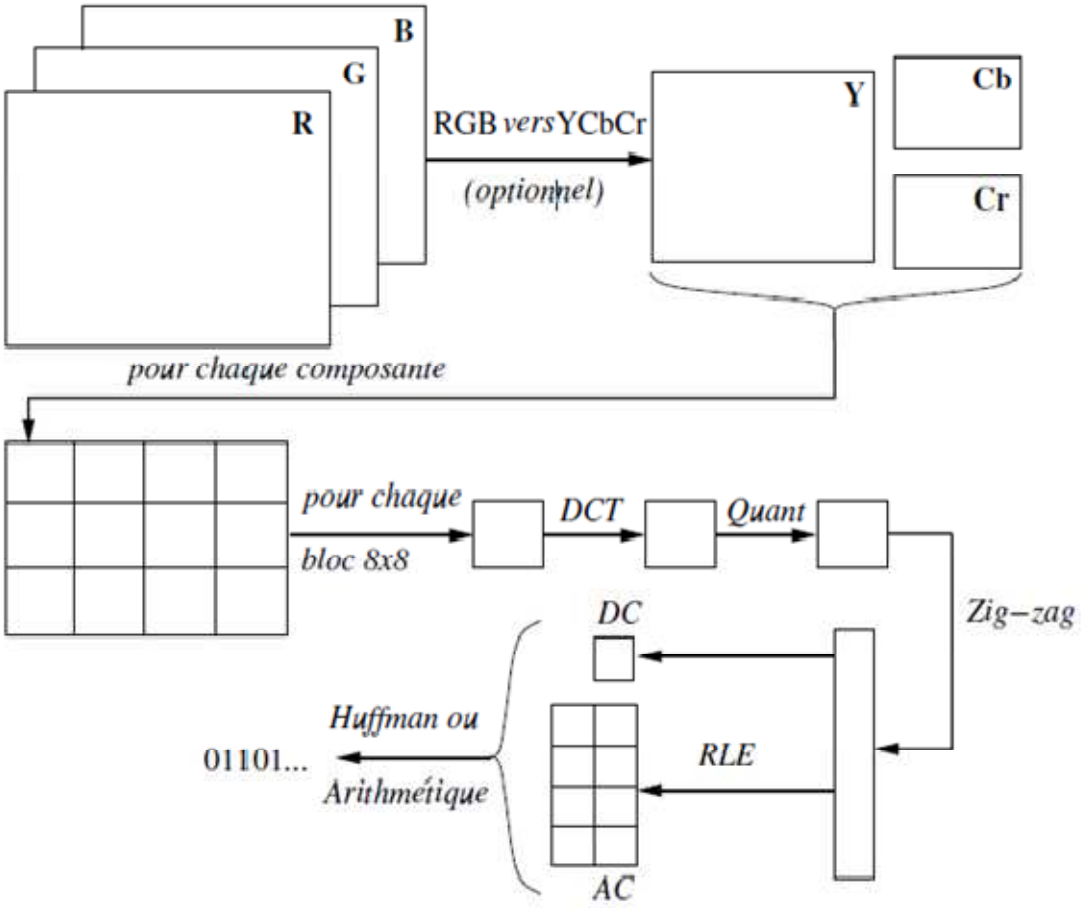


Figure I.10: le déroulement de l'algorithme JPEG

Exemple :

Compression :

Matrice de pixels d'entée

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147

148	155	136	155	152	147	147	136

Matrice DCT

1210	-18	15	-9	23	-9	-14	-19
21	-34	26	-9	-11	11	14	7
-10	-24	-2	6	-18	3	-20	-1
-8	-5	14	-15	-8	-3	-3	8
-3	10	8	1	-11	18	18	15
4	-2	-18	8	8	-4	1	-7
9	1	-3	4	-1	-7	-1	-2
0	-8	-2	2	1	4	-6	0

Matrice de quantification

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Matrice DCT quantifiée

403	-4	2	-1	2	-1	-1	-1
4	-5	3	-1	-1	1	1	0
-1	-3	0	0	-1	0	-1	0
-1	0	1	-1	0	0	0	0
0	1	1	0	-1	1	1	1
0	0	-1	0	0	0	0	0
1	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0

Le résultat de linéarisation de la matrice des fréquences quantifiées est :

403, -4, 4, -1, -5, 2, -1, 3, -3, -1, 0, 0, 0, -1, 2, -1, -1, 0, 1, 1, 0, 1, 0, 1, -1, -1, 1,-1, -1, 1, 0, 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0.

Le codage se fait en donnant les suites de valeurs nulles le nombre de 0 successifs, les valeurs non nulles en utilisant RLE, et Huffman consécutivement.

Décompression :

Matrice DCT déquantifiée

1209	-20	14	-9	22	-13	-15	-17
20	-35	27	-11	-13	15	17	0
-7	-27	0	0	-15	0	-19	0
-9	0	13	-15	0	0	0	0
0	13	15	0	-19	21	23	25
0	0	-17	0	0	0	0	0
15	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matrice de pixels de sortie

142	143	154	141	133	153	179	179
139	152	129	151	144	154	163	181
150	156	139	166	162	163	154	172
163	145	160	153	151	153	145	154
168	150	156	145	140	139	141	159
148	164	133	164	158	140	136	163
130	159	123	164	165	140	134	145
148	156	140	148	159	146	153	141

I.3.2.2.2. La transformée en ondelettes :

Les ondelettes c'est d'abord une théorie mathématique récente d'analyse du signal développée dans les années 80. On peut considérer qu'il s'agit d'une extension de l'analyse de Fourier.

La transformée en ondelettes consiste à décomposer un signal sur une base de fonctions particulières$\psi_{a,b}$, appelées ondelettes. Ces ondelettes se déduisent d'une fonction générique Ψ par de simples translations temporelles et par des changements d'échelles (dilatation/ contraction), Dans le cas monodimensionnel, la fonction s'écrit :

$$\psi_{a,b} = |a|^{-1/2}\,\Psi\left(\frac{t-b}{a}\right)$$

Où l'indice a détermine un facteur d'échelle et l'indice b est un facteur de translation (position).

Cette fonction génériqueΨ, appelée "ondelette mère", est à la fois bien localisée en temps et en fréquence. C'est une simple fonction oscillante sur un intervalle et nulle en dehors. Il en existe une infinité.

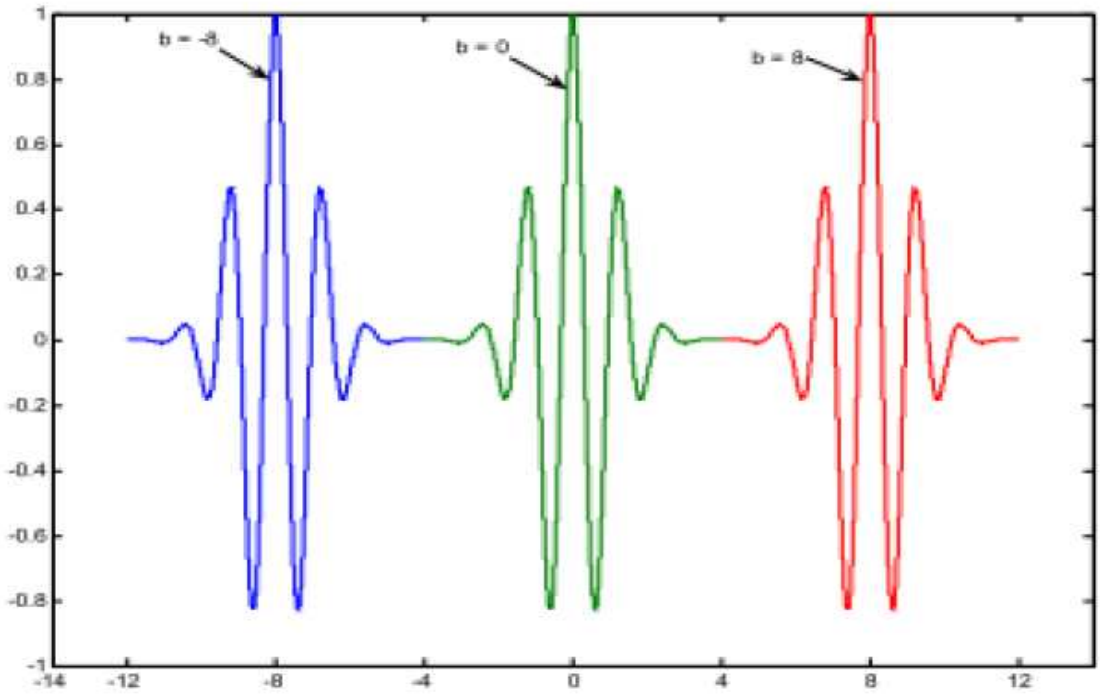
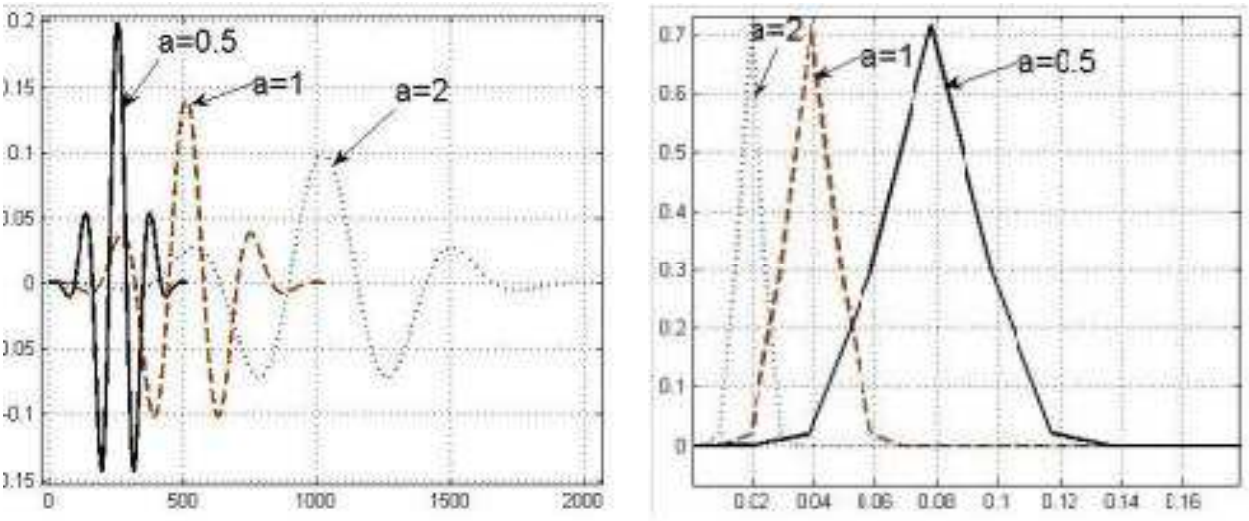


Figure I.11: propriétés de translation d'une ondelette (a est constante) [19]



(a) **(b)**

Figure I.12: propriétés de l'ondelette mère ; contractée et dilatée [20]

Nous constatons que si l'ondelette est dilatée dans le temps elle a un spectre plus concentré autour de sa fréquence centrale. Le contraire est constaté lorsque l'ondelette est contractée. Dans la figure I.12-(a) présente les propriétés temporelles, et La figure I.12-(b) présente les propriétés fréquentielles de 3 échelles différentes d'une ondelette [21].

- *a= 0.5* ondelette contractée.
- *a= 1* ondelette mère.
- *a= 2* ondelette dilatée.

En respectant l'inégalité de Heisenberg, en gardant la surface du rectangle constante, lors de l'analyse des composantes hautes fréquences (petite échelle) la transformée en ondelettes favorise la résolution temporelle, et la résolution fréquentielle lors de l'analyse des composantes basses fréquences (grand échelle) figure I.13.

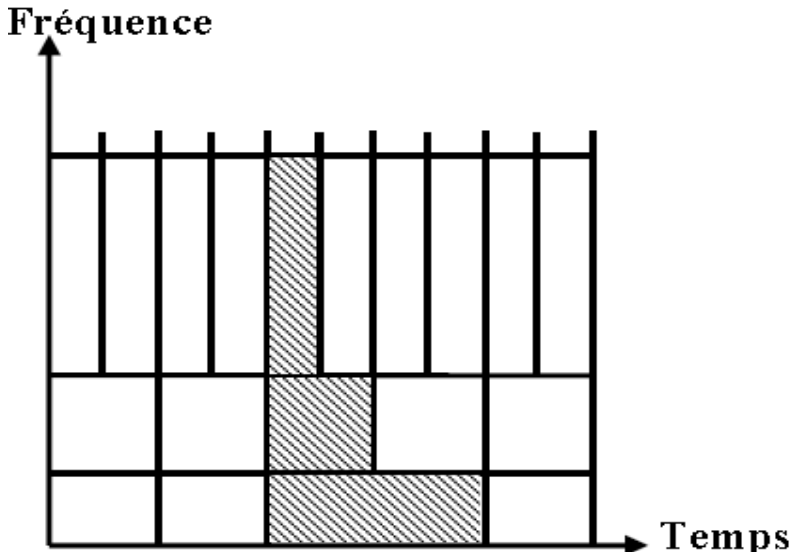


Figure I.13: Plans Temps-Fréquence de la transformée en ondelettes [22]

On distingue principalement deux types de transformée en ondelettes :

I.3.2.2.2.1. Transformée en Ondelettes Continue (CWT):

La transformée en ondelette continue utilise des translations et des dilatations de la fonction ondelette mère durant tout l'intervalle temporel de manière continue [23].

Elle est définie par l'équation suivant:

$$CWT_x(a,b) = \int_{-\infty}^{+\infty} x(t).\psi_b^a(t)dt$$

Le scalogramme, défini par le carré du module de la transformée en ondelettes continues, est :

$$SC_x(b) = |CWT_x(a,b)|^2$$

La transformée en ondelettes inversée est:

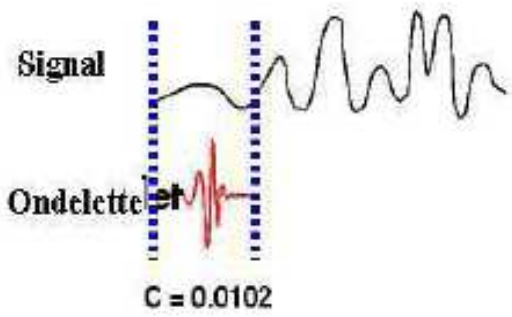
$$x(t) = \frac{1}{c_\psi}\int_{a=0}^{+\infty}\int_{b=-\infty}^{+\infty}\frac{CWT_x(a,b)}{a^2}\psi_a^b dadb$$

$\psi\left(\frac{t-b}{a}\right)$ Représente l'ondelette fille et c_ψ Constant d'admissibilité (condition).

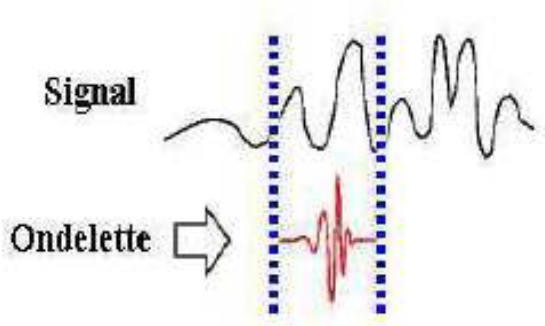
La Transformée en Ondelette Continue se déroule selon les étapes suivantes [24]:

1- Prendre une ondelette et la comparer à une section au début du signal original.

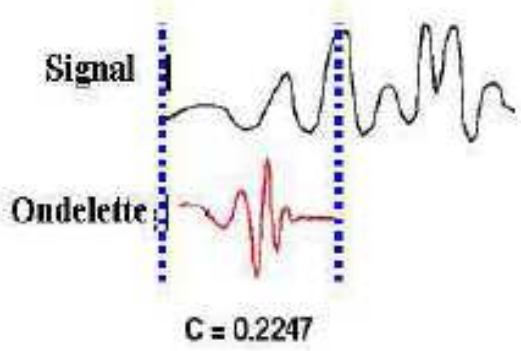
2- Calculer le coefficient *C (s,τ)*. Par exemple:



3- Translater l'ondelette à droite et répéter l'étape 1 et 2 sur tout le signal. Par example:



4- Dilater ou compresser l'ondelette et répéter les étapes 1 à 3. Par example:



5- Répéter les étapes 1 à 4 pour toutes les échelles choisies.

I.3.2.2.2.2. Transformée en ondelettes discrète (DWT):

La transformée en ondelettes discrète (TOD) est produite pour surmonter le problème de la TOC. En effet il est évident que la TOC ne peut être manuellement calculée en utilisant les équations analytiques, le calcul intégral, etc, c'est donc aux ordinateurs de calculer cette transformées. Il devient donc nécessaire de discrétiser les transformées. La TOD, contrairement à la TOC, fournit suffisamment d'information, tant pour l'analyse que pour la reconstruction du

signal original, en un temps de calcul notablement réduit. La TOD translate et dilate l'ondelette selon des valeurs discrètes des facteurs d'échelle et de translation (la fonction ondelettes et les facteurs de dilatation et de translation sont discrets) telles que:

$$\boldsymbol{a} = \boldsymbol{a_0^j}$$

$$\boldsymbol{d} = \boldsymbol{k d_0 a_0^j}$$

Où : $a_0 > 1; d_0 > 0; k, j \in Z$.

Les coefficients de la TOD sont définis par [25] :

$$\boldsymbol{D_x(a_0, b_0)} = \int_{-\infty}^{+\infty} \boldsymbol{x(t).\psi_{b_0}^{a_0}(t)dt}$$

Avec

$$\boldsymbol{\psi_{b_0}^{a_0}} = \boldsymbol{|a_0^j|^{\frac{-1}{2}}\psi\left(\frac{k d_0 a_0^j}{a_0^j}\right)}$$

Dans la majorité des cas, on utilise a_0=2 *et* b_0=2 l'on obtient la transformée en ondelette discrète dyadique. Le signal original (t) peut être reconstruit à partir des coefficients obtenus par la transformée en ondelette discrète et il est donné par l'équation suivante:

$$\boldsymbol{x(t)} = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} \boldsymbol{D_x(a_0, b_0).|a_0^j|^{\frac{-1}{2}}\psi\left(\frac{k d_0 a_0^j}{a_0^j}\right)}$$

Dans la transformation en ondelette discrète, on parle souvent d'approximation et de détail.

L'approximation correspond à la haute échelle, c'est à dire aux composantes de basse fréquence du signal. Les détails sont à basses échelles c'est les composantes de hautes fréquences. Notons qu'approximation et détail émergent comme deux signaux lorsque le signal original traverse deux filtres complémentaires.

I.3.2.2.2.3. Transformée en Ondelette à deux Dimensions :

L'extension à deux dimensions de la TOD (TOD-2D) est essentielle pour la transformation des signaux à deux dimensions (2D), comme une image numérique, qui est effectué par application itérative du DWT unidimensionnel. En considérant l'image comme une matrice de données composée de lignes et de colonnes de vecteurs de signaux, un DWT 2D à un seul niveau doit être effectué sur l'image dans les deux étapes suivantes dans l'ordre suivant [26]:

a) Le DWT 1-d doit être effectué sur chaque ligne d'image, produisant un tableau de données intermédiaire filtré horizontalement passe-bas et un filtre passe-haut horizontal, chaque moitié aussi large que le réseau d'images d'origine, comme illustré sur la figure I.14 (b);
b) Le DWT 1-d doit être appliqué à chaque colonne des deux tableaux de données intermédiaires pour produire quatre sous-bandes, comme le montre la figure I.14 (c).

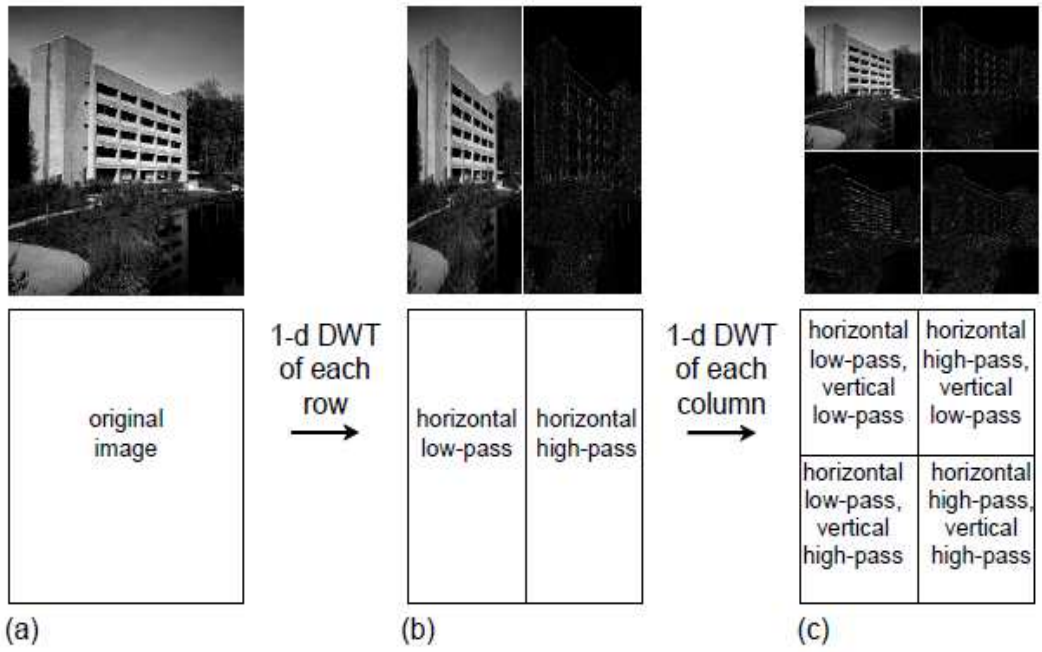


Figure I.14 : DWT 2D (niveau un) [26]

Chacune des quatre matrices de données de sous-bande obtenues est moitié moins large et moitié plus haute que la matrice d'image d'origine. Dans les illustrations, ces sous-bandes sont souvent représentées disposées en un seul réseau qui a la même taille que le réseau d'images d'origine (voir figure I.14 (c)). En commençant en haut à gauche et en procédant dans le sens des aiguilles d'une montre sur la figure I.14 (c), La TOD-2D fournit à chaque échelle les quatre sous-images suivantes:

- Une image de basse résolution : LL.
- Une image de détails verticaux : HL.
- Une image de détails diagonaux : HH.
- Une image de détails horizontaux : LH.

Pour augmenter l'efficacité de la compression, la corrélation restant dans la sous-bande LL après la décomposition DWT 2D est exploitée en appliquant d'autres niveaux

DWT pour produire un DWT 2D à plusieurs niveaux. Cette norme recommandée spécifie trois niveaux de décomposition (voir figure I.15).

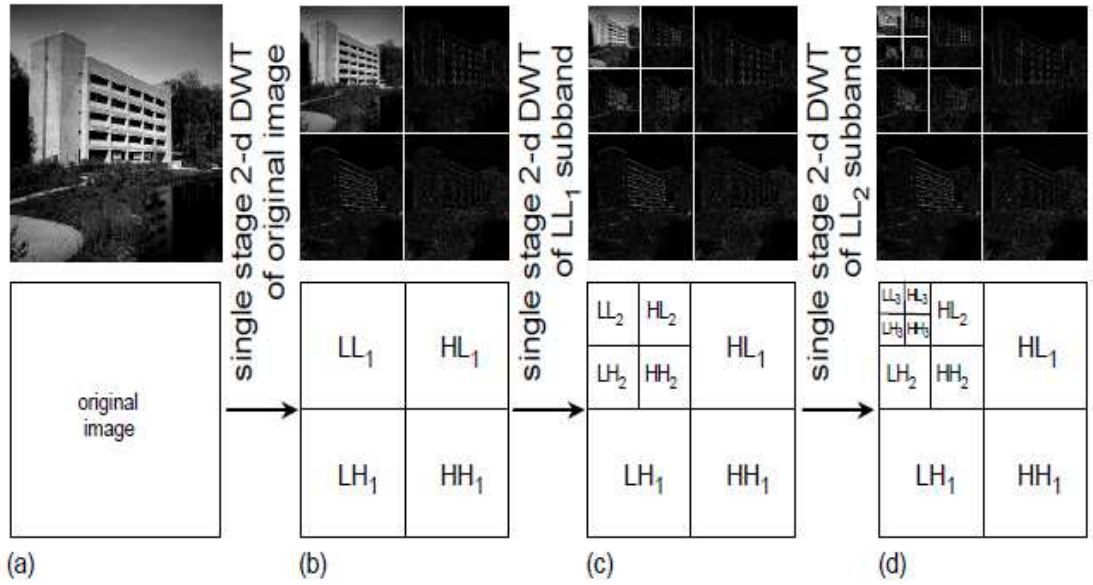


Figure I.15: DWT 2D (niveaux trois) [26]

On donne maintenant une idée de la méthode de compression qui utilise les ondelettes. Elle est résumée par ce schéma (figure I.16) [27]:

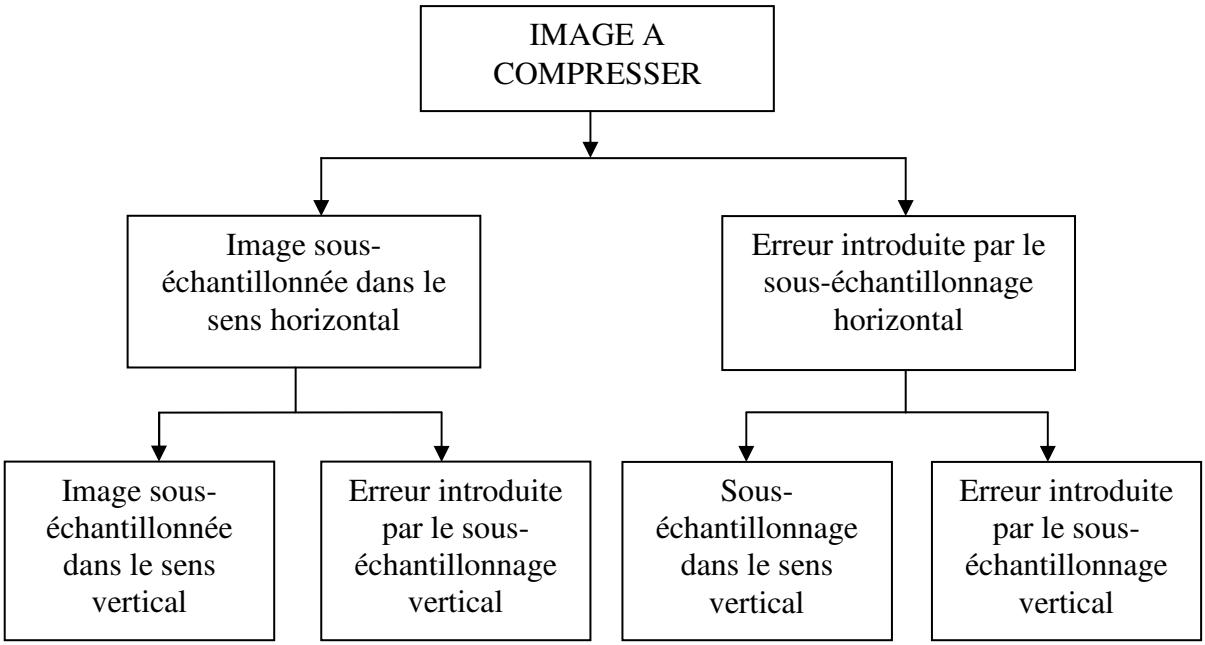


Figure I.16: La méthode de compression qu'utilise les ondelettes.

I.3.2.2.2.4. Algorithmes Pyramidal de Burt & Adelson [28]:

- On fait un sous-échantillonnage de l'image dans le sens horizontal.
- On calcule l'erreur entre l'image originale et l'image sous-échantillonnée dans le sens horizontal.
- Pour chacune des 2 images obtenues, on fait un sous-échantillonnage dans le sens vertical.
- Pour chacune des 2 images obtenues, on calcule l'erreur dans le sens vertical.
- On obtient une image dont la résolution est divisée par 2 et 3 images qui codent les erreurs entre l'image originale et l'image sous-échantillonnée
- On répète cette transformation un certain nombre de fois puis on effectue une quantification.
- On abandonne les détails inférieurs à un certain niveau et on code les valeurs restantes.

I.4- Conclusion :

La compression d'image est une partie extrêmement importante de l'informatique moderne. En ayant la possibilité de compresser des images à une fraction de leur taille d'origine, un espace disque précieux et coûteux peut être économisé. En outre, le transport d'images d'un ordinateur à

un autre devient plus facile et (c'est pourquoi la compression d'image a joué un rôle important dans le développement de l'internet).

En fin, on peut en conclure qu'il existait de nombreuses manières de compresser des données. On en a déduit qu'elles peuvent être classées en deux grandes catégories d'après leurs propriétés, la compression sans pertes d'informations qui permet de retrouver exactement toute l'information contenue dans les données originale après la décompression, ainsi que la compression avec pertes d'informations qui comporte une perte de données pendant le processus.

Afin d'arriver à une meilleure qualité des images des méthodes et des mesures sont introduite pour favoriser des méthodes de compression par rapport aux autres selon le domaine d'utilisation de ces dernières. Le deuxième chapitre sera un survol sur le sujet de la cryptographie.

Chapitre II :

Introduction à la cryptographie

Chapitre II

Introduction à la cryptographie

II.1. Introduction :

L'homme connaît le cryptage depuis des temps très reculés, où avait utilisé diverses méthodes et techniques pour envoyer un message secrètement. Ce sont des méthodes qui transforment le message en clair en message incompréhensible ou qui cachent le message par une image, un texte ou autres choses sans qu'une personne étrangère puisse s'en apercevoir.

Avec le développement des réseaux de communication comme internet, les réseaux de la téléphonie mobile et la télévision à péage. Partant du principe que ces réseaux ne sont pas sûres et que l'information peut être interceptée au cours de sa transmission, divers types de cryptage ont évolué pour assurer la confidentialité et l'authenticité des informations échangées.

II.2. Cryptographie :

La cryptographie est un terme composé des deux mots grecs anciens «Kruptos » qui signifie « cacher » et « graphein » qui signifie « écrire ». Ce qui signifie, la cryptographie concerne de rendre des données de sa forme originale (texte clair) à une forme inintelligible à autre que qui-de-droit, en d'autres termes texte chiffré ainsi que la conversion inverse. Le texte se forme originale ou clair peut être une suite de bits, un fichier de texte, un enregistrement de voix numérisé, ou une image numérique.

II.3. Terminologie :

- **Cryptologie :** Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse [29].

  Cryptologie = Cryptographie + Cryptanalyse

- **Cryptographie :** La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné [29].
- **Cryptanalyse :** Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés [29].

- **Crypto-système :** Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné [29].

  L'algorithme est en réalité un triplet d'algorithmes :

  - ✓ L'un générant les clés K,
  - ✓ Un autre pour chiffrer M, et
  - ✓ Un troisième pour déchiffrer C.
- **Chiffrement et déchiffrement :** Le chiffrement consiste à transformer une donnée afin de la rendre incompréhensible par une personne autre que celle autorisée. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement [29].
- **Texte clair :** Texte original intelligible tel qu'il se présentait avant tout chiffrement [29].
- **Texte chiffré (cryptogramme) :** le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair [29].
- **La clef :** Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations [29].
- **Stéganographie :** La stéganographie (du grec steganos, couvert et graphein, écriture) est l'art de cacher un message secret au sein d'un autre message porteur (texte, image, son, vidéo...) de caractère anodin, de sorte que l'existence même du secret en soit dissimulée. Alors qu'avec la cryptographie, la sécurité repose sur le fait que le message chiffré soit incompréhensible pour les personnes non autorisées, avec la stéganographie, la sécurité repose sur le fait que la présence même d'un message secret ne sera sans doute pas soupçonnée et détectée [30].
- **Fonctions de hachage :** Lors d'échanges de messages cryptés, il est important de pouvoir s'assurer que le message n'a pas été altéré ou modifié par un tiers pendant l'envoi. Les fonctions de hachage permettent alors de s'assurer de l'intégrité du message.
  - ✓ Une fonction de hachage h est une fonction qui, à partir d'un document x (fichier) de taille quelconque, calcule une chaine de bits h(x) d'une taille fixée (m) nommée empreinte (ou haché, ou condensé, ou encore résumé) [30].
- **La signature numérique :** est définit comme des données ajoutées à un message ou une transformation cryptographique d'un message permettant à un destinataire de :
  - ✓ Authentifier l'auteur d'un document électronique.

- ✓ Garantie son intégrité.
- ✓ Protéger contre la contrefaçon (seul l'expéditeur doit être capable de générer la signature) -> non-répudiation. La signature électronique est basée sur l'utilisation conjointe d'une fonction de hachage et de la cryptographie asymétrique [31].

- **Certificat électronique :** Un certificat numérique est un bloc de données contenant, dans un format spécifié, les parties suivantes :
  - ✓ La clé publique d'une paire de clés asymétriques,
  - ✓ des informations identifiant le porteur de cette paire de clés (qui peut être une personne ou un équipement), telles que son nom, son adresse IP, son adresse de messagerie électronique, son URL, son titre, son numéro de téléphone, etc.
  - ✓ l'identité de l'entité ou de la personne qui a délivré ce certificat (autorité de certification), Ex. Verisign,
  - ✓ La signature numérique des données générée par la personne ou l'entité prenant en charge la création ou l'authentification de ce certificat et servant d'autorité de certification.

  Usuellement, on distingue deux familles de certificats numériques :
  - ✓ les certificats de signature : utilisés pour signer des e-mails ou s'authentifier sur un site web.
  - ✓ les certificats de chiffrement : les gens qui vous envoient des e-mails utilisent la partie publique de votre certificat pour chiffrer le contenu que vous serez seul à pouvoir déchiffrer.

  Il existe deux façons distinctes de créer des certificats électroniques :
  - ✓ le mode décentralisé (le plus courant) qui consiste à faire créer, par l'utilisateur (ou, plus exactement par son logiciel ou carte à puce) la biclef cryptographique et de remettre la partie publique à l'AC qui va y adjoindre les informations de l'utilisateur et signer l'ensemble (information + clé publique)
  - ✓ le mode centralisé qui consiste en la création de la biclef par l'AC, qui génère le certificat et le remet avec la clé privée à son utilisateur.

  Les certificats électroniques respectent des standards spécifiant leur contenu de façon rigoureuse. On trouve parmi les plus connus et les plus utilisés :
  - ✓ la norme X.509 en version 1, 2, et 3, sur lequel se fondent certaines infrastructures à clés publiques.

- ✓ OpenPGP, format standard (normalisé dans le RFC 2440) de logiciels comme GnuPG.

Un Certificat électronique est géré tout au long de son cycle de vie (création, renouvellement et révocation) par l'autorité de Certification (CA) au moyen d'une infrastructure à clés publiques, ou PKI pour Public Key Infrastructure en anglais [32].

II.4. Principe d'un système cryptographique :

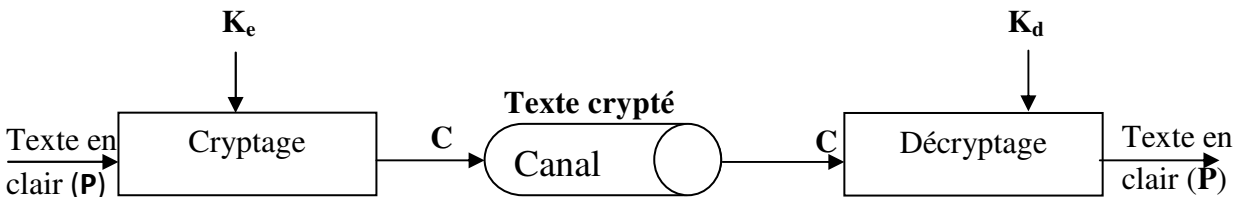


Figure II.1: Principe d'un système cryptographique

Dans le système cryptographique de la figure II.1, la fonction de cryptage notée E_{K_e} transforme un message appelé texte en clair (plaintext) et noté *P* en un texte crypté (ciphertext) noté *C*, qu'est le résultat de cryptage selon la formule suivante [33]:

$$C = E_{K_e}(P)$$

Où K_e est la clé de cryptage.

La conversion inverse ou le décryptage se fait avec la fonction D_{K_d} transforme *P* en *C*, qu'est le résultat de décryptage selon la formule suivante :

$$P = D_{K_d}(C)$$

Où K_d est la clé de décryptage.

Le type de relation qui unit les clefs K_e et K_d utilisées dans le cryptage et le décryptage permet de définir deux grandes catégories de systèmes cryptographiques [34]:

- ✓ *Les systèmes à clé secrète:* la clé est un secret partagé entre l'émetteur et le destinataire *(K_e=K_d).*
- ✓ *Les systèmes à clé publique:* aucune information secrète n'est partagée entre l'émetteur et le destinataire *($K_e \neq K_d$).*

II.5. Qualités d'un crypto-système:

Les qualités demandées à un système cryptographique sont résumées par les mots clefs suivants :

II.5.1. Principes de Kerckhoffs en cryptographie :

Pour briser un crypto-système, un opposant cherche à obtenir deux éléments d'information :

- Quel est le type de système de codage utilisé ?
- Quelle est la clé d'encodage utilisée ?

Bien entendu, son travail est simplifié (mais certainement pas terminé) s'il connait le type de système utilisé. Par contre, si son connaît la clé de chiffrement, le déchiffrement est immédiat.

Ce principe consiste à affirmer que la sécurité d'un crypto-système ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré.

Le premier à avoir formalisé ce principe est Auguste Kerckhoffs en janvier 1883 dans l'article "La cryptographie militaire" paru dans "le Journal des Sciences Militaires", où il disait [35]:

« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions ; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas ; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants :

1. Le système doit être matériellement, sinon mathématiquement, indéchiffrable.
2. Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.
3. La clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.
4. Il faut qu'il soit applicable à la correspondance télégraphique.

5. Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.
6. Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer ».

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

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

II.5.2. Les objectifs de la cryptographie :

La cryptographie a pour but de garantir la protection des communications transmises sur un canal public contre différents types d'adversaires. Il existe quatre grands objectifs pour le cryptage des données numériques, non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité:

1. **La confidentialité** ou masquage des données, caractéristique la plus utilisée, vise à rendre l'information inintelligible pour celui qui n'est pas en possession de la clef.
2. **L'authentification** autorise à l'émetteur de signer son message (par exemple par un mot de passe crypté), ce qui permet de contrôlé l'accès à des ressources uniquement aux personnes autorisées. Ainsi, le récepteur n'aura pas de doute sur l'identité du l'émetteur.
3. **L'intégrité** quant à elle va assurer au récepteur que le contenu du message n'a pas été altéré durant la communication.
4. **Le non répudiation** est la garantie qu'aucun des deux individus correspondants ne pourra nier la transaction. Elle se divisée en trois:
   - ✓ *Non-répudiation d'origine* l'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est effectivement le cas.
   - ✓ *Non-répudiation de réception* le receveur ne peut nier avoir reçu le message et il peut prouver qu'il ne l'a pas reçu si c'est effectivement le cas.
   - ✓ *Non-répudiation de transmission* l'émetteur du message ne peut nier avoir envoyé le message et il peut prouver qu'il ne l'a pas fait si c'est effectivement le cas.

La caractéristique majeure en imagerie est bien sûr la confidentialité. Mais la caractéristique intégrité, ainsi que les deux autres sont aussi importantes pour la protection des images.

II.6. Classification des algorithmes de cryptage :

Les algorithmes de cryptage peuvent être divisés en plusieurs classifications, selon la clé, la technique, le domaine de cryptage et le pourcentage des données cryptées.

II.6.1. Classification selon la clé de cryptage :

II.6.1.1. Le cryptage symétrique:

Le cryptage symétrique est aussi appelé cryptage à clé secrète (ou encore dit conventionnel), c'est la plus ancienne forme de chiffrement. Comme illustré à la figure II.2, en cryptage symétrique, les clés de cryptage et de décryptage sont identiques.

Car la sécurité d'un tel algorithme symétrique repose sur cette clé, l'expéditeur doit être transmise la clé de chiffrement / déchiffrement au destinataire à travers un canal sécurisé. Cet échange de clés est un inconvénient principal de cryptographie symétrique. Ce qui peut constituer une faille à la sécurité du système. Il est préférable d'échanger les clés manuellement lorsque cela est possible.

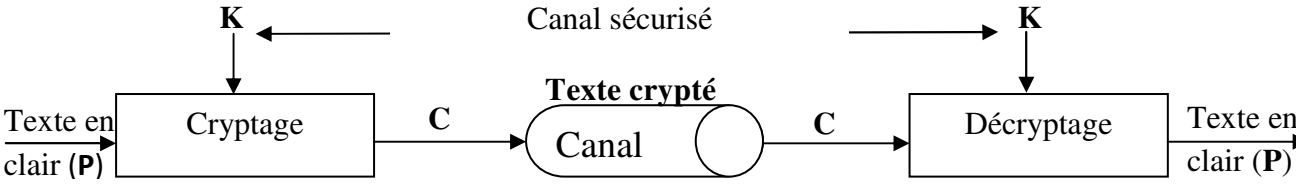


Figure II.2: Schéma simple d'un chiffrement symétrique.

Les algorithmes les plus répandus sont : RC4 (Rivest Cipher 4), DES (Data Encryption Standard), AES (Advanced Encryption Standard), 3DES (Triple DES), …etc.

II.6.1.2. Le cryptage asymétrique :

En 1976, Whitfield Diffie et Martin Hellman sont publiés le principe de chiffrement asymétrique (appelé aussi chiffrement a clés publiques) dans un ouvrage sur la cryptographie. La cryptographie asymétrique utilise deux clés différentes pour chaque utilisateur (voir figure II.3) :

Une clé publique pour le chiffrement K_e: diffusée en général et n'importe qui peut utiliser pour crypter un message.

Une clé secrète pour le déchiffrement K_d: est gardée privée et n'est connue que de l'utilisateur et qu'est sensé d'être en mesure de faire déchiffrement.

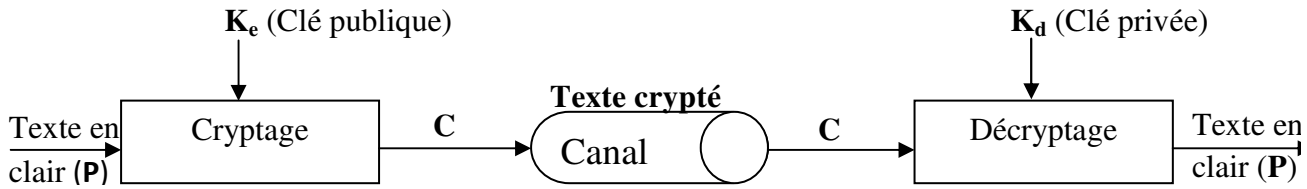


Figure II.3: Schéma simple d'un chiffrement asymétrique.

Le principal avantage de cryptage asymétrique que symétrique consiste à résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé puisque la clé privée n'est connue que par l'utilisateur. Bien que restent beaucoup moins efficaces en termes de temps de calcul que les algorithmes symétriques. Le cryptage à clé publique peut être préféré pour générer de petites séquences comme des signatures ou des clés secrètes pour le cryptage symétrique. Le cryptage symétrique peut être préféré pour crypter des grandes quantités de données comme les images.

Le premier système de cryptage à clé publique a été proposé en 1978 par R. Rivest, A. Shamir et L. Adleman, trois chercheurs du MIT, qui ont donné leur nom au système baptisé RSA.

II.6.1.3. Le cryptage hybride :

La cryptographie hybride utilise des algorithmes à clé publique et des algorithmes à clé privée, d'où l'adjectif hybride. Ce faisant, il combine les avantages des deux systèmes et pallie à certains inconvénients. En effet, un chiffrement hybride est rapide mais ne présente pas de faiblesse au niveau de la clé comme un chiffrement à clé publique.

La plupart des systèmes hybrides fonctionnent de la manière suivante:

Une clé aléatoire (ou pseudo-aléatoire) est générée pour l'algorithme symétrique (par exemple AES). Elle varie généralement entre 128, 256 ou 512 bits selon les algorithmes. Le destinataire génère alors une clé publique et une clé privée. La clé publique sert à chiffrer la clé aléatoire. Étant donné que cette dernière est courte, la chiffrer est rapide, alors que chiffrer le message avec un algorithme asymétrique aurait été bien plus long. Il ne reste plus qu'à envoyer le message chiffré accompagné de la clé chiffrée correspondante. Le destinataire utilise alors sa clé privée pour déchiffrer la clé aléatoire. Avec cette dernière, il retrouve le message via un déchiffrement symétrique[36] (voir figure II.4).

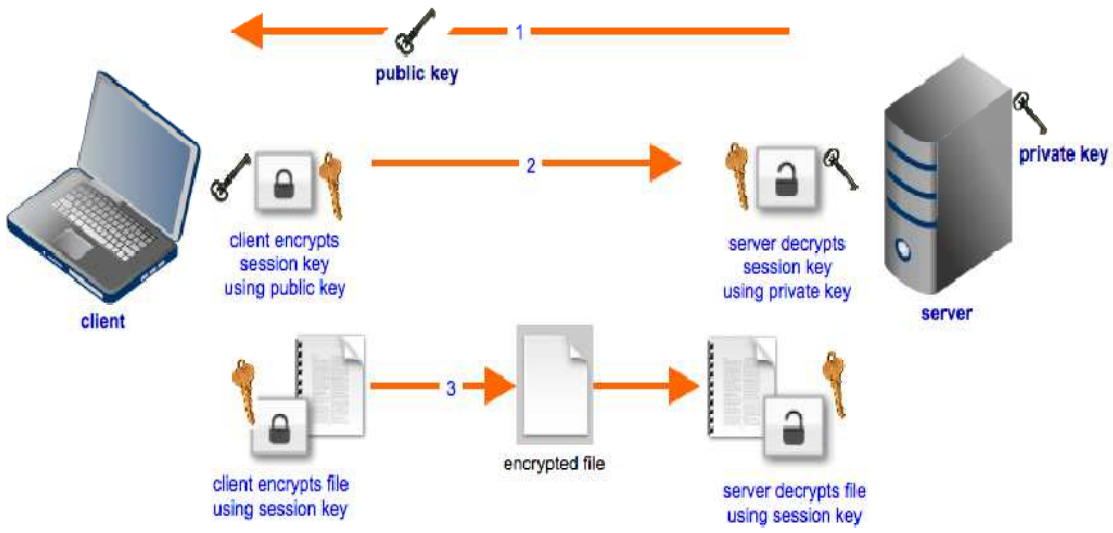


Figure II.4: Schéma d'un chiffrement hybride [37]

II.6.1.4. Comparaison entre les crypto-systèmes symétriques et asymétriques :

On peut résumer la comparaison entre les crypto-systèmes symétriques et asymétriques dans le tableau ce dessous:

Le type de crypto-système	Les avantages	Les inconvénients
crypto-système symétrique	• Clés relativement courtes (128 ou 256 bits). • Rapide. • Facile.	• Gestion des clés difficiles (nombreuses clés). • Difficulté de distribuer la clé secrète. • Ne permet pas de signature électronique.
crypto-système asymétrique	• Utilise deux clés différentes. • Fournit des garanties d'intégrité et de non répudiation par signature électronique.	• Des clés plus longues (1024 à 4096 bits). • Lenteur de calcul.

	• Très utile pour échanger les clés.	• Difficile.

Tableau II.1: La comparaison entre les crypto-systèmes symétriques et asymétriques [38]

II.6.2. Classification selon la technique de cryptage :

Les algorithmes de chiffrement peuvent être classés en fonction de la technique de cryptage en chiffrements par blocs et en chiffrements par flots:

II.6.2.1. Chiffrement par blocs (Bloc cipher):

Un chiffrement par blocs est un type d'algorithme de chiffrement symétrique qu'est découpé le texte en clair en blocs d'une taille fixée. Chaque bloc est crypté l'un après l'autre (voir figure II.5).

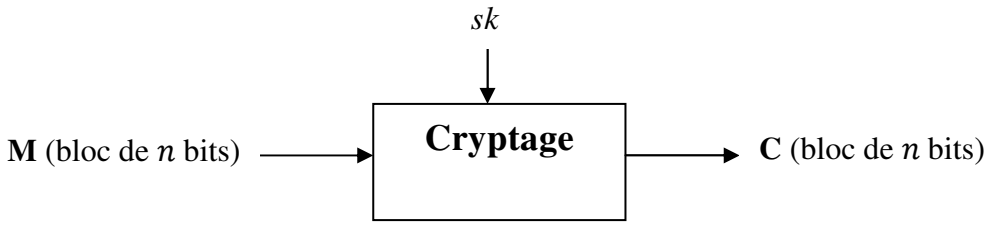


Figure II.5: Schéma simple d'un cryptage par blocs.

Une bonne sécurité est définie par une clé assez longue. Les clés très longues sont plus coûteuses en travail à cause notamment de leur génération, de leur transmission, de leur espace mémoire et de la difficulté de s'en rappeler (mots de passe).

La taille des blocs a un impact sur la sécurité et sur la complexité : plus la taille du bloc est grande, plus le chiffrement est efficace, mais plus les algorithmes et les dispositifs de cryptage et de décryptage sont complexes. Un exemple de cryptage par blocs est le schéma (DES) adopté par le gouvernement américain, en 1977, comme algorithme de chiffrement standard officiel.

II.6.2.2. Chiffrement par flots (Stream cipher):

Dans un cryptage par flots est appelé aussi cryptage en continu, le chiffrement fonctionne généralement sur de petites unités de texte en clair, généralement des bits (voir figure II.6).

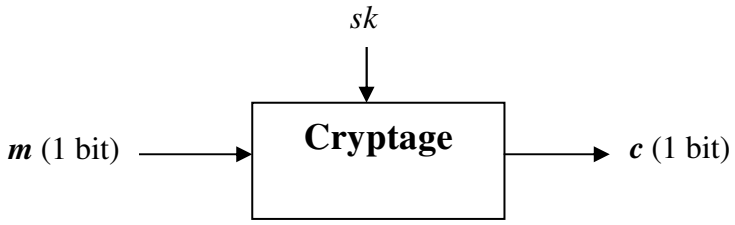


Figure II.6: Schéma simple d'un cryptage par flots.

Un crypto-système par flots génère une séquence de bits en tant que clé (appelée flux de clé) en utilisant un générateur de nombres pseudo-aléatoires (PRNG) qui étend une courte clé secrète (par exemple 128 bits) en une longue chaîne de bits (flux de clé). Le chiffrement est effectué en combinant le flux de clé avec le texte en clair habituellement par l'opération XOR bit à bit.

Les chiffrements par flots sont beaucoup plus rapides qu'un chiffrement par blocs, ils sont parfaitement adaptés à des moyens de calcul et de mémoire (cryptographie en temps réel) comme la cryptographie militaire, ou les communications pour garantir la confidentialité : GSM, Bluetooth, WiFi. Exemple de cryptage par flots est le schéma RC4 (1987).

II.6.2.3. Avantages et inconvénients du chiffrement par bloc et par flot :

- ✓ Le chiffrement par flot est plus rapide, surtout en implantation matérielle car la complexité matérielle est plus faible. Au contraire le schéma par bloc est plus lent et a une implantation matérielle ou logicielle plus coûteuse.
- ✓ Les chiffrements par flots peuvent être traités avec une mémoire limitée, ils traitent le message clair bit par bit et il n'est donc pas nécessaire de stocker tout le texte clair comme c'est le cas pour les systèmes de chiffrement par blocs, qui sont ne peut commencer à chiffrer et à déchiffrer un message que si l'on connaît la totalité d'un bloc.
- ✓ Les chiffrements par flots sont utilisés dans les communications pour garantir la confidentialité et Les schémas par blocs sont bien adaptés au stockage informatique.
- ✓ Les chiffrements par flot ne requièrent évidemment pas de padding, c'est-à-dire l'ajout de certains bits au message clair dont le seul objectif est d'atteindre une longueur multiple de la taille du bloc. Ceci peut s'avérer particulièrement souhaitable dans les applications où la bande passante est très limitée ou quand le protocole employé impose la transmission de paquets relativement courts [39].
- ✓ Un autre avantage du chiffrement par flot est que contrairement aux chiffrements par bloc, le processus de déchiffrement ne propage pas les erreurs de transmission.

Supposons qu'une erreur survenue au cours de la communication ait affecté un bit du message chiffré. Dans le cas d'un chiffrement à flot, cette erreur affecte uniquement le bit correspondant du texte clair, et ne le rend donc généralement pas complètement incompréhensible. Par contre, dans le cas d'un chiffrement par bloc, c'est tout le bloc contenant la position erronée qui devient incorrect après déchiffrement. Ainsi, une erreur sur un seul bit lors de la transmission affecte en réalité 128 bits du message clair [39].

II.6.3- Classification selon le pourcentage des données cryptées :

En ce qui concerne la quantité de données cryptées, le cryptage peut être divisé en cryptage total et cryptage partiel (également appelés le cryptage sélectif), selon le pourcentage des données cryptées.

Le cryptage sélectif est une approche récente permettant de réduire les temps de calcul pour des énormes volumes de données numériques à transmettre sur le réseau avec des clients ayant différentes capacités de réception. Cette approche ne chiffre qu'une partie des données afin de diminuer le temps de calcul tout en assurant une certaine sécurité. Dans le domaine de cryptage d'images, cryptage sélectif protège les parties les plus importantes des images tout en minimisant le temps de calcul pour des applications temps réel. Beaucoup de méthodes de cryptage sélectif ont été créées avec une approche de cryptage pour des images codées par transformée en cosinus discrète (DCT) [40].

II.6.4. Classification selon le domaine de cryptage :

Les algorithmes de chiffrement des images peuvent être classés selon le domaine d'application : les méthodes du domaine temporel/spatial ou bien celle du domaine fréquentiel. Le domaine fréquentiel est obtenu par une transformation discrète comme la DCT et la DFT.

II.6.4.1. Cryptage d'images dans le domaine spatial :

Dans le domaine spatial, on applique le schéma de cryptage sur le plan d'image lui même, et les approches de cette catégorie sont basées sur une manipulation directe des pixels d'une image.

Dans ces algorithmes, le chiffrement détruit la corrélation entre les pixels et rend les images cryptées incompressibles. Les pixels de l'image peuvent être reconstruits (récupérés) complètement par un processus inverse sans aucune perte d'information.

Les algorithmes de cryptage d'image dans le domaine spatial existants peuvent être classés en deux catégories.

- Dans la première catégorie, un pixel est considéré comme le plus petit élément, et une image numérique est considérée comme un ensemble de pixels.
- Toutefois, dans la deuxième classe, un pixel peut être en outre divisé en bits, sur lesquels des opérations au niveau de bits sont effectuées. Par exemple, un pixel dans une image en niveaux de gris est généralement constitué de 8 bits [41].

II.6.4.2. Cryptage d'images dans le domaine fréquentiel :

Les schémas de cryptage dans le domaine fréquentiel sont basés sur la modification de la fréquence de l'image en utilisant une transformation, ainsi, la reconstruction des pixels de l'image originale dans le processus de décryptage cause généralement une perte d'information [41].

II.7. La cryptographie basée sur une clé symétrique :

II.7.1. Principe :

Le processus de la méthode de cryptage basée sur une clé symétrique utilise deux opérations logiques, l'une est «XOR» et la seconde est une opération de «décalage circulaire». En décalage circulaire, nous avons utilisé à la fois des opérations de décalage circulaire gauche et droite. Tous ces types d'opérations fonctionnent entre la valeur binaire de l'image secrète compressée et la clé symétrique secrète respectivement pour produire des données chiffrées [42]. L'architecture de cette méthode est illustrée à la figure II.7.

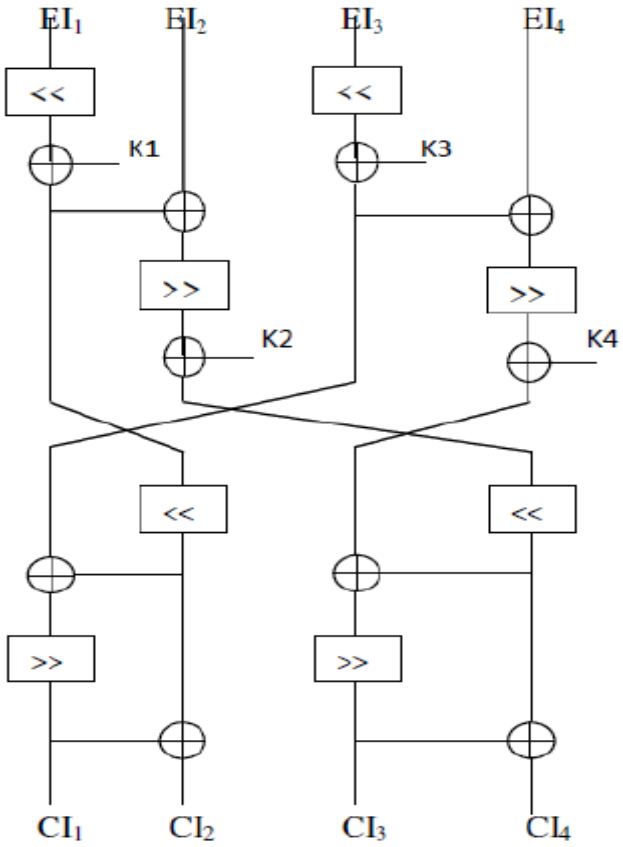


Figure II.7: Architecture de cryptage [42]

II.7.2. Le déroulement de chiffrement:

Le déroulement de cette méthode au phase de chiffrement se fait comme suite [42]:

1. Initialement, lisez la valeur binaire de l'image sécrétée compressée (EI) et sélectionnez 128 bits à la fois pour un traitement ultérieur avec une clé secrète (K) qui a également une taille de 128 bits.
2. Ensuite, EI et K sont divisés en quatre sous-parties égales (EI1, EI2, EI3, EI4) et (K1, K2, K3, K4) qui sont respectivement de 32 bits.
3. Après cela, nous avons appliqué un décalage circulaire gauche de 2 bits et un décalage circulaire droit de 2 bits sur les sous-parties de EI et effectuer une opération XOR entre les sous-parties de EI. L'opération XOR est également exécutée entre les sous-parties de EI et K.
4. Enfin, effectuez toutes les étapes prédéfinies une par une pour obtenir les données de chiffrement. Pour plus de description, pour plus d'informations consultez l'algorithme de chiffrement dans le dernier chapitre.

II.7.2. Le déroulement de déchiffrement :

L'architecture de décryptage proposée est illustrée à la figure III.8.

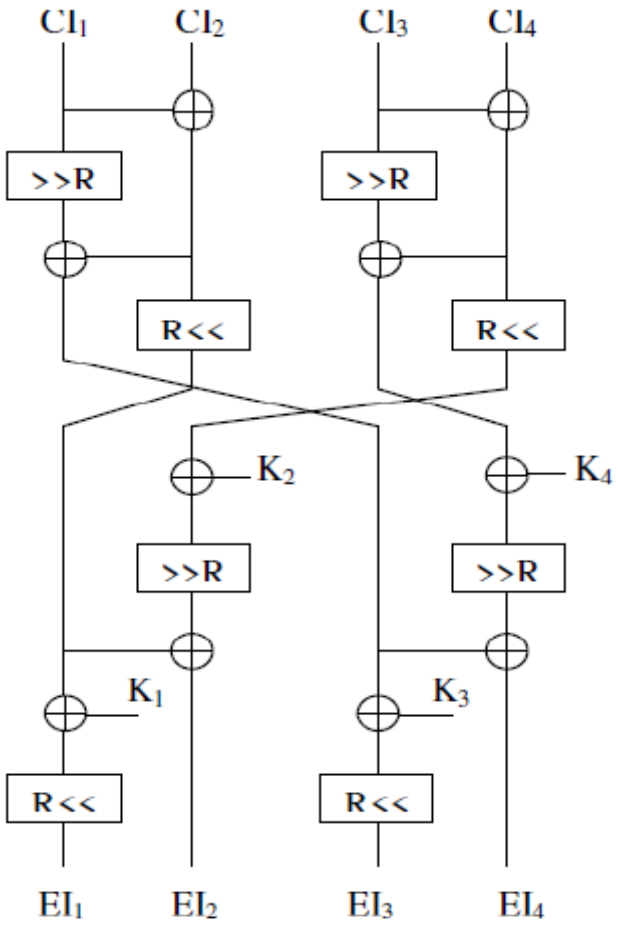


Figure II.8: Architecture de décryptage [42]

Le déroulement de décryptage est juste un processus inverse de cryptage où nous obtenons les données originales à partir des données de chiffrement [42] :

1- Nous lisons la valeur binaire de l'image cryptée (CI) et sélectionnons 128 bits à la fois pour un traitement ultérieur avec la clé secrète (K) qui a également une taille de 128 bits.
2- L'image cryptée (CI) et la clé K sont divisées en quatre sous-parties égales comme (CI_1, CI_2, CI_3, CI_4) et (K_1, K_2, K_3, K_4) qui sont respectivement de 32 bits.
3- Maintenant, une fois de plus, nous avons utilisé la même opération logique comme «XOR» et décalage circulaire, mais de manière différente pour obtenir les données d'origine. Opération de décalage circulaire exécutée dans l'ordre inverse dans le processus de décryptage comme celui que nous avons appliqué un décalage circulaire gauche de 2

bits et un décalage circulaire droit sur la sous-partie de CI en sens inverse et en effectuant une opération XOR entre les sous-parties de CI.

4- Effectuez également une opération XOR entre les sous-parties de CI et K. pour plus d'informations consultez l'algorithme de déchiffrement dans le dernier chapitre.

II.8. Quelques applications de la cryptographie :

La cryptographie est utilisée aujourd'hui dans de nombreuses applications : dans les téléphones portables, sur Internet ou pour la télévision à péage. Dans le cas des téléphones mobiles, la cryptographie est utilisée pour assurer la confidentialité des communications. En effet, la loi sur les télécommunications oblige les opérateurs à garantir la sécurité des communications des utilisateurs. En particulier dans le cas des téléphones mobiles, les communications entre le téléphone et la station hertzienne sont chiffrées. On utilise uniquement la cryptographie à clé secrète et l'algorithme de chiffrement est un algorithme par flot appelé A5. Sur Internet, la cryptographie permet de garantir la confidentialité de certaines communications comme la transmission du code d'une carte bleue (protocole SSL) ou d'assurer la confidentialité, l'intégrité et l'authentification de l'émetteur dans les messageries électroniques (protocole S/MIME).

Enfin, la cryptographie est aussi utilisée dans le cas des télévisions à péage pour que seuls les usagers autorisés puissent avoir accès aux programmes. Ainsi, les programmes sont chiffrés et seuls les abonnés connaissent la clé de déchiffrement. De plus, la cryptographie permet aujourd'hui par exemple de détecter les décodeurs pirates mais, pour des raisons économiques, ces systèmes sont peu employés [43].

II.9. Conclusion :

Dans ce chapitre nous avons présenté une introduction générale sur la cryptographie et montré qu'il existait de nombreuses algorithmes afin de sécuriser la transmission et le stockage des données soit textuelles ou image. Ensuite abordé les différent types de classifications des techniques de cryptage, et expliqué la différence entre eux, en mentionnant les avantages et les inconvénients de chaque technologie séparément.

Chapitre III :

Implémentation et résultat

Chapitre III

Implémentation et résultats

III.1. Introduction :

Dans ce chapitre, nous expliquons la cryptographie basée sur une clé symétrique, et le montrons en détail les algorithmes de chiffrement et de déchiffrement de cette méthode, et donnons une traduction de l'interface graphique de notre application de compression et de cryptage d'image. En plus d'afficher les résultats obtenus.

III.2. Environnement de développement :

Dans cette partie nous allons citer l'environnement matériel (Hardware) et logiciel (Software) utilisés dans l'implémentation de notre application.

III.2.1. Environnement matérielle :

L'implémentation de notre application a été réalisée sur un micro-portable HP personnel fonctionnant sous le système d'exploitation **Microsoft Windows 10 Home 64 bits** ayant les caractéristiques suivantes :

- Processeur: AMD A8-5545M APU with Radeon(tm) HD Graphics 1.70GHz
- Mémoire RAM: 8, 00 GO.
- Disque dure: 1 To SATA.
- Carte graphique: AMD Radeon HD 8510G

III.2.2- Environnement logiciel :

Nous avons implémenté notre application avec le langage de programmation **Matlab** avec la version **R2019a**.

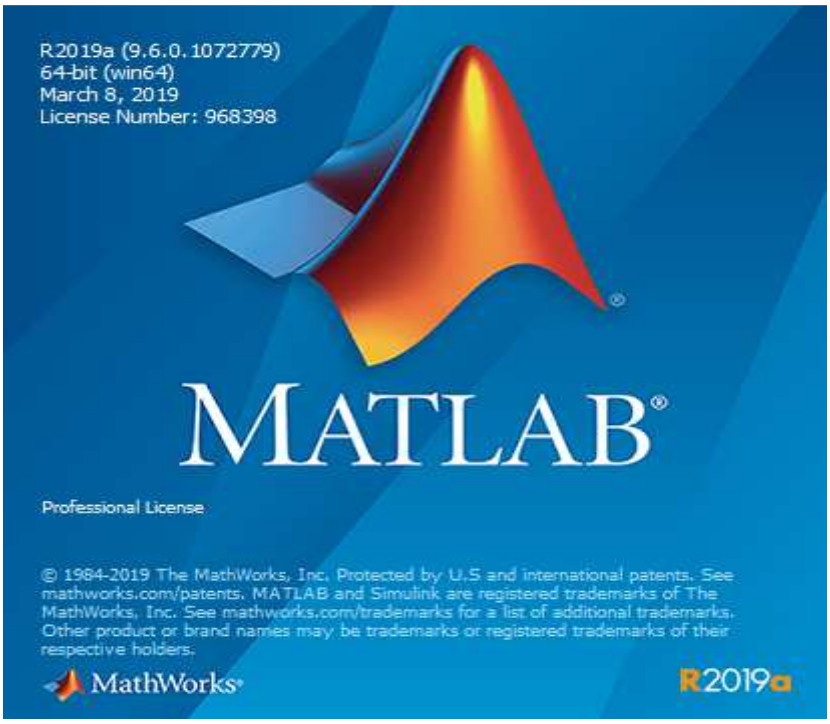


Figure III.1: Matlab R2019a

Le logiciel Matlab (abréviation de Matrix Laboratory) est un logiciel de calcul numérique et de visualisation graphique qui fonctionne sous Windows et sous Linux, utilise dans de nombreux domaines d'application (calcul matriciel, traitement de signal, traitement d'images, visualisations graphiques, etc.).

Afin de rendre l'utilisation des fonctions de traitement d'image plus facile, nous avons choisi de créer une interface graphique interactive appeler GUI (Graphical User Interface) sous MATLAB, navigable à la souris ou au clavier. Elle permette à l'utilisateur d'interagir avec un programme informatique, grâce à différents objets graphiques (boutons, menus, cases à cocher…).

Depuis la version 5.0 (1997), MATLAB possède un outil dédié à la création des interfaces graphiques GUI appelé GUIDE (Graphical User Interface Développent Environnent).Le GUIDE est un constructeur d'interface graphique qui regroupe tous les outils dont le programmeur a besoin pour créer une interface graphique de façon intuitive [44].

III.3. Aperçu d'application élaboré :

III.3.1. Hiérarchie :

Au départ, nous sélectionnons une image secrète, puis appliquons une technique de compression d'image qui est une transformation en ondelettes pour réduire la taille de l'image secrète. Après cela, nous lisons des données binaires à partir d'une image compressée qui sont transmises à la technique de cryptage pour les convertir en données de cryptage en utilisant la technique de cryptographie symétrique. Ensuite, nous faisons le processus inverse, c'est-à-dire décoder et décompresser pour atteindre l'image secrète. En plus de tout cela, calculer les paramètres et afficher le résultat de décompression «CR, MSE, PSNR» pour bien valider la qualité de l'image reconstruite. La figure III.2 illustre l'organigramme du logiciel élaboré.

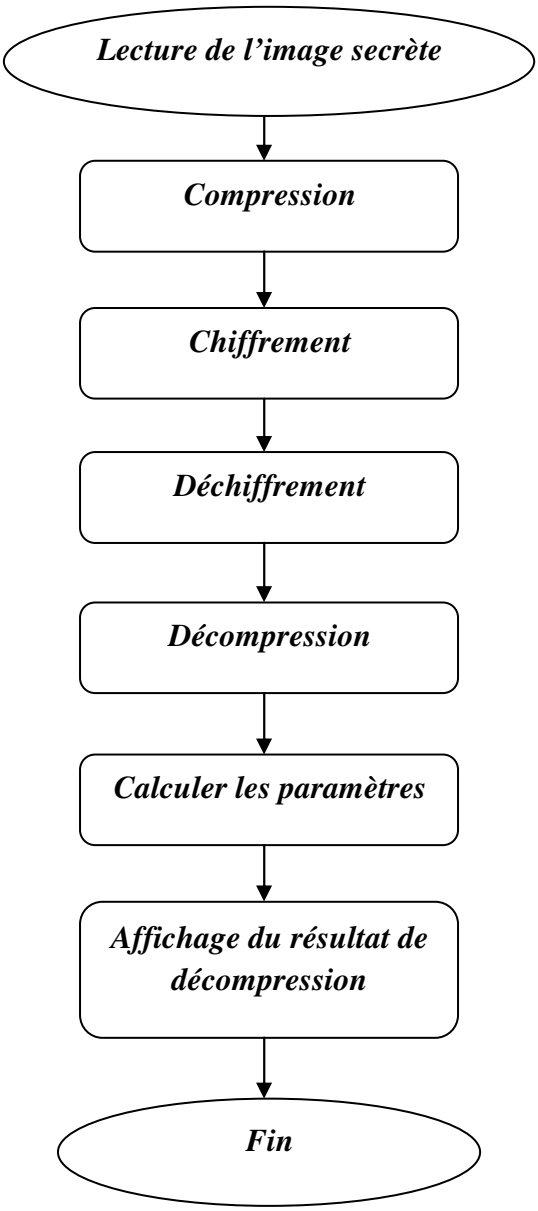


Figure III. 2 : Organigramme d'application élaboré

III.3.2. L'interface graphique de notre application:

On exécute le fichier 'ImageEncryptionGui.m' de notre projet contient des fonctions, qu'elle utilise DWT (ondelette discrète) comme méthode de compression d'image et la technique de cryptographie symétrique comme méthode de cryptage d'image compressée.

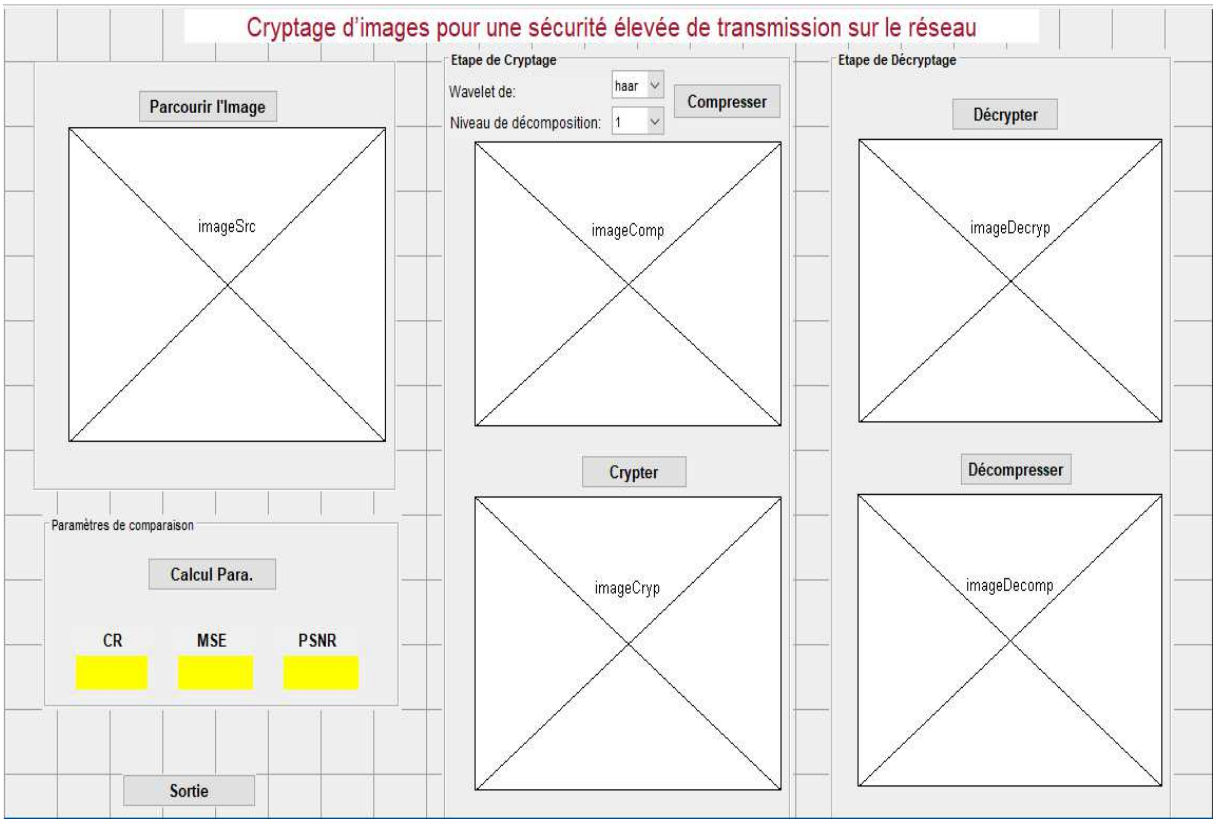


Figure III.3: L'outil GUIDE

III.3.3. Description de l'interface graphique :

La figure III.3 représente l'interface graphique de notre application de compression et de cryptage d'image. On a présenté maintenant le rôle de chaque bouton et Pop-Up Menu:

1- Le bouton « Parcourir l'image » pour parcourir et choisi une image qu'existe déjà, et afficher dans l'axe «imageSrc ».
2- Le bouton « Compresser » pour compresser l'image choisi, et afficher dans l'axe «imageComp ».
3- Le bouton « Crypter » pour crypter l'image compressée, et afficher dans l'axe « imageCryp».

4- Le bouton « Décrypter » pour décrypter l'image cryptée, et afficher dans l'axe «imageDecryp ».
5- Le bouton « Décompresser » pour décompresser l'image, et afficher dans l'axe «imageDecomp ».
6- Le bouton « Calcul para. » pour afficher CR (Compression Ratio), MSE (Mean Square Error) et PSNR (Peak Signal to Noise Ratio) dans des zones de texte correspondantes.
7- Le bouton «Sortie» pour quitter l'application.
8- 'Pop-Up Menu 1' pour choisir entre les ondelettes de Haar ou les ondelettes de Daubechies1 (db1).
9- 'Pop-Up Menu 2' pour choisir entre les différentes niveaux de décomposition.

III.3.4. Bibliothèque d'image:

La figure III.4 montre l'image utilisée dans ce chapitre. Cette image présentait toutes les qualités pour illustrer la compression numérique, car elle comporte un grand nombre de détails et de textures, qui permet de bien tester les différents algorithmes de traitement d'image.

Cette image a été utilisée pour la première fois en 1973, par Alexander Sawchuk, un professeur de génie électrique, recherchait un exemple d'image à scanner pour une conférence sur la compression d'image numérique. C'est alors qu'il aperçoit un exemplaire du magazine playboy dont il scanne la partie haute de l'image pour ne garder que la tête de la modèle surnommée « Lenna » [45].

Figure III. 4: Lenna.bmp. 260ko

III.4. Résultats:

Cette section montrera divers résultats de l'application des techniques de compression et de cryptage d'image:

L'interface III.5 se montre la forme de compression, cryptage, décryptage et de décompression de l'image secrète :

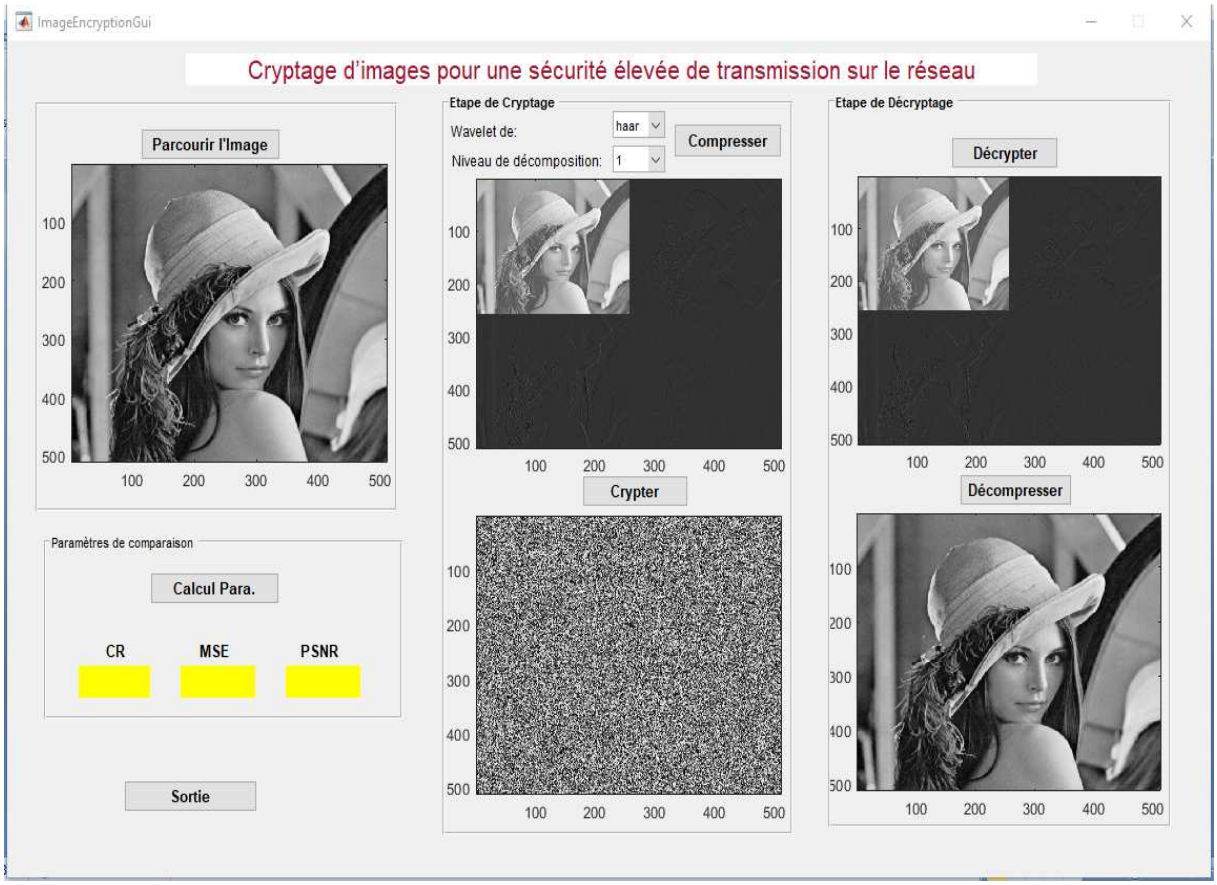


Figure III.5: Forme de compression et de cryptage

III.4.1. La compression :

La figure III.6 montre l'image au niveau de gris de différentes niveaux de compression par transformée en ondelettes de Haar :

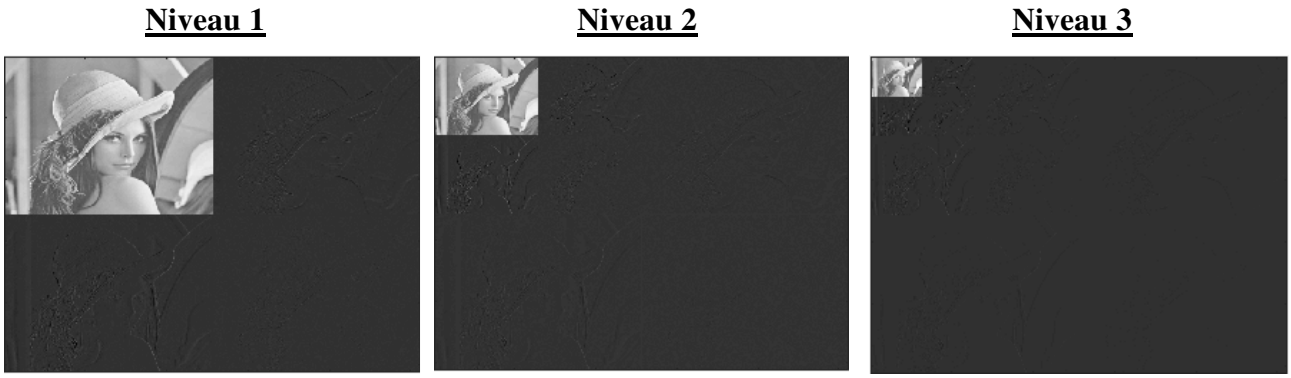


Figure III.6 : Transformée en ondelettes de Haar

La figure III.7 montre l'image au niveau de gris de différentes niveaux de compression par transformée en ondelettes de Daubechies1 (db1):

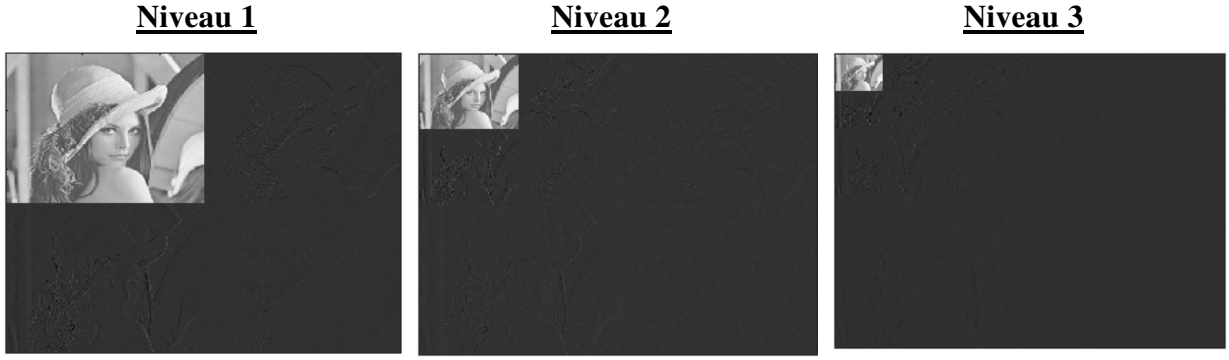


Figure III.7 : Transformée en ondelettes de Daubechies1 (db1)

III.4.2. Le cryptage:

La figure III.9 montre l'image au niveau de gris est cryptée :

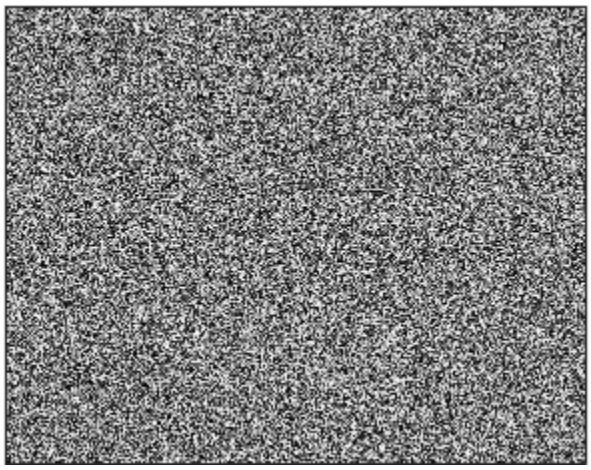


Figure III.8: L'image cryptée.

III.4.3. Le décryptage:

La figure III.9 montre l'image au niveau de gris est décryptée :

Figure III.9: L'image décryptée.

III.4.4. La décompression :

La figure III.10 montre l'image au niveau de gris après la décompression (compression par Transformée en ondelettes de Haar à différentes niveaux):

Niveau 1 **Niveau 2** **Niveau 3**

Figure III.10: L'image décompressée (compression par Transformée en ondelettes de Haar)

La figure III.11 montre l'image au niveau de gris après la décompression (compression par Transformée en ondelettes de Daubechies1 (db1) à différentes niveaux):

Figure III.11: L'image décompressée (compression par Transformée en ondelettes de db1)

III.5. Conclusion:

Dans ce chapitre, nous passons l'image secrète à travers les différentes étapes de notre application, en commençant par le choix de l'image secrète jusqu'à la décompression, en passant par le processus de compression, de cryptage et de décryptage.

En l'étape de compression, nous avons testé sur l'image secrète deux type de compression par transformée en ondelettes (transformée en ondelettes de Haar et de Daubechies1) a différentes niveaux (trois niveaux).

En plus de ce qui précède, l'image a été cryptée via l'algorithme de chiffrement de la cryptographie basée sur une clé symétrique. Ensuite, le processus inverse via l'algorithme de déchiffrement. Et la dernière étape était la décompression pour obtenir une image approximative de l'image secrète.

Conclusion générale :

Pour le moment, les images sont de plus en plus utilisées par des individus ou des organisations, entraînant une transmission accrue de ce type de données entre les réseaux en général et l'Internet en particulier et la confidentialité de ce type de données est devenue indispensable pour ne pas être utilisé par des personnes illégales. Par conséquent, de nombreuses techniques de chiffrement ont été développées, et certaines des plus courantes et des plus utilisées ont été expliquées dans le deuxième chapitre de cette mémoire.

La compression des données est appelée à prendre un rôle encore plus important, en raison du développement des différents réseaux de télécommunications. Son importance est surtout due au décalage qu'existe entre les possibilités matérielles des dispositifs que nous utilisons dans notre vie quotidienne et les besoins qu'expriment les applications. Nous avons consacré le premier chapitre à expliquer certains des algorithmes les plus populaires: le codage de Shannon-Fano, Huffman, Run length encoding (RLE) et Lempel-Ziv-Welch (LZW) pour les algorithmes de compression sans perte, alors que nous avons expliqué deux types de méthode de compression avec perte la compression JPEG et la transformation en ondelettes.

Nous avons élaboré une technique de compression et sécurité d'images pour assurer la sécurité de transmission sur le réseau. Coté compression et cryptage, nous avons utilisé la compression par transformée en ondelettes et la cryptographie basée sur une clé symétrique pour le coté chiffrement.

[1] https://www.computerhope.com/jargon/i/image.htm. consulté le 05/07/2020.

[2] KADDOUR Chakib, “Généralités sur les traitements d’images”.

[3] S. Pigeon, “Contributions à la compression de données”, Thèse présentée à la Faculté des arts et sciences en vue de l’obtention du grade Philosophie Doctor (Ph. D.) en Informatique, Université de Montréal, Décembre 2001.

[4] V. Beaudoin, “Développement de nouvelles techniques de compression de données sans perte”, Thèse Maître en sciences (M.Sc.), Université Laval Québec, pp 08-09, 2008.

[5] B.O’Hanen, M.Wisan,“JPEG Compression”, Academic press, December 16, 2005.

[6] M.Sharma, “Compression Using Huffman Coding”. International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.

[7] A.Shahbahrami, R.Bahrampour, M. S.Rostami, M. A Mobarhan, “EVALUATION OF HUMMAN AND ARITHMETIC ALGORITHEM FOR MULTIMEDIA COMPRESSION STANDARDS”, International Journal of Computer Science Engineering and Applications, August 2011.

[8] H. Husseen, S. Mahmud, and R. J. Mohammed, “Image Compression using Run Length Encoding Algorithm”, For Pure and Applied Sciences, Vol. 24, Issue 1, p4, 2011.

[9] T. Cormen, C. Leiserson, R. Rivest, C. Stein, ”Introduction to Algorithms” Third Edition, Massachusetts: MITPress, 2010.

[10] Dheemanth H N, “LZW Data Compression”, American Journal of Engineering Research (AJER), vol. 3, no. 2, pp. 22-26, 2014.

[11] KADDOUR Chakib, “Généralités sur les traitements d’images”.

[12] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html, visité le 05/06/2020.

[13] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html, visité le 07/06/2020.

[14] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html, visité le 15/06/2020.

[15] Muzhir Al-Ani. Fouad H Awad, “THE JPEG IMAGE COMPRESSION ALGORITHM”. International Journal of Advances in Engineering & Technology IJAET. May 2013.

[16] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html, visité le 28/06/2020.

[17] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html, visité le 28/06/2020.

[18] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html, visité le 28/06/2020.

[19] Brahim Mohamed, "Utilisation des Ondelettes dans la Segmentation d'images Medicales", Mémoire de Magister, Ecole Nationale Polytechnique.

[20] H. Sharabty. "Diagnostic de la somnolence d'un opérateur : Analyse des signaux physiologiques". Thèse de Doctorat, Université Paul Sabatier Toulouse III, 2007.

[21] Hamdad Nassima, "Transformée de Huang-Hilbert : application de la détection des défauts", Mémoire de Magister, Université Mouloud Mammeri, Mars 2013.

[22] M.Chendeb. "Détection et classification des signaux non stationnaires par utilisation des ondelettes. Application aux signaux électromyographiques utérine". Thèse de Doctorat, Université de Technologie de Troyes, 2006.

[23] A. Graps, "An introduction to Wavelets", IEEE Computational Science and Engineering, vol. 2, no. 2 ,1995.

[24] O. Soltani. "Restauration D'Images Satellites Via la Transformée en Ondelette" .Thèse de Magister, 2011. Université de Batna.

[25] F. Truchetet. "Ondelettes pour le signal numérique". Editions Hermes, 1998.

[26] Consultative Committee for Space Data Systems (CCSDS). "IMAGE DATA COMPRESSION". Washington, DC, USA. September 2017.

[27] R. Sylvain, "La Compression de Données", Club Photoshop de Nantes. Conférence du 14 octobre 1999. P 18.

[28] Chérif TAOUCHE, "Implémentation d'un Environnement Parallèle pour la Compression d'Images à l'aide des Fractales", Thème Magister en Informatique, Université Mentouri Constantine, P 40,2005.

[29] Leonard M Adleman. "Molecular computation of solutions to combinatorial problems". American Association for the Advancement of Science, Vol. 266, Issue 5187:1021–1024, 11 Novembre 1994.

[30] AK Verma, Mayank Dave, and RC Joshi. “Dna cryptography : a novel paradigm for secure routing in mobile ad hoc networks (manets) ”. Journal of Discrete Mathematical Sciences and Cryptography, 11(4) :393–404, 2008.

[31] http://helios.mi.parisdescartes.fr/~mea/cours/Mi/crypto_synthese.pdf , consulté le 25/06/2020.

[32] http://helios.mi.parisdescartes.fr/~mea/cours/Mi/crypto_synthese.pdf , consulté le 05/07/2020.

[33] Abd El-Samie, Fathi E, and al., “Image encryption : a communication perspective”, CRC press, 2013.

[34] B. Schneier, “Cryptographie appliquée : Algorithme, protocoles et codes sources en C”, Vuibert Informatique, deuxième édition, janvier 2001.

[35] Kerckhoffs, A., “La cryptographie militaire”, Journal des sciences militaires, Janvier 1883.

[36] Mohand-Amokrane BIR Lyes DAHMOUNI, “Etude et implémentation d’algorithmes de chiffrement à clé secrète et à clé publique : Application au cryptage de la parole”, master académique en Automatique et Systèmes, Université Mouloud Mammeri De Tizi-Ouzou, 2018.

[37] L. SAOUDI, “initiation à la cryptographie”, support de cours du module Sécurité informatique, université de Msila, Année 2015/2016.

[38] Merdjal choumaissa, Merakchi Ahlam, “Cryptage d’image par un signal unidimensionnel quelconque”, Mémoire de Master informatique vision artificielle, Université LARBI BEN M’HIDI, OUM EL BOUAGHI, Année 2018.

[39] NKAPKOP Jean De Dieu, “Evaluation d'un algorithme de cryptage chaotique des images basé sur le modèle du perceptron”, Mémoire de Master II en EEA, Université de Ngaoundéré, 2012.

[40] William Puech, José Rodrigues, Jean-Eric Develay-Morice, “Transfert sécuriste d'images médicales par codage conjoint : cryptage sélectif par AES en mode par flot et compression JPEG”, Traitement du Signal, 23 (3), Gretsi, 2006.

[41] A. Beloucif, “Contribution à l’étude des mécanismes cryptographiques”, Thèse de Doctorat en Informatique, Université de Batna2, 2016.

[42] Piyush Kumar Shukla, "Design and Development of Image Security Technique by Using Cryptography and Steganography: A Combine Approach", International Journal of Image, Graphics and Signal Processing, Avril 2018.

[43] https://www.techniques-ingenieur.fr , consulté le 05/07/2020.

[44] https://briot-jerome.developpez.com/matlab/tutoriels/introduction-programmation-interfaces-graphiques/ , consulté le 05/07/2020.

[45] https://phototrend.fr/2016/07/dessous-images-lenna-icone-informatique-de-playboy/, consulté le 05/07/2020.