الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Mention Électronique
Spécialité système de télécommunication

présenté par

Ameur Fathi

&

Zerrouki Fatima Roumayssa

Conception et réalisation d'un système hybride pour la compression et la sécurisation des documents

Proposé par : Mlle BENBLIDIA NADJIA & Mlle REGUIEG F.ZOHRA

Année Universitaire 2017-2018

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.

Nous adressons le grand remerciement à notre promotrice Mlle BenblidiaNadjia et notre Co-promotrice Mlle ReguiegF.Zohrapour leurs patiences, leurs conseils et leurs encouragements.

Nous tenons également à remercier messieurs les membres de jury pour l'honneur qu'ils nous ont fait en acceptant de siéger à notre soutenance, tout particulièrement :

Mr Nouamane pour nous avoir fait l'honneur de présider le jury de cette mémoire.

Nous souhaitons exprimer notre gratitude à Mr Zahir pour avoir faire de lecteur de notre mémoire, aller l'examiner et il peut évaluer cette mémoire. Nous vous remercions pour l'intérêt que vous avez porté à ce travail et pour vos précieux conseils et remarques.

Finalement, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues et à tout ce qui participe de réaliser ce mémoire.
Ainsi que l'ensemble des enseignants qui ont contribué à notre formation.

Dédicaces

A l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur : maman que j'adore.

A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet : mon fiancé Nasr El-Dine sans oublié ma belle-famille.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à mon frère Abdelmounaim et mes sœurs Sarah & Asma, mon neveu Samy, mon beau-frère Yassine, et mon binôme Fathi, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et frères de cœur, toi Bouchra, Amina, Hamida, H.Abdelrrahmane, Billel et Sidou.

Zerrouki F.Roumayssa

Dédicaces

Je dédie ce modeste travail à :

A mes parents . Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagnera toujours. Que dieu vous procure bonne santé et longue vie.

A mes chers et adorables frères Hachemi Amine et Akram

A mes chers oncles et tantes, veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.

A mes amis de toujours : Billel, Sidou, Hinane, Farah

En souvenir de notre sincère et profonde amitié et des moments

Agréables que nous avons passés ensemble.

Ameur Fathi

ملخص: يتم استخدام تقنيات الضغط والتشفير أكثرفأكثر كل يوم لان الضغط يقلل من حجم الملف والتشفير يوفر الأمان لملف معين يجب ارساله عبر شبكة غير موثوقة. في المشروع، هدفنا الرئيسي هو ضغط ملف صورة باستخدام خوارزمية ضغط وتشفيره وارساله بحيث يحصل المستلم عل ملف الصورة بتنسيق أكثر انضغاطا للقيام بذلك، قمنا أولا بمسح ملف الصورة وضغطناه باستخدام نظام مختلط (DWT-Huffman). بعد ذلك، قمنا بتشفيرها باستخدام خوارزمية AES. تم تطبيق العملية العكسية للحصول على ملف الصورة الاصلي.

كلمات الجوهرية : ضغط الصور، نسبة الضغط، PSNR, التشفير.

Résumé : Les techniques de compression et de cryptage sont de plus en plus utilisées de nos jours, car la compression réduit la taille des fichiers et le cryptage assure la sécurité de fichier particulier qui doit être envoyé sur un réseau peu fiable. Dans ce projet, notre objectif principal est de compresser un fichier image à l'aide d'un algorithme de compression, le crypter et l'envoyer afin que le destinataire obtienne le fichier image dans un format compressé(plus petite taille). Pour ce faire, nous avons d'abord importer un fichier image et l'avons compressé en utilisant un système hybride (ondelettes -Huffman). Après cela, nous l'avons crypté à l'aide d'un algorithme AES (standard de chiffrement avancé). Le processus inverse a été appliqué pour obtenir le fichier image d'origine.

Mots clés : compression d'image, taux de compression, PSNR, chiffrement

Abstract: Compression and encryption techniques are being used more and more nowadays, because compression reduces file's size and encryption provides security for a particular file that must be sent over an unreliable network. In this project, our main goal is to compress an image file using a compression algorithm, encrypt it, and send it so that the recipient gets the image file in a more compressed format. To do this, we first scanned an image file and compressed it using a hybrid system (DWT-Huffman). After that, we encrypted it using an AES algorithm. The reverse process was applied to obtain the original image file.

Keywords: image compression, compression rate ratio, PSNR, encryption

Listes des acronymes et abréviations

RLC	Run Length Coding
LZW	Lempel Ziv Welch
DCT	Transformée Cosinus Discrète
JPEG	Joint Photographic Expert Group
GIF	Graphic Interchange Format
DWT	Discrete Wavelete Transform
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
RLE	Run Length Encoding
AES	Advanced Encryption Standard
DES	Data Encryption Standard
IDEA	International Data Encryption Algorithm
HMI	Human-machine Interface
GUI	Graphical User Interface

Table des matières

Liste des figures

Introduction générale

L'utilisation des technologies de l'information et des télécommunications dans la vie quotidienne a évolué ces dernières années d'une façon notable. La compression et le cryptage de données sont deux technologies dont l'importance croit d'une manière exponentielle dans une myriade d'applications.

Actuellement, Les chercheurs ont développé de nombreuses méthodes de compression de données déduites de la théorie de l'information et faisant appel à de nombreux domaines des mathématiques et de l'informatique.

La compression est un traitement sur une donnée qui a pour but de diminuer sa taille et donc de faciliter son stockage. La compression d'image fait l'objet de nombreuses études qui portent sur l'amélioration des algorithmes de compression ainsi que la mise au point de nouvelles techniques et formats de compression. Deux sortes de techniques permettent la compression des images : les méthodes réversibles, c'est à dire sans pertes, qui conduisent à de faibles taux de compression et celles appelées irréversibles et qui permettent de compresser fortement les images mais au prix de certaines distorsions [1].

Parmi les méthodes dites sans pertes, citons la technique RLC (codage par plage), le codage de Huffman et la compression LZW (LempelZiv Welch). Ces méthodes opèrent avec des facteurs de compression allant de 1,2 à 2,5[2] .Quant aux méthodes à fort taux de compression, les plus efficaces sont basées sur de puissants outils tels que la transformée discrète en cosinus (DCT) et la transformée en ondelettes. Avec de tels outils, le fichier image subit une forte compression mais avec des pertes se traduisant par des dégradations plus ou moins perceptibles à l'œil nu lors de la décompression.

Parmi les méthodes à fort taux de compression, on distingue le format JPEG (Joint Photographic Expert Group) et la méthode GIF (Graphic Interchange Format)[3].JPEG

reste néanmoins le plus couramment employé actuellement. Son principe est de diviser l'image en blocs carrés de 8 x 8 pixels et de coder les valeurs les plus proches dans chaque bloc sur quelques bits. En procédant de la sorte, nous pouvons atteindre des taux de compression supérieurs à 90 % tout en restituant des images avec une perte d'informations à peine perceptible [4]. Mais, à l'instar d'autres méthodes irréversibles, le temps d'exécution est très lent surtout dans le cas d'images de grande taille. De plus, des artefacts apparaissent dans celles-ci sous forme de mosaïques. Pour réduire les distorsions causées par les méthodes à fort taux de compression, nous avons élaboré dans notre étude une technique hybride basé sur la transformée en ondelette discrète DWT et Huffman.

En outre, l'usage excessif des réseaux informatiques pour le transfert des données doit évidemment obéir à un double objectif : la réduction du volume des données afin de désencombrer le maximum possible les réseaux publics de communication et la confidentialité en vue de garantir un niveau de sécurité optimum. Dans ce sens et afin d'assurer l'optimisation et la sécurisation de la transmission et du stockage des images fixes, nous proposons d'appliquer un cryptage à base de l'algorithme AES sur notre technique hybride basé sur DWT et Huffman.

Pour mener à bien notre travail, nous avons structuré notre mémoire en quatre chapitres :

- le premier chapitre introduit la compression ; des définitions et des notions essentielles sur les différents types de compression présentées. Nous décrivons par la suite, les algorithmes utilisés ainsi que les paramètres permettant d'évaluer leurs performances.

- le deuxième chapitre, nous amène dans le monde de la cryptologie ; en commençant par un court historique suivi par une présentation des méthodes de cryptage parmi les plus utilisées.

- le troisième chapitre décrit l'approche de compression d'images hybride élaborée dans le cadre de ce travail. Dans un premier temps, la méthode exploite la transformée en ondelette discrète DWT et le codage

Huffman. Par la suite, nous introduisons le cryptage à base de l'algorithme AES à la méthode de compression d'images hybride précédemment élaborée.

- le dernier chapitre illustre l'application développée sous MATLAB. Nous commencerons par présenter l'environnement de travail ainsi que l'interface générale de notre application. Puis, nous illustrons quelques résultats obtenus en appliquant notre système à différentes images.

Chapitre 1 Techniques De Compression

1.1 Introduction

On parle aujourd'hui de globalisation ; il faut communiquer en temps réel n'importe où dans le monde entier. La vitesse des communications ne peut pas être augmentée sans l'aide des techniques de compression. Tous les types des signaux intéressants pour les transmissions : les logiciels, les textes, la parole, la musique, les images, doivent être comprimés. La compression ne doit pas conduire à des distorsions saisissables et doit être réalisée avec des taux de compression les plus élevés possibles. Comme les données sont de plus en plus volumineuses, la compression est indispensable pour le stockage et la transmission des données, notamment les images.

1.2 La compression

La compression d'images minimise la taille en octets d'un fichier graphique sans dégrader la qualité de l'image à un niveau inacceptable. La réduction de la taille des fichiers permet de stocker davantage d'images dans une quantité donnée de disque ou d'espace mémoire. Cela réduit également le temps nécessaire pour que les images soient envoyées. En informatique, la compression de données est la technique dans laquelle on emploie une paire de fonctions C et D sur des chaines. La fonction C a pour objectif de compresser les données X et la fonction D, de les décompresser. L'effet souhaite est d'avoir $|C(x)| < |x|$.

On peut distinguer deux grandes familles de la compression [5]: les méthodes dites sans perte ou réversibles garantissent la restitution parfaite des images, alors que les méthodes dites avec pertes ou irréversibles modifient plus ou moins la valeur des pixels.

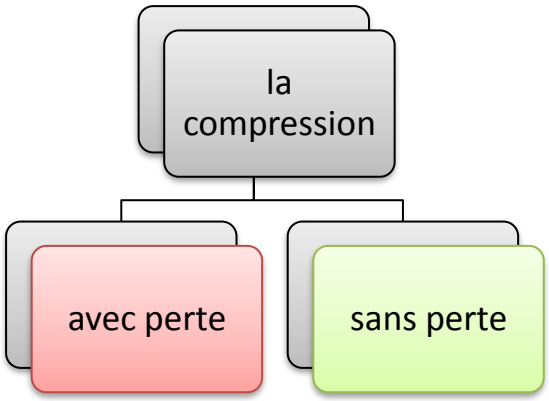

Figure 1. 1 : Types de compression

1.2.1 La compression sans perte

Comme son nom l'indique, ce type de compression n'occasionne aucune perte de données.

Cette compression conservatrice est utilisée dans des applications comme l'archivage des images médicales, l'imagerie satellitaires (le coût des images est élevé et les détails sont importants), les textes, les programmes et tout autre type de données nécessitant une conservation à l'identique de données [6]. Plusieurs méthodes sont employées parmi lesquelles nous citons :

- Codage de Huffman
- Codage de Shannon-Fano
- Codage arithmétique

1.2.1.1 Le codage de Huffman

C'est un processus qui permet de compresser des données informatiques afin de libérer de la place dans la mémoire d'un ordinateur. Or tout fichier informatique (qu'il s'agisse d'un fichier texte, d'une image ou d'une musique) est formé d'une suite de caractères. Chacun de ces caractères étant lui-même codé par une suite de 0 et de 1 [7].

La procédure de codage suit les étapes suivantes :

- Les probabilités d'occurrence de chaque message sont placées dans une liste dans un ordre décroissant. Nous dirons que la liste est composée d'enfants.
- Les deux probabilités les plus faibles sont identifiées en fin de liste.
- La somme des deux probabilités est placée à sa place dans la liste triée. Elle constitue un nœud parent. Les deux enfants sont retirés de la liste.
- Le chemin «enfant de plus faible probabilité, parent» est codé par un 1, l'autre par un0
- La procédure reprend à l'étape 2 jusqu'à ce qu'il ne reste plus qu'une probabilité dans la liste.

Malgré son ancienneté, cette méthode est toujours remise au goût du jour, et offre des performances appréciables. En effet, beaucoup de recherches en algorithmique ont permis d'améliorer les fonctionnalités de la méthode Huffman de base, en utilisant par exemple les arbres binaires, arbres équilibrés, etc. [8]

1.2.1.2 Codage de Shannon-Fano

Une méthode de codage basée sur de simples connaissances de la probabilité d'occurrence de chaque symbole dans le message [9].

La procédure se décrit ainsi :

- Classer les n fréquences non nulles {f i} par ordre décroissant.
- Répartir la table des fréquences en deux sous tables de fréquences proches. Poursuivre l'arborescence jusqu'à ce que toutes les fréquences soient isolées.

1.2.1.3 Codage Arithmétique

Contrairement aux algorithmes de Huffman et de Shannon-Fano qui associent à des symboles des motifs binaires dont la taille dépend de leur distribution, le codeur arithmétique traite le fichier dans son ensemble, en lui associant un unique nombre décimal rationnel. Ce nombre compris entre 0 et 1, possède d'autant moins de chiffres après la virgule que le fichier, dont il est issu, est redondant.

Ces chiffres décimaux dépendent non seulement des symboles du fichier dans l'ordre où ils apparaissent, mais aussi de leur distribution statistique [9].

1.2.2 La compression avec perte

Afin d'atteindre de forts taux de compression, on utilise les méthodes avec perte. Les algorithmes issus de cette catégorie, délivrent après compression une image différente qui contient beaucoup moins d'information que l'image originale. Cette modification est plus ou moins visible, selon le degré de compression. L'attrait de cette famille est qu'elle peut obtenir un rendement formidablement grand. Cependant, on ne peut utiliser ce genre de compression que pour des sons, vidéos et images, mais pas sur des fichiers ou sur du texte puisque ceux-ci ne doivent subir aucune dégradation [10].

Quelques exemples de techniques de compression avec perte sont les suivants :

- JPEG
- JPEG 2000

1.2.2.1 La compression JPEG

"Joint Photographic Expert Group" ou JPEG a été voté comme norme internationale en 1992 [10]. Ce standard de compression s'applique à des images couleur et/ou en niveaux de gris. Il a été appliqué notamment sur des images satellitaires, médicales,...La compression JPEG permet des taux de compression importants. Le processus de compression et de décompression JPEG irréversible comporte six étapes principales représentées (figure 1.2)

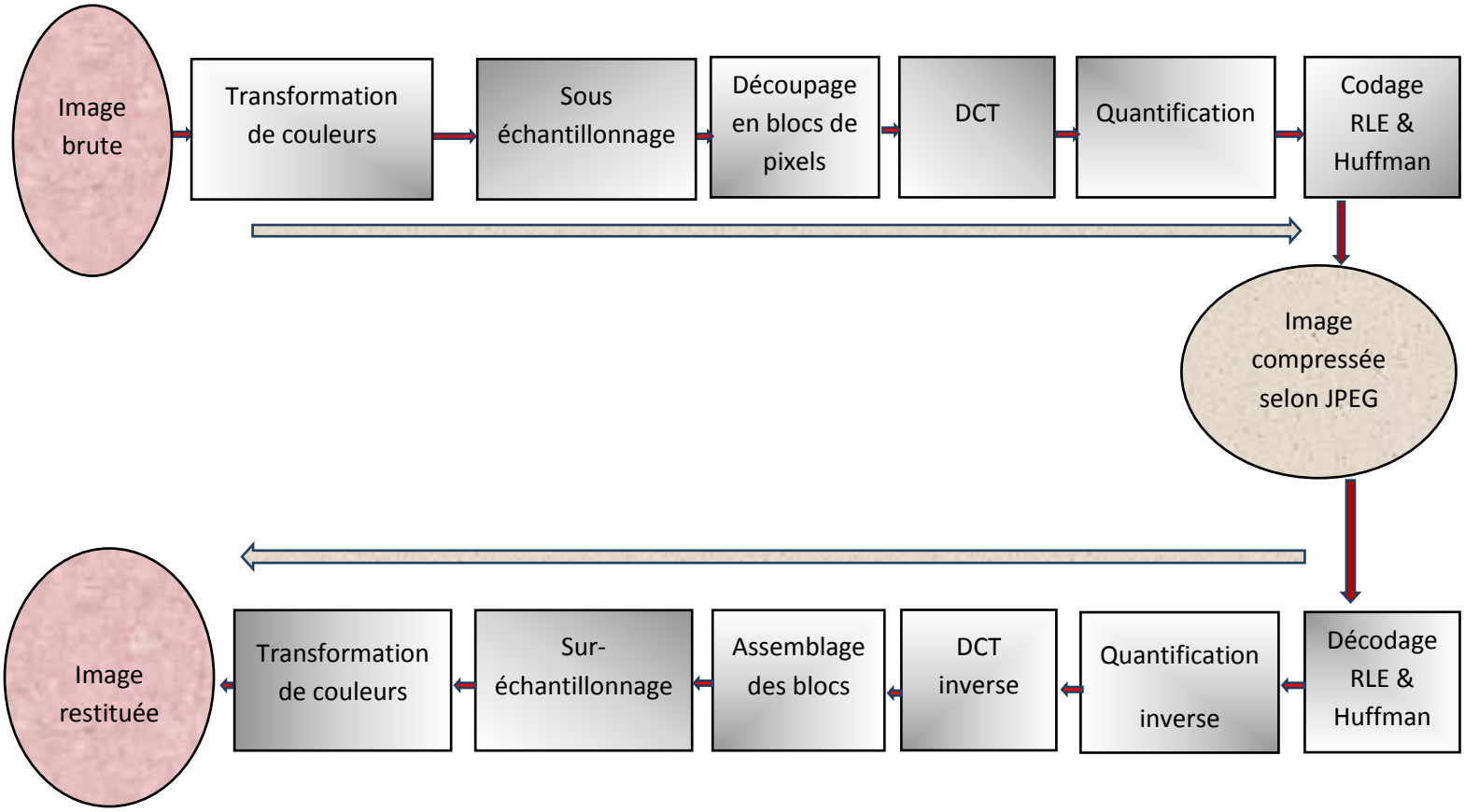

Figure 1. 2: Compression JPEG [11].

1. Passage du modèle initial des couleurs de l'image (souvent RVB) au modèle de type chrominance/luminance (YCrCb) [11].

Y est l'information de luminance, et Cb et Cr sont deux informations de chrominance. Cette étape va permettre de préparer la prochaine étape : le sous-échantillonnage.

2. Réduction de l'information de chrominance ; la vision humaine est sensible à la chrominance plus qu'à la chrominance.

3. Découpage de l'image en blocs de pixels : ces blocs sont des matrices de pixels de 8x8.
4. Application de la fonction DCT (Transformée en Cosinus Discrète)

   Cette fonction DCT est une transformation en série (Fourier), qui délivre une représentation non plus spatiale, mais dans le domaine fréquentiel.

   Cette étape consiste à appliquer à la matrice de pixels la formule mathématique de DCT afin d'obtenir une matrice des fréquences qui sera utilisée dans l'étape suivante.
5. Quantification : l'algorithme attribue à chaque fréquence, cellule de la matrice 8x8 pixels, un coefficient de perte. L'algorithme annulera ou diminuera les hautes fréquences qui, représentent les plus petits détails de l'image.

   L'atténuation de l'amplitude des différentes fréquences est déterminée par le ratio demandé par l'utilisateur.

Enfin, la matrice obtenue subit une représentation en zigzag, selon le codage RLE, et est compressée ensuite avec la méthode de Huffman.

6. Codage RLE (Run Length Encoding) :c'est un codage très simple. Il repose sur le fait que dans une image, il existe de nombreuses répétitions d'un même pixel, ou d'une même séquence de pixel, tous juxtaposés. Ainsi, au lieu de coder chaque pixel d'une image, le RLE propose de coder d'une part le nombre de répétitions, et d'autre part la séquence ou l'octet à répéter [10].

   Par exemple, pour la chaîne "MMMMMMMBBBBMMM", un codage RLE donnerait "7M4B3M"

Les étapes de la décompression s'effectuent dans l'ordre inverse de la compression suivant les méthodes définies précédemment [11].

1.2.2.2 La compression JPEG 2000

La norme JPEG 2000 définit le codage des images numériques et est destinée à supplanter la norme JPEG. L'extension des images au format JPEG 2000 est *.jp2* [10]. L'originalité du JPEG 2000 est qu'il permet de compresser les images avec ou sans perte d'informations. La norme ne décrit cependant que la méthode de décompression qui doit être utilisée pour les images de ce format. Dès lors, les

développeurs qui désirent implémenter un algorithme de compression conforme à JPEG 2000 procèdent à leur guise tant que leur méthode répond aux exigences de la norme.

La compression JPEG 2000 peut débuter par la transformation des trois composants colorimétriques de l'image en un coefficient de luminance et deux coefficients pour exprimer la couleur. Contrairement au JPEG, cette première étape de compression n'est pas obligatoire.

Contrairement au JPEG, le codeur JPEG 2000 utilise une transformation par ondelettes des pixels de l'image. Il s'agit d'une transformation des pixels de l'image en fréquences où chaque pixel correspond à une et une seule fréquence. Cette opération produit plusieurs sous-images par divisions successives de l'image source [10].Ces sous-images rassemblent chacune un intervalle de fréquences. Pour la majorité des images, les fréquences hautes sont moins nombreuses que les fréquences basses car les fréquences hautes signifient que les pixels de l'image sont très différents les uns des autres, un phénomène rare dans une image. Ensuite vient l'étape de quantification où se produit l'élimination des fréquences les plus hautes en fonction d'un taux de compression donné. Néanmoins, si la compression est non destructrice, l'étape de quantification n'est pas effectuée.

1.3 Paramètres de performances des méthodes de compression d'image

On retrouve dans les divers articles concernant la compression, des évaluations de performances difficilement comparables, dans la mesure où elles ne sont pas fondées sur les mêmes principes. Il semble donc utile de préciser les moyens de mesurer la compression qui sont retenus par les spécialistes [12].

Les principaux paramètres qui permettent d'évaluer une méthode de compression sont : le taux de compression, le débit, le temps d'exécution et les mesures de distorsions. En pratique, il s'agit de réaliser le meilleur compromis possible entre le taux de compression et la qualité de l'image décompressée tout en minimisant le temps d'exécution [10].De plus, l'algorithme permettant d'implémenter ces méthodes sur ordinateur, doit être robuste aux erreurs de transmission, flexible, et permettre

une transmission progressive, c'est à dire transmettre une image dont la qualité à la réception s'affine progressivement.

1.3.1 Taux de compression

Le taux de compression est défini comme le rapport entre le nombre total de bits nécessaires pour représenter l'information originale et le nombre total de bits du fichier binaire à stocker qui résulte de la méthode de compression :

$$\mathrm{RC}\ (\%) = \frac{\text{nombre de bits codés}}{\text{nombre de bits de l image originale}} \times 100 \qquad (1.1)$$

Dans la pratique, on utilise plutôt le débit pour mesurer le pouvoir de compactage d'une méthode. Le débit est exprimé en bits par pixel :

$$\mathrm{RC}\ (\mathrm{bpp}) = \frac{\text{nombre de bits codés}}{\text{nombre de bits de l image originale}} \ * Nb\ \ Bit \qquad (1.2)$$

bpp (bite par pixel)

1.3.2 Débit

Le débit désigne la quantité d'informations transférée en l'espace d'une seconde. L'unité de mesure est le bit par seconde (bit/s) et ses différents multiples (Kilo-bits par seconde, Mega-bits par seconde...).

Le choix du débit aura un impact direct sur la taille occupée par le fichier final et sur sa qualité. Plus le débit est important, plus le fichier final sera « lourd » et sa qualité meilleure.

1.3.3 Mesure de la distorsion

La distorsion (*D*) est l'erreur introduite par l'opération de compression. La mesure de distorsion utilisée généralement en compression d'image, est l'erreur quadratique moyenne *MSE* (*Mean Square Error*) [10].

$$MSE = \frac{1}{N}\sum_{p\in P}(X_p - \hat{X}_p)^2 \qquad (1.3)$$

Où P désigne l'ensemble des N pixels de l'image, et X_p et$\hat{X}_p$ sont respectivement les amplitudes des pixels sur les images originale et reconstruite. L'utilisation de cette mesure d'erreur est préférée dans le traitement des données, puisqu'il est possible de développer des algorithmes qui la minimisent. Cependant, la distorsion des images comprimées est parfois mesurée par d'autres valeurs, comme l'erreur absolue moyenne ou l'erreur absolue maximale. Cette dernière semble plutôt trop sensible aux imperfections occasionnelles. Il est aussi vraisemblable que l'œil tient beaucoup plus compte des erreurs à grandes amplitudes, ce qui favorise la mesure quadratique. Cependant, l'ordonnance des images basée sur ces mesures est en général le même [13]

Au lieu de communiquer les valeurs de MSE, les études donnent en général le rapport crête signal sur bruit (*Peak Signal to Noise Ratio*, PSNR). Au lieu de mesurer la distorsion, cette valeur mesure la fidélité de la compression, puisqu'elle est proportionnelle à la qualité. Tout de même, elle est une fonction de MSE ; sa définition et son utilisation proviennent du domaine des signaux audio/vidéo :

$$\text{PSNR}=10log_{10}\frac{X^2max}{MSE} \quad (1.4)$$

Où X_{max} désigne la luminance maximale possible. Une valeur de PSNR égale à ∞ correspond à une image parfaitement reconstruite, et elle décroît en fonction de la distorsion. Le PSNR relie donc l'erreur quadratique moyenne à l'énergie maximale de l'image [13].

1.3.4 Le temps d'exécution

La contrainte du temps est un facteur essentiel dans l'évaluation des performances de toute méthode de compression, elle revient à calculer le temps pris par la compression et la décompression des images. Cette contrainte est plus au moins imposée selon l'application visée par la compression (transmission ou archivage). En effet, il serait dommage, dans une application de transmission, que le temps gagné par une réduction de la taille des données à transmettre soit inférieur au temps passé à la

compression décompression [12]. Cette qualité sera cependant moins cruciale dans des applications visant l'archivage de données.

1.4 Conclusion

Nous avons commencé ce chapitre par une présentation des problèmes de taille rencontrées lors de la manipulation de l'image et le besoin de compression comme solution à ces problèmes.

Ensuite nous avons essayé de faire un récapitulatif sur les notions élémentaires de la compression d'images en présentant les deux types de compression et citer quelques techniques, sans oublier de parler sur les paramètres de performance qui servent à évaluer ces techniques.

D'autre part, pour garder la confidentialité des informations représentées dans les images nous avons recours à la cryptographie. Dans le chapitre suivant nous allons décrire les principes ainsi que les méthodes relevant de la cryptographie.

Chapitre 2 Techniques de Cryptologie

2.1 Introduction

Les réseaux de communication numériques sont largement utilisés pour l'échange des informations (texte, audio, image, vidéo, …etc.). La sécurité de ces informations échangées est devenue une nécessité primordiale dans beaucoup d'applications des organismes civils ou militaires, citons par exemple, l'internet, la téléphonie mobile, les distributeurs de billets, les abonnements aux chaînes de télévision payantes, le commerce électronique et les cartes à puce, afin d'assurer la confidentialité et d'empêcher toute modification ou exploitation non autorisée des données. L'une des méthodes connues pour la réalisation efficace de cet objectif est le cryptage qui rend l'information complètement ou partiellement illisible et incompréhensible

2.2 Terminologie et bref historique

L'origine de la cryptologie mot réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments : «cryptos », qui signifie caché et « logos » qui signifie mot. La cryptologie est aussi vieille que l'écriture elle-même, et a été utilisée depuis des milliers d'années pour assurer les communications militaires et diplomatiques. Par exemple, le célèbre empereur romain Jule César utilisait un algorithme de chiffrement pour protéger les messages à ses troupes. Dans le domaine de l'un de cryptologie on peut voir deux visions : la cryptographie et la cryptanalyse [14]. Donnons dans ce qui suit quelques définitions [15] :

- Cryptographie : étymologiquement elle correspond à « écriture secrète ». C'est l'´étude des techniques et des pratiques de chiffrement qui assurent l'inviolabilité des textes et des données.
- Chiffrement : transformation à l'aide d'une clé d'un message en clair (dit texte clair) en un message incompréhensible (dit texte chiffré ou cryptogramme).

- Crypto système : C'est le système consistant en un algorithme de chiffrage et d'un algorithme de déchiffrage. Il permet de chiffrer un message clair, en un message chiffré non clair ; le déchiffrage permet par la suite de retourner au message original.
- Décryptage : retrouver le message clair correspondant à un message chiffré sans posséder la clé de déchiffrement.
- Cryptanalyse : science analysant les cryptogrammes en vue de les décrypter (cassage sans clé).

2.3 Cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité [16].

2.3.1 Usage de la cryptographie

La cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité

2.3.2 La confidentialité

Elle consiste à rendre l'information intelligible à d'autres personnes que les acteurs de la transaction

2.3.3 L'intégrité

Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication.

2.3.4 L'authentification

Elle consiste à assurer l'identité d'un utilisateur, c.-à-d. de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès

peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

2.3.5 Le non répudiation

De l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

2.4 Les méthodes de cryptages

Il n'existe en réalité que deux façons de crypter [14] :

- La méthode dite symétrique utilise la même clé pour chiffrer et déchiffrer.
- La méthode asymétrique utilise une clé publique pour chiffrer, et une clé publique pour déchiffrer.

2.4.1 Le chiffrement symétrique

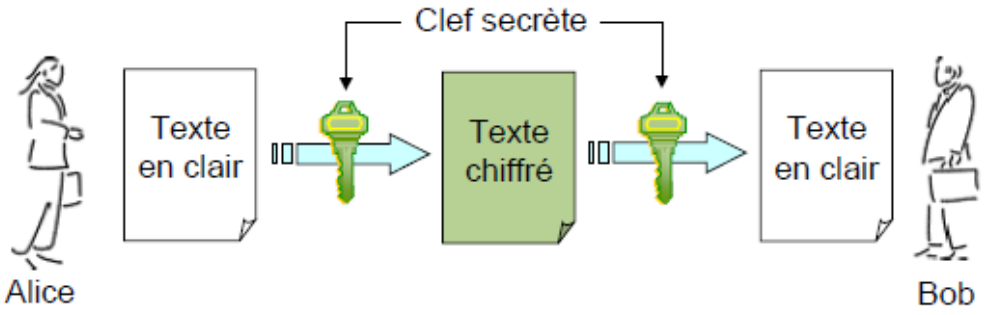

Figure 2. 1: Principe du chiffrement symétrique [17].

2.4.1.1Principe

L'émetteur et le récepteur partagent la même clé secrète. Autrement dit, avec la même clé on peut chiffrer et déchiffrer les messages

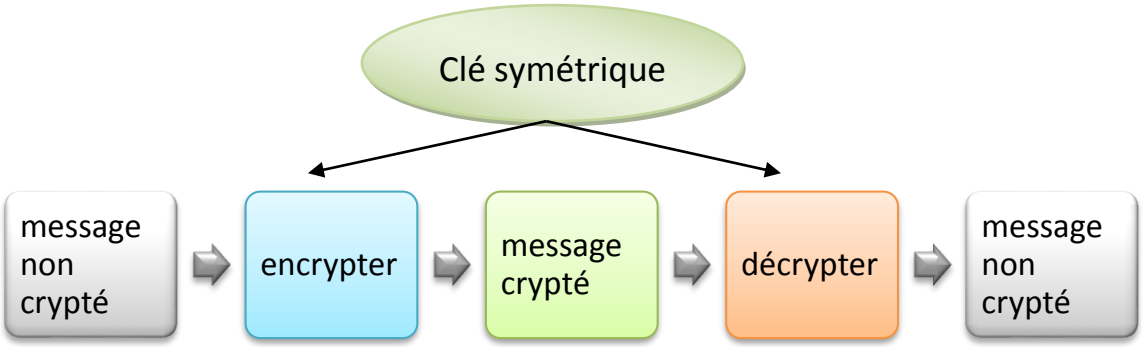

Figure 2. 2: Système de cryptage / décryptage [18].

2.4.1.2Les types d'algorithmes symétriques

Il existe deux types d'algorithmes symétriques :

- Les algorithmes de chiffrement en continu, qui agissent sur le message en clair un bit à la fois.
- Les algorithmes de chiffrement par bloc, qui opèrent sur le message en clair par groupes de bits appelés blocs.

1. Algorithmes de chiffrement en continu

Qui opèrent sur le message en clair un bit à la fois. Le principe consiste à générer un flux pseudo aléatoire et de le combiner avec l'information bit à bit par l'opération XOR.[19]

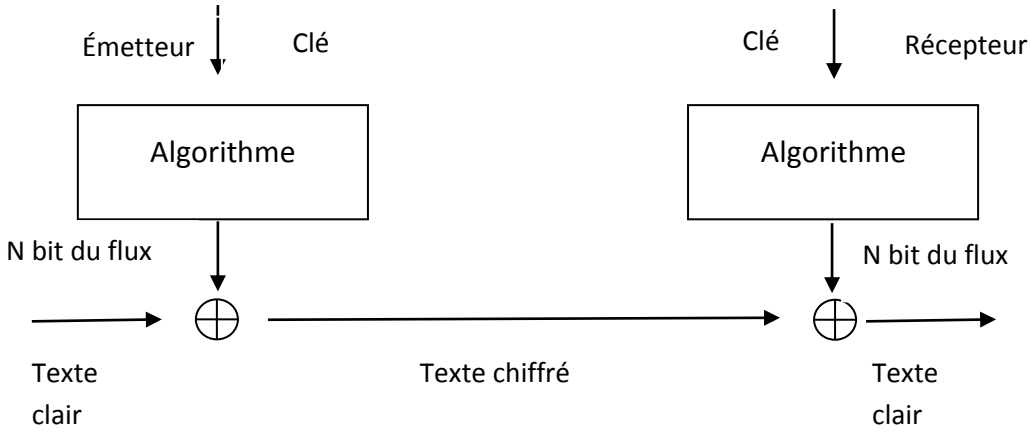

***Figure 2. 3*: Schéma chiffrement AES [19]**

A la réception, on applique le même mécanisme, et on restitue l'information.

2. Algorithmes de chiffrement par bloc

Ils opèrent sur le message en clair par groupe de bits. La taille typique des blocs est 64 bits, ce qui est assez grand pour interdire l'analyse et assez petit pour être pratique [20].

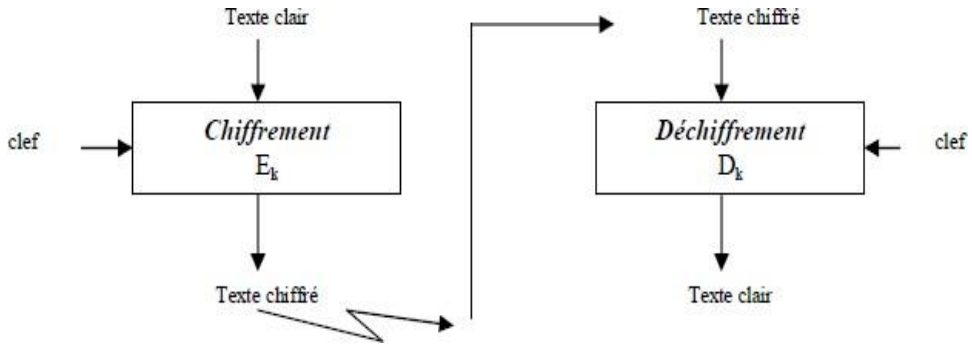

Figure 2. 4 : Chiffrement par Bloc [19]

2.4.1.3 Principaux algorithmes à clé privée

Parmi les algorithmes à clé privée, nous citons :

1. **DES (Data Encryption Standard)**

Jusqu'à très récemment, le système de chiffrement à clef secrète le plus utilisé était le DES [20]. Le DES opère sur des blocs de 64 bits et utilise une clef secrète de 56 bits. Il est vulnérable aux attaques exhaustives. C'est pourquoi la plus part des applications l'utilise maintenant sous la forme d'un triple DES, deux clefs constituent de trois chiffrements DES successifs avec deux clefs secrètes. Cette technique permet de doubler la taille de la clef secrète (112 bits).Plus précisément, pour chiffrer avec le triple DES, on effectue d'abord un chiffrement DES paramétré par une première clef de 56 bits, puis un déchiffrement DES paramétré par une seconde clef et à nouveau un chiffrement DES avec la première clef. Seules deux clefs sont utilisées dans la mesure où l'emploi de trois clefs secrètes différentes ne permet pas d'accroitre la sécurité de l'algorithme.

2. IDEA (INTERNATIONAL DATA ENCRYPTIONALGORITHM)

Plus récent que le DES, l'IDEA, contrairement aux autres algorithmes de codage, a été breveté par la société suisse Ascom. L'utilisation non commerciale de cet algorithme est cependant permise sous réserve d'une autorisation d'Ascom. Cet algorithme opère sur des blocs de 64 bits, mais utilise une clef de 128bits qui sera transformées en 52 blocs de 16 bits. DEA est considéré comme étant assez nettement supérieur au DES en terme de sécurité.sa vitesse d'exécution reste comparable avec le DES. Ses implémentations hardware sont simplement légèrement plus rapides [20].

3. AES (ADVANCEDENCRYPTIONSTANDARD)

L'algorithme de chiffrement AES (Advanced Encryption Standard) est le système standard de chiffrement par bloc et a pour objectif de remplacer le DES qui devient vulnérable. Le nombre de rondes de l'algorithme AES dépend de la taille de la clef et de la taille des blocs de données. Par exemple, le nombre de rondes est 9 si les blocs et la clef sont de longueur 128 bits. Pour crypter un bloc de données avec AES, il faut d'abord effectuer l'étape nommée AddRoundKey qui consiste à appliquer un « OU exclusif » (XOR) entre une sous-clé et le bloc. Après, nous entrons dans l'opération d'une ronde. Chaque opération régulière de ronde implique quatre étapes. La première est l'étape nommée SubByte, où chaque octet du bloc est remplacé par une autre valeur issue d'une S-box. La seconde étape est l'étape nommée ShiftRow où les lignes sont décalées cycliquement avec différents offsets. Dans la troisième étape, nommée MixColumn, chaque colonne est traitée comme un polynôme, multipliée sur GF(28) (Galois Field) par une matrice. La dernière étape d'une ronde est à nouveau l'étape nommée AddRoundKey, qui est un simple « OU exclusif » entre la donnée actuelle et la sous-clé de la ronde courante [21].

L'algorithme AES effectue une routine supplémentaire finale qui est composée des étapes SubByte, ShiftRow et AddRoundKey avant de produire le chiffrement final.

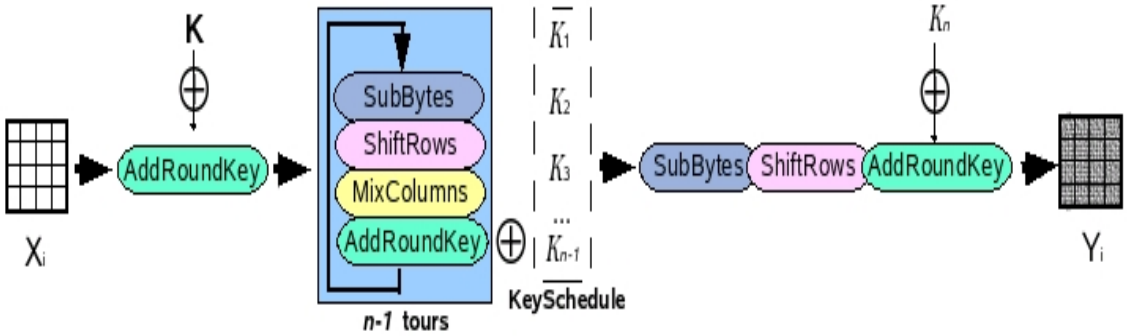

Figure 2. 5 : les étapes AES

2.4.1.4La faiblesse du système symétrique

Un système symétrique peut être d'une grande robustesse, il faut seulement transmettre la clé de façon sécurisée. Le seul moyen certain de transmettre la clé de façon sécurisée, c'est de se connecter sur un canal sécurisé au préalable, ou d'échanger une clé de façon physique [22].

2.4.2 Le chiffrement asymétrique

2.4.2.1 Principe

Tous les algorithmes évoqués jusqu'à présent sont symétriques en ce sens que la même clef est utilisée pour le chiffrement et le déchiffrement. Le problème essentiel de la cryptographie symétrique est la distribution des clefs : pour que n personnes puissent communiquer de manière confidentielle il faut n (n-1)/2 clefs**.**

L'idée de base des crypto-systèmes à clefs publiques a été proposée dans un article fondamental de Diffie et Hellman en 1976. Le principe fondamental est d'utiliser des clefs de chiffrement et déchiffrement différentes, non reconstructibles l'une à partir de l'autre :

- une clef publique pour le chiffrement
- une clef secrète pour le déchiffrement

Ce système est basé sur une fonction à sens unique, soit une fonction facile à calculer dans un sens mais très difficile à inverser sans la clef privée.

Pour faire une explication imagée, la clef publique joue le rôle d'un cadenas. Imaginons que seul Bob possède la clef (clef secrète), Alice enferme son message dans une boîte à l'aide du cadenas et l'envoie à Bob. Personne n'est en mesure de lire le message puisque seul Bob possède la clef du cadenas [22].

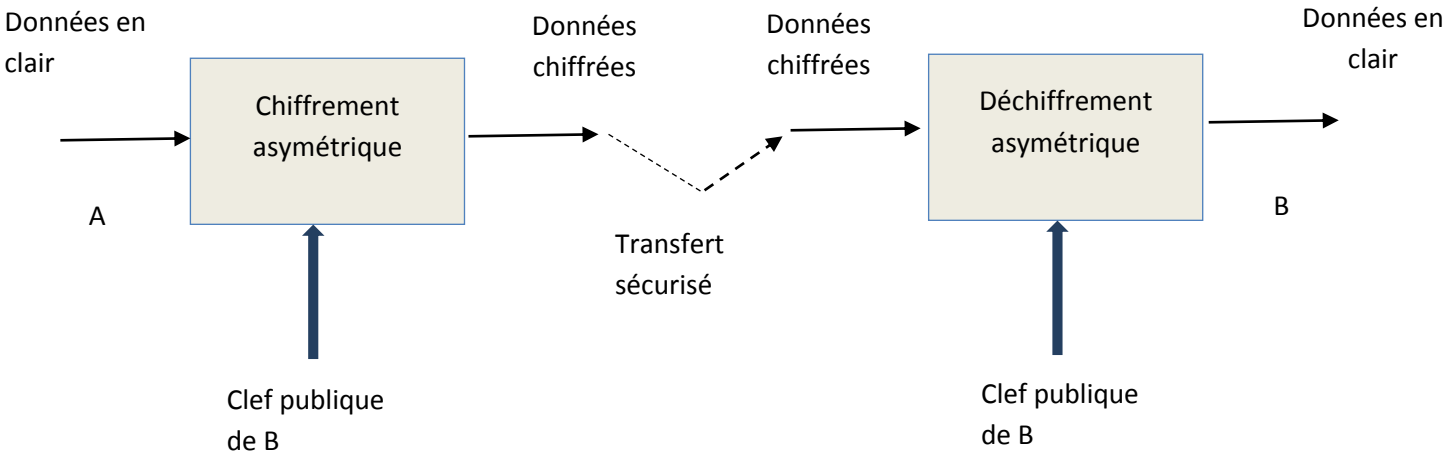

Figure 2. 6 : chiffrement-déchiffrement asymétrique [22]

2.4.2.2Les principaux algorithmes à clé publique

Parmi les principaux algorithmes à clé publique, citons :

- Diffie et Hellman

C'est en 1976 que Whitfield Diffie et Martin Hellman, de l'Université Stanford, proposent un principe de chiffrement entièrement nouveau : la cryptographie à clé publique, ou asymétrique. Expliquons leur procédé de façon imagée :

Alice doit recevoir un message de Bob, mais elle ne fait pas confiance au facteur qui pourrait ouvrir sa lettre. Comment peut-elle être sûre de recevoir ce message sans qu'il soit lu ?

Alice va d'abord envoyer à Bob un cadenas ouvert, dont elle seule possède la clé. Ensuite, Bob va placer son message dans une boîte, qu'il fermera à l'aide de ce

cadenas, avant de l'envoyer à Alice. Le facteur ne pourra donc pas ouvrir la boîte, puisque seule Alice possède la clé

Ainsi, un système cryptographie à clé publique est en fait basé sur deux clés

- Une clé publique, pouvant être distribuée librement, c'est le cadenas ouvert
- Une clé secrète, connue uniquement du receveur, c'est le cadenas fermé

C'est la raison pour laquelle on parle de chiffrement asymétrique [23].

En résumé, on dispose d'une fonction P sur les entiers, qui possède un inverse S. On suppose qu'on peut fabriquer un tel couple (P, S), mais ne connaissant uniquement que P, il est impossible (ou au moins très difficile) de retrouver S. Autrement dit, il faut déterminer mathématiquement des fonctions difficilement inversibles, ou "à sens unique"

Trouver de telles fonctions semblait ardu aux mathématiciens. En effet, comment imaginer une fonction qui soit à sens unique pour tout le monde, excepté pour son créateur qui peut l'inverser grâce à la connaissance d'une information particulière (la clé) ? Ce sont Diffie et Hellman qui ont les premiers à donner une réponse à cette question

- ❖ Protocole de Diffie et Hellman

Parallèlement à leur principe de cryptographie à clé publique, Diffie et Hellman ont proposé un protocole d'échanges de clés totalement sécurisé, basé sur des fonctions difficiles à inverser [23].

- ➢ **RSA**

L'algorithme le plus célèbre d'algorithme à clef publique a été inventé en 1977 par RonRivest, Adi Shamir et Len Adleman, à la suite de la publication de l'idée d'une cryptographie à clef publique par Diffie et Hellman. Il fut appelé RSA, des initiales de ces inventeurs.

RSA est basé sur la difficulté de factoriser un grand nombre en produit de deux grands facteurs premiers. L'algorithme fonctionne de la manière suivante :
Imaginons que Bob souhaite recevoir d'Alice des messages en utilisant RSA [22].

❖ **génération des clefs**

- p et q, deux grands nombres premiers sont générés au hasard grâce à un algorithme de test de primalité probabiliste, avec n = p×q.
- Un nombre entier e premier avec (p-1) × (q-1) est choisi. Deux nombres sont premiers entre eux s'ils n'ont pas d'autre facteur commun que 1.
- L'entier d est l'entier de l'intervalle [2, (p-1) × (q-1) [tel que ed soit congrue à 1

Modulo (p-1) (q-1), c'est-à-dire tel que ed-1 soit un multiple de (p-1) (q-1) [24].

❖ **distribution des clefs**

Le couple (n, e) constitue la clef publique de Bob. Il la rend disponible à Alice en lui envoyant ou en la mettant dans un annuaire. Le couple (n, d) constitue quant à lui sa clef privée.

❖ **chiffrement du message**

Pour crypter le message Alice représente le message sous la forme d'un ou plusieurs entiers M compris entre 0 et n-1. Elle calcule C= M^emod n grâce à la clef publique (n, e) de Bob et envoie C à Bob.

❖ **déchiffrement du message**

Bob reçoit C et calcule grâce à sa clef privée C^d mod n.Il obtient ainsi le message initial M [22].

2.4.2.3 La faiblesse du système asymétrique

Le risque principal dans l'utilisation des clés asymétriques est celui de l'attaque de l'homme du milieu, c'est-à-dire la possibilité qu'une partie adverse intercepte les clés publiques échangées pour les remplacer par les siennes [25]. Il pourrait alors déchiffrer et signer tous les messages échangés [26].

2.4.3 Comparaison entre le cryptage symétrique et asymétrique

Cryptage symétrique	*Cryptage asymétrique*
• Chiffrement à clé privé (une seule clé est utilisée pour le cryptage et le décryptage).	• Chiffrement à clé publique (utilisation de la clé publique pour le cryptage et la clé privé pour le décryptage).
• Très facile.	• Difficile par rapport au cryptage symétrique.
• Très rapide.	• Plus lent.
• Les clés de chiffrement symétrique doivent être conservées en toute sécurité.	• les clés publiques qu'ils utilisent sont sans danger pour être publié n'importe où parce que pour obtenir la clé privée à partir d'une clé publique peut prendre de très longues durées de travail.

Tableau 2.1 : cryptage symétrique VS cryptage asymétrique

2.5 Conclusion

Dans ce chapitre, nous avons présenté une introduction générale sur la cryptographie, nous avons distingué deux classes importantes des méthodes de chiffrement, c'est le cryptage symétrique à clé secrète et le cryptage asymétrique à clé publique. Nous avons aussi montré la puissance et la faiblesse de chaque type d'algorithme de chiffrement.

Chapitre 3 Proposition d'une architecture hybride

3.1 Introduction

Les applications qui utilisent des données multimédias, comme les images, sont de plus en plus présentes dans notre vie quotidienne. Ainsi manipuler les images (stocker ou transmettre,…) devient un enjeu stratégique. De plus, la protection de ces données est devenue à son tour un domaine attirant pour les chercheurs afin de préserver la confidentialité de ces données. Le volume grandissant de ces données nécessite un temps de calcul de plus en plus grand, ce qui a amené les chercheurs à développer des techniques de compression et de cryptage dédiées à une application donnée et qui soient simples, rapides et efficaces.

3.2 Compression d'image sans cryptage

Comme on a vu précédemment, la compression est une opération qui consiste à réduire les informations utilisées pour représenter une image sans l'altérer ou avec une dégradation contrôlée de la qualité de cette dernière lors de sa reconstruction. Dans la littérature, il existe deux types de compression avec ou sans pertes d'information.

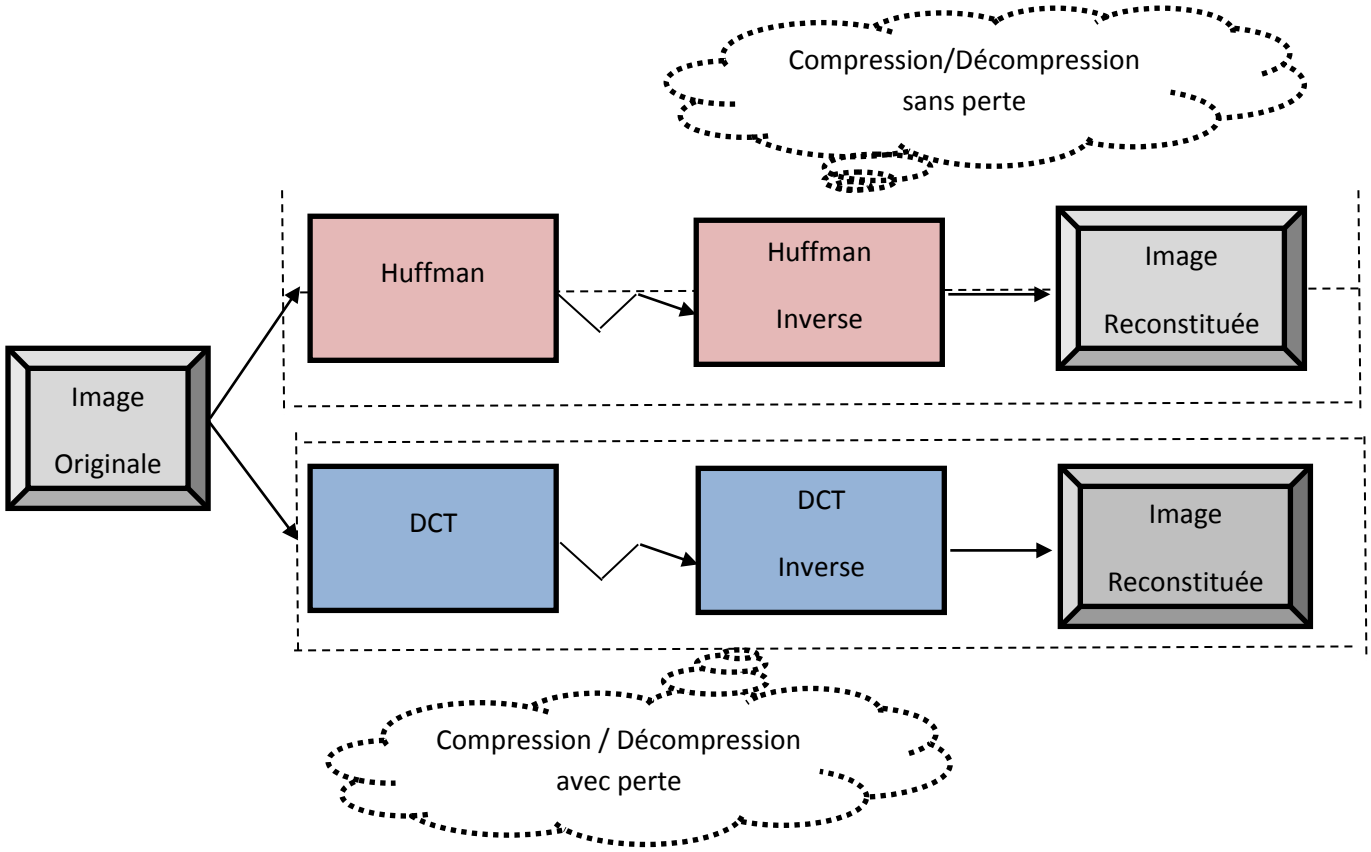

Figure 3. 1 : Compression d'image (avec/sans) perte sans chiffrement

3.2.1 Compression d'image sans perte

Le codage de Huffman est la technique la plus populaire pour supprimer la redondance de codage [27]. Le codage Huffman commence par calculer la probabilité de chaque symbole dans l'image. Les probabilités des symboles sont disposées dans un ordre décroissant formant des nœuds feuilles d'un arbre. Lorsque les symboles sont codés individuellement, le code Huffman est conçu en fusionnant les symboles probables les plus bas et ce processus est répété jusqu'à ce qu'il ne reste plus que deux probabilités de deux symboles composés. Ainsi, un arbre de code est généré et les codes Huffman sont obtenus à partir de l'étiquetage de l'arbre de code.

a L'algorithme de HUFFMAN dynamique

a.1 codage

```
// Table d'occurrences
// Actuel arbre -> table de correspondance des symboles
Arbre A;
// On lit le premier caractère
// Tant que la chaine n'est pas finie, on calcule...
Tant Que(C!=eof) {
  // Si ce caractère est connu, on renvoie son code (stocké dans l'arbre)
  Si (NbOccurences[C]!=0) {
EcrireCode(C, A);
  } Sinon {
    // Sinon, on insère un code symbolique 'phi' suivi du code du caractère
EcrireCode('phi', A);
    // indique la présence d'un nouveau caractèresupplémentaire
NbOccurences[phi]+=1;
EcrireCode(C, A);}
NbOccurrence[C]+=1;
  // A chaque itération on recalcule l'arbre
MettreAJour(A);
  C=LireCaractere();}
```

On remarque qu'à l'étape n, l'arbre est identique à celui qu'on aurait construit en utilisant HUFFMAN statique avec le message contenant les n premiers symboles du message complet. Le symbole 'phi' (que nous notons dorénavant Ø) est un caractère symbolique signifiant l'initialisation (il est donc codé avec un code prédéfini, par exemple '0'). À chaque fois que l'on rencontre un nouveau caractère, on insère le code

de suivi Ø des n bits du nouveau caractère. Celui-ci est ensuite inséré dans l'arbre de HUFFMAN.

a.2 Décodage

```
// Table d'occurrences
intNbOccurences
// Actuel arbre -> table de correspondance des symbolesArbreA;
// On lit le premier caractèreC=LireSuiteCode(Entree);
// Tant que la chaine n'est pas finie, on calcule...
// Nota Bene: a l'initialisation, le seul code connu est 'phi'TantQue(C!=eof) {
  // On utilise la propriété du préfixe: on retire de C un code unique B
  // représente par les k premiers bits de C  B=CodeAssocie(C);
  // Ce caractère est forcement connu, on renvoie son code (stocke dans l'arbre)
  D=Décoder(B, A);Si (D=='phi') {
    // On lit (et écrit) les n bits associes au nouveau caractère
    // L'index sur C avance donc de n bitsLireEcrire(C);} Sinon {
NbOccurences[phi]+=1Ecrire(D);}
NbOccurrence[C]+=1;
  // A chaque itération on recalcule l'arbreMettreAJour(A);
  C=LireSuiteCode();}
```

Cette méthode dynamique est plus rapide et donne souvent de meilleurs résultats (à cause du stockage de l'arbre qui n'existe pas puisqu'il est construit au fur et à mesure).

3.2.2 Compression d'images avec perte

a DCT

Le passage par la DCT a été l'idée majeure pour la compression JPEG. En effet ce processus appartient à une classe d'opérations mathématiques, tout comme la Transformée de Fourier. Elle permet un changement de domaine, tout en gardant

exactement la même fonction étudiée. Dans notre cas, on étudie une image, c'est à dire une fonction à 3 dimensions : X et Y, indiquant le pixel, et Z avec la valeur du pixel en ce point. Dans le cas d'une image couleur, il faut donc considérer indépendamment trois fonctions, pour chacun des canaux RGB [28].

a.3 Algorithme de DCT

Pos=1

***W*=**Génération_de_valeurs_de_poids_aléatoires **() ;**

I=1 ;

Tant que (I<taille de ligne LL2)

J=1 ;

Tant que (J<taille de colonne LL2)

%% Appliquer la DCT pour chaque bloc

Block2x2=**Appliquer_DCT**(LL2[I, J]);

%% convertit immédiatement un bloc en un vecteur 1*4

L1x4=**Convertion_Block_en_tableau**(Bloc) ;

DC [*Pos*]=L1x4 [1] ; %% stockerla première valeurdans la Colonne-DC

Arr [*Pos*]=0 ; %% initialisation avant l'addition

pour K=2 à 4

Arr [*Pos*]= Arr [*Pos*] + L1x4 [*K*] * *W* (*K*-1) ;FIN **; %%** *fin pour*

Pos++ ;

J=J+2 ;

Fin **; %%** *fin tant que*

I=I+2 ;

Fin ; *%% fin tant que*

Taille bloc =8; %% Bloc de taille 8x8

I=1;*LOC*=1

Tant que (I< taille de la colonne de LH2)

J=1;

Tant que (J< taille de la ligne de LH2)

%% lire un bloc 8x8 des hautes frequencies de la sous-bande

*Block[1..taille bloc*taille bloc]*=**Lire_Bloc_de_la_Matrice**(I,J);

%% cette fonctioncoche le contenu de la case "Bloquer", a-t-il unevaleur non nulle?

Si (**Cocher_Bloc**(*Bloc*, *taille bloc*) = = 'NO')

*POSTION [LOC] =I; POSTION [LOC+*1*] =J;*

%% Enregistrer le localisationd'originedubloccontenant les donnéesnon nulles

*LOC=LOC+*2;

%% Transférer le contenu du bloc à un tableau unidimensionnelle

Pour*K=1:taille_bloc*taille_bloc*

Tableau_réduit[P]= Bloc[k];

++P;

Fin*;*

Fin;

J=J+ taille_bloc;

Fin;

I=I+ taille_bloc;

Fin;

3.3 Le chiffrement

La cryptographie moderne s'attaque en fait plus généralement aux problèmes de sécurité des communications. Le but est d'offrir un certain nombre de services de sécurité comme la confidentialité, l'intégrité, l'authentification des données transmises. La confidentialité est historiquement le premier problème posé à la cryptographie (comme dans les systèmes bancaires, de télécommunications ou militaires). Il se résout par la notion de chiffrement [29].

3.3.1 AES

L'AES opère sur des blocs de 128 bits (plaintext P) qu'il transforme en blocs cryptés de 128 bits (C) par une séquence de Nr opérations ou "rounds", à partir d'une clé de 128, 192 ou 256 bits. Suivant la taille de celle-ci, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds.

1. Choix de l'AES

Le choix de cet algorithme répond à de nombreux critères plus généraux dont nous pouvons citer les suivants :

- ✓ sécurité ou l'effort requis pour une éventuelle cryptanalyse.
- ✓ facilité de calcul : cela entraine une grande rapidité de traitement
- ✓ besoins en ressources et mémoire très faibles
- ✓ flexibilité d'implémentation : cela inclut une grande variété de plateformes et d'applications ainsi que des tailles de clés et de blocs supplémentaires
- ✓ hardware et software : il est possible d'implémenter l'AES aussi bien sous forme logicielle que matérielle (câblé)
- ✓ simplicité : le design de l'AES est relativement simple [30].

2. Principe de fonctionnement

Le schéma suivant (figure 3.2) décrit succinctement le déroulement du chiffrement :

- BYTE_SUB (Byte Substitution) est une fonction non-linéaire opérant indépendamment sur chaque bloc à partir d'une table dite de substitution.
- SHIFT_ROW est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opère des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).
- MIX_COL est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit matriciel sur le corps de Galois (28).
- Le + entouré d'un cercle désigne l'opération de OU exclusif (XOR).

- Ki est la nième sous-clé calculée par un algorithme à partir de la clé principale K. Le déchiffrement consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous-clés.

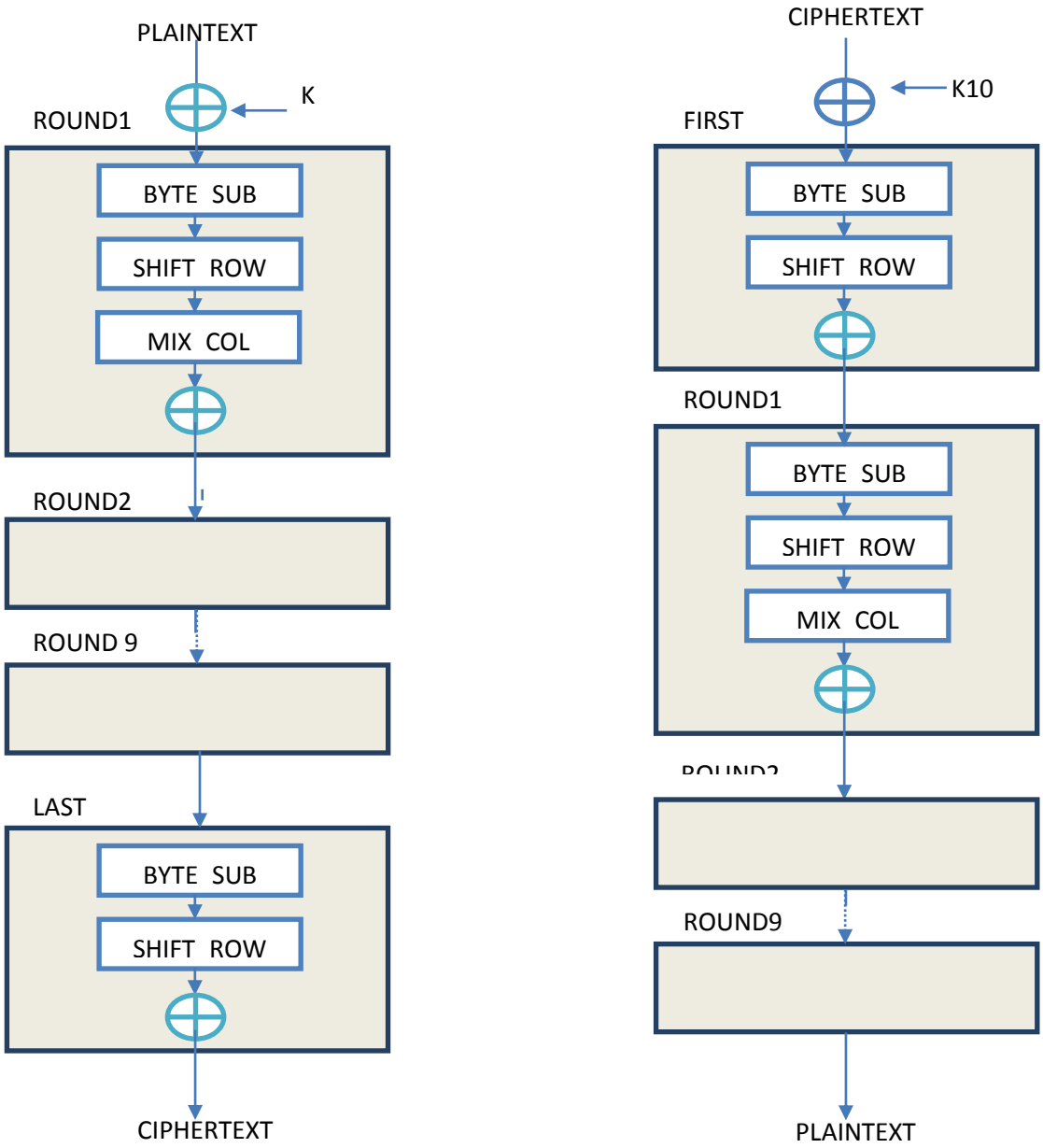

Figure 3. 2 : principe de l'algorithme AES

3.4 Compression d'image avec chiffrement

L'usage excessif des réseaux informatiques pour le transfert des données doit évidemment obéir à un double objectif : la réduction du volume des données afin de désencombrer le maximum possible les réseaux publics de communication et la confidentialité en vue de garantir un niveau de sécurité optimum.

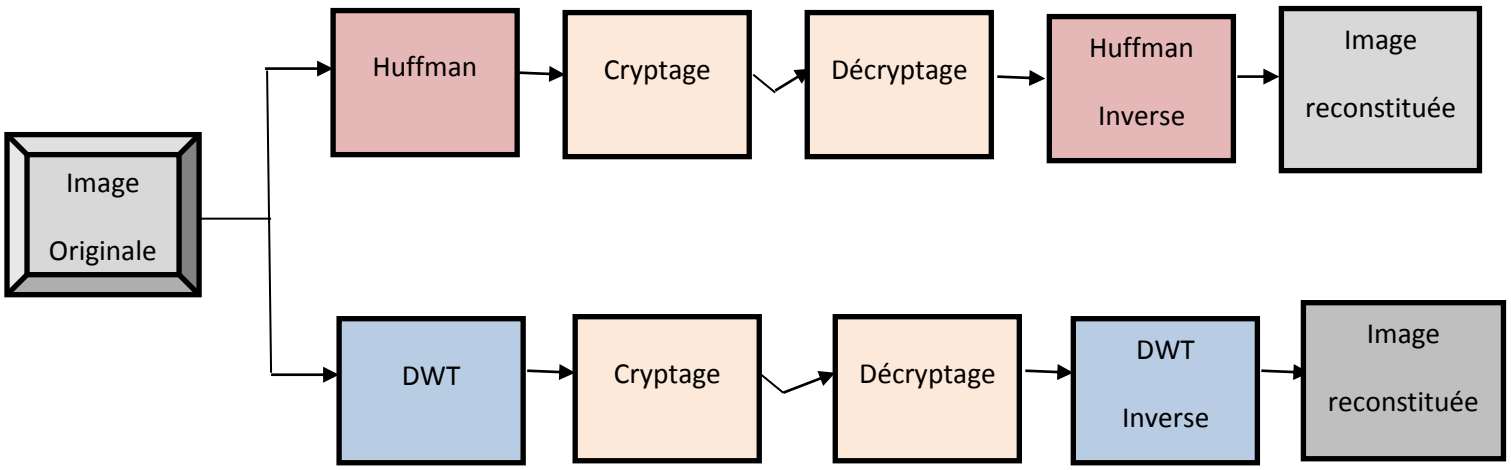

Figure 3. 3: compression (avec/sans) perte avec chiffrement

Le cryptage permet de modifier l'information source suivant un algorithme bien défini et de façon complexe afin qu'elle devienne inintelligible, sauf pour le destinataire autorisé à lire le message, donc seul à connaître les règles utilisées pour le décryptage.

3.4.1 Compression d'un système hybride (DWT-Huffman)

La Transformée en Ondelettes est devenue en quelques années un sujet de recherche très débattu. On ne compte plus aujourd'hui les applications qui utilisent cette technique. Il s'agit d'un algorithme permettant de calculer une représentation d'un signal en bandes de fréquences indépendantes. Cette représentation est particulièrement utile pour le traitement d'images.

Dans cette partie, on va fusionner la transformée en ondelettes discrètes avec le codage de Huffman.

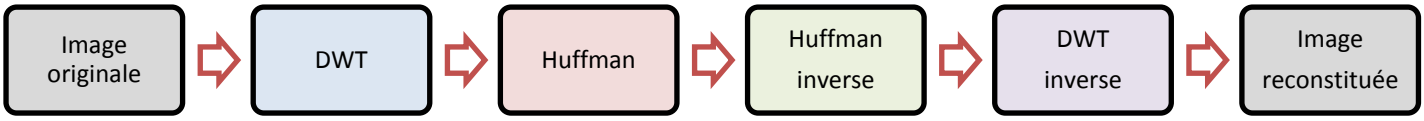

Figure 3. 4: Compression d'une image à l'aide du système hybride (DWT-Huffman)

a La transformée en ondelettes discrètes (DWT)

Le principe de l'algorithme consiste à diviser l'image en quatre à chaque itération : trois blocs concernant les détails de l'image, et le quatrième correspondant aux informations les plus importantes pour l'œil (les basses fréquences), qui sert de base pour la prochaine itération. Pour décomposer cette image, on utilise donc deux filtres issus du choix d'ondelette : un filtre passe-haut et un filtre passe-bas.

A partir de ces ondelettes, nous formons donc les deux filtres : nous noterons H le filtre passe-haut, et L le filtre passe-bas. Cette phase s'appelle la phase d'analyse.

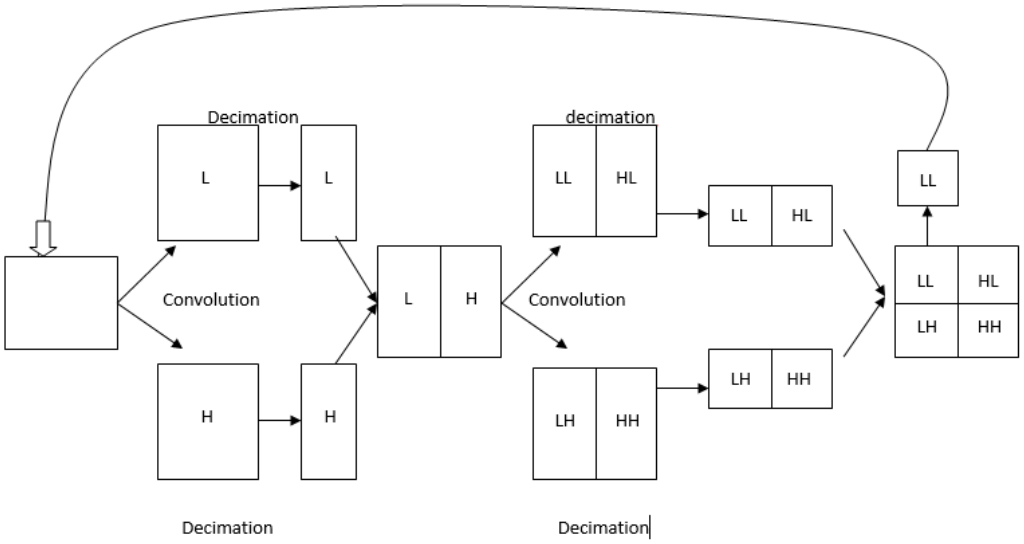

Figure 3. 5: Principe de la DWT

La ligne du haut correspond aux images convolées avec le filtre L. La ligne du bas correspond aux images convoluées avec le filtre H.

L'itération suivante se fera en prenant pour l'image de base, la partie LL, correspondant à la convolution par le filtre passe-bas (horizontalement, et verticalement). Le format JPEG2000 limite le nombre d'itérations D entre 0 et 32, les

valeurs par défaut étant la plupart du temps entre 4 et 8. Si on regarde plus généralement l'effet de cette transformation, il apparait que cet algorithme concentre l'énergie de l'image dans les blocs LL de plus haut niveau de décomposition. Ensuite, tous les autres blocs ne sont que des détails de l'image. Ainsi des manières de compresser et donc de mettre à zéro une partie des coefficients de cette matrice reviennent à analyser les blocs de plus haut niveau de décomposition, et de faire une supposition : les valeurs faibles dans les hauts degrés de décomposition tendent vers zéro au fur et à mesure que l'on remonte les niveaux.

Cette méthode permettra de compresser énormément les zones relativement continues, mais en gardant l'ensemble des discontinuités, et cela en suivant le contour de l'image, puisque les filtres passe-haut sont appliqués dans toutes les directions : verticale, horizontale et diagonale (composition des deux).

La figure suivante résume le principe de fonctionnement.

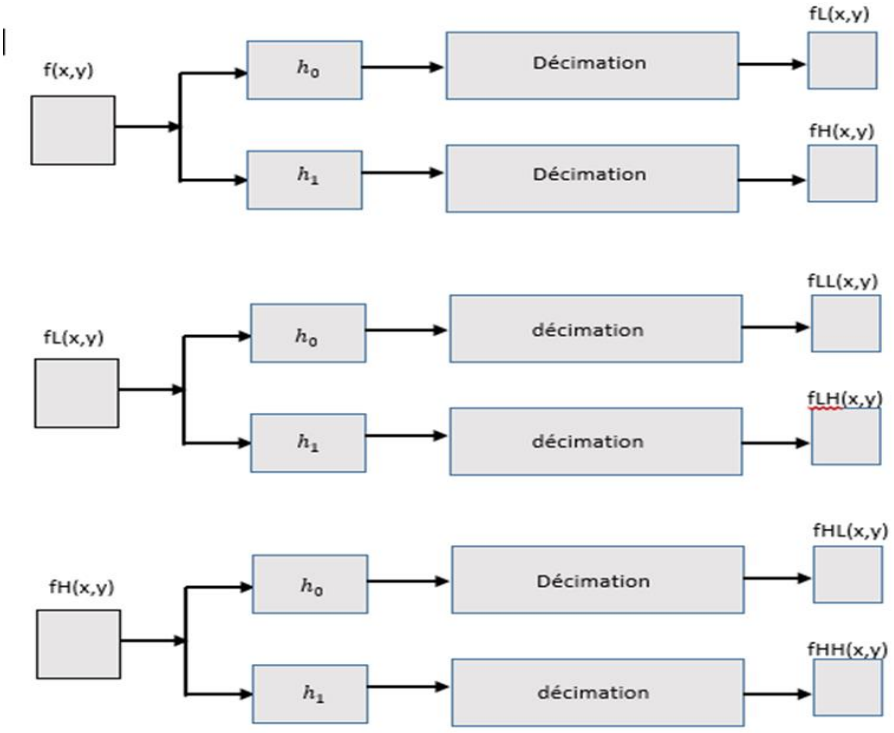

Figure 3. 6: Schéma de décomposition

3.4.2 Principe du système hybride proposé pour la compression et la sécurisation des images

L'idée primordiale est de réaliser un système combinant la compression (DWT-Huffman) et le cryptage ; il s'agit donc d'appliquer le cryptage sur les données de la compression.

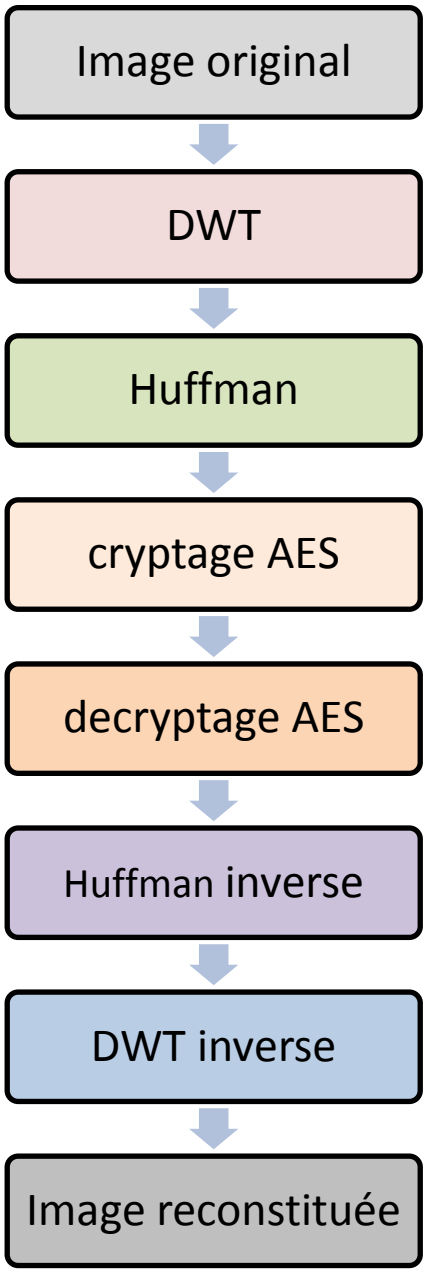

Figure 3. 7: Schéma de la compression hybride (DWT-Huffman) avec cryptage

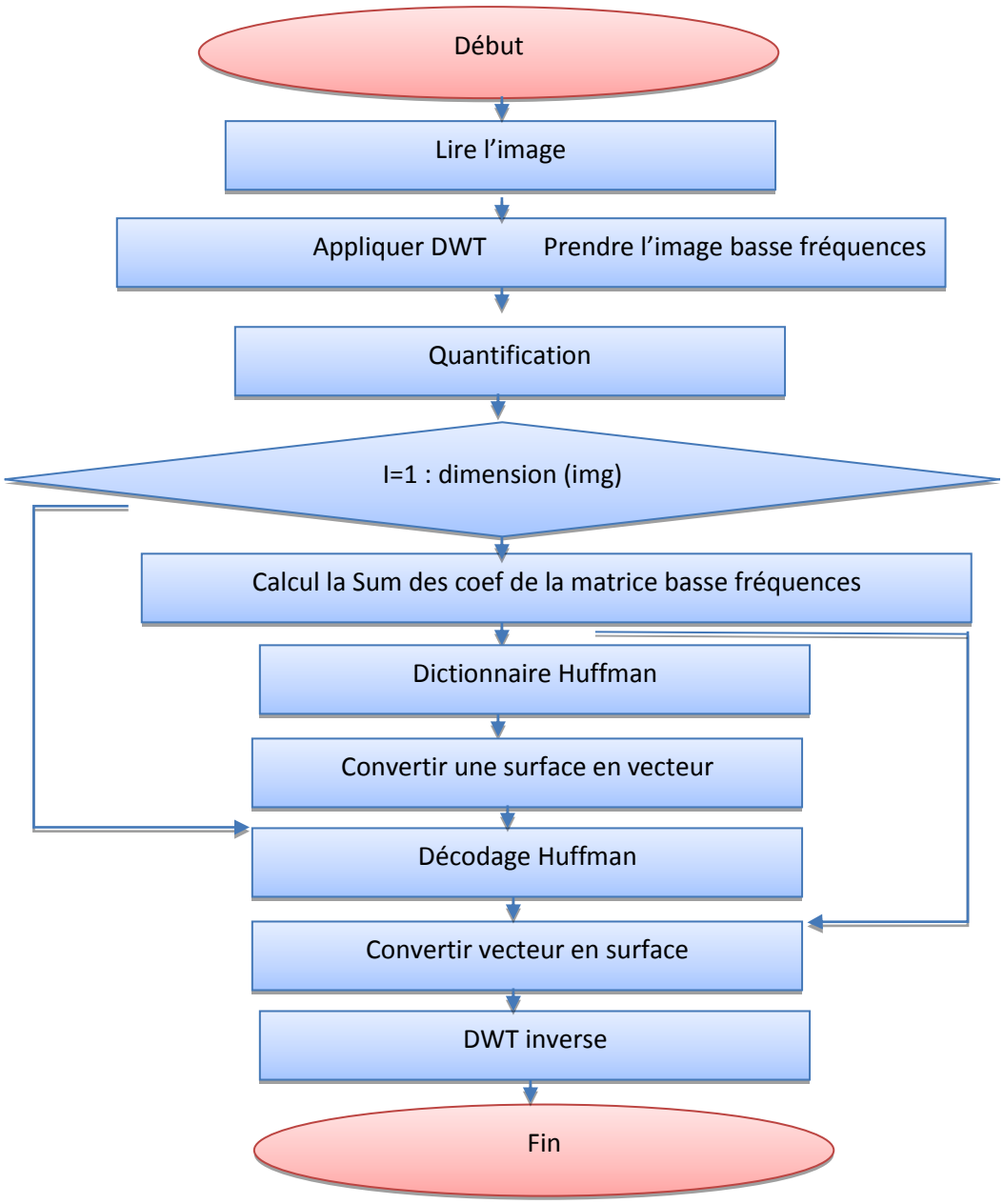

Figure 3. 8 : Organigramme du système hybride réalisé

3.5 Conclusion

Dans ce chapitre, nous avons présenté de différents types de compression (perte / sans perte) avec et sans chiffrement.

Dans une première partie, nous avons décrit quelques schémas de la compression sans cryptage et isolé les blocs de codage.

Ensuite, nous avons appliqué le cryptage AES sur les deux types de compression en introduisant brièvement le principe de fonctionnement du cryptage AES.

Nous avons terminé ce chapitre par présenter le schéma synoptique du système hybride de compression et cryptage d'images.

Chapitre 4 Mise en œuvre du système hybride DWT-Huffman avec chiffrement

4.1 Introduction

Pour la réalisation de l'interface d'analyse, deux solutions s'offraient :

- Utilisation d'un langage de programmation qui offre une richesse graphique conséquente.
- Utilisation d'un langage dédié au calcul scientifique et qui offre une interprétation évolué.

Cette ouverture permet l'ajout de fonction à ce même, qui sont assemblées sous forme de boite à outils, ce qui étend le champ d'action à des domaines aussi varie le contrôle, l'optimisation, le traitement d'image et bien évidemment le traitement du signal.

Notre choix s'est porté sur la deuxième catégorie qui inclue des logiciels tels que : Mathematica, Maple, Mathcad et Matlab. Du fait qu'il correspond exactement à nos besoins logiciels, nous nous sommes fixés finalement sur MATLAB.

4.2 Environnement de travail

4.2.1 matériels utilisés

L'implémentation de notre application « APP » a été réalisée sur un micro-portable fonctionnant sous le système d'exploitation Microsoft Windows 8.1 dont les performances sont les suivantes :

- Processeur Intel core(TM) i5-4200u CPU 1.6Ghz.
- Fréquence de 2.3 GHz.
- Mémoire RAM de 4 Go DDR2.
- Disque 500 Go SATA 5400 tours/mn.
- Carte graphique AMD Catalyst control redeon R5M240

4.2.2 Langage de programmation

- **MATLAB**

Le logiciel Matlab est un logiciel de manipulation de données numériques et de programmation dont le champ d'application est essentiellement les sciences appliquées. Son objectif, par rapport aux autres langages, est de simplifier au maximum la transcription en langage informatique d'un problème mathématique, en utilisant une écriture la plus proche possible du langage naturel scientifique. Le logiciel fonctionne sous Windows et sous Linux. Son interface de manipulation HMI utilise les ressources usuelles du multifenêtrage. Son apprentissage n'exige que la connaissance de quelques principes de base à partir desquels l'utilisation des fonctions évoluées est très intuitive grâce à l'aide intégrée aux fonctions.

Dans notre travail proposé on va créer une interface graphique .Quesque c'est une interface graphique ?

Les interfaces graphiques (ou interfaces homme-machine) sont appelées GUI (pour Graphical User Interface) sous MATLAB. Elles permettent à l'utilisateur d'interagir avec un programme informatique, grâce à différents objets graphiques (boutons, menus, cases à cocher…). Ces objets sont généralement actionnés à l'aide de la souris ou du clavier.

Malgré le fait que les interfaces graphiques semblent secondaires par rapport au développement du cœur d'une application, elles doivent néanmoins être conçues et développées avec soin et rigueur.

Leur efficacité et leur ergonomie sont essentielles dans l'acceptation et l'utilisation de ces outils par les utilisateurs finaux.

Une bonne conception et un développement maitrisé permettent également d'en assurer une meilleure maintenabilité. [31]

Depuis la version 5.0 (1997), MATLAB possède un outil dédié à la création des interfaces graphiques appelé GUIDE (pour 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.

4.3 Aperçu du logiciel réalisé

Le logiciel que nous avons implémenté est une mise en œuvre facile : pas de mot clés à connaitre ni de programme à écrire, l'utilisation est constamment guide en cliquant sur les boutons selon notre choix.

4.3.1 Hiérarchie

Notre interface présente une structure arborescente qui offre à l'utilisateur un bon suivi des applications effectuée et une meilleure représentation de ses données. Toutes les applications sont utilisées automatiquement à la fin de chaque session.la figure 4.1 illustre l'organigramme du logiciel élaboré.

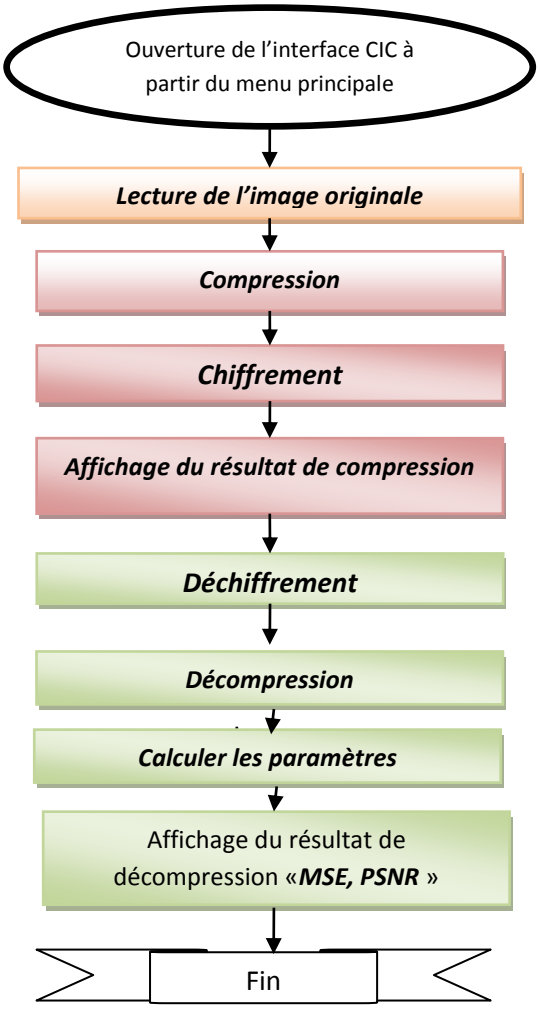

Figure 4. 1 : Organigramme de logiciel élaboré 1

L'application « APP », développée sous environnement MATLAB, consacre la première partie à la compression et le chiffrement des images suivant l'algorithme hybride basé sur la transformée DWT et le codage Huffman, et pour le chiffrement nous appliquerons l'algorithme d'AES. En deuxième partie nous faisant la décompression et le déchiffrement, pour bien valider la qualité de l'image reconstruite on calcul les paramètres de la distorsion à savoir le PSNR et MSE.

4.3.2 Description des modules

La figure ci-dessous présente l'interface de l'application qui s'intitule « Notre système Hybride pour la compression et la sécurisation des images »

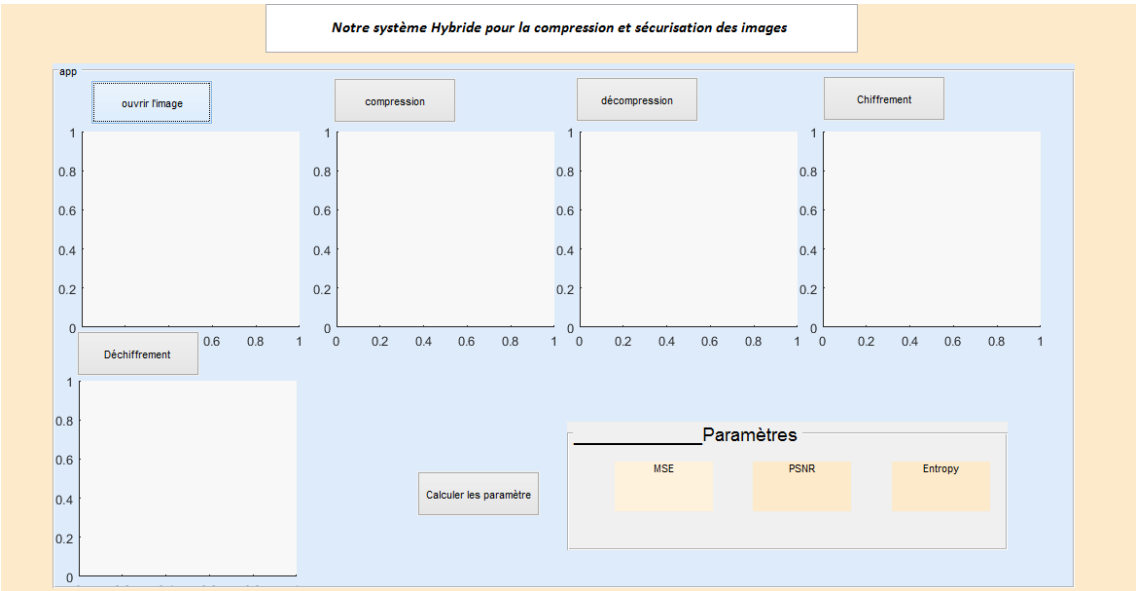

Figure 4. 2 : Application avant exécution

Principe de fonctionnement de l'application

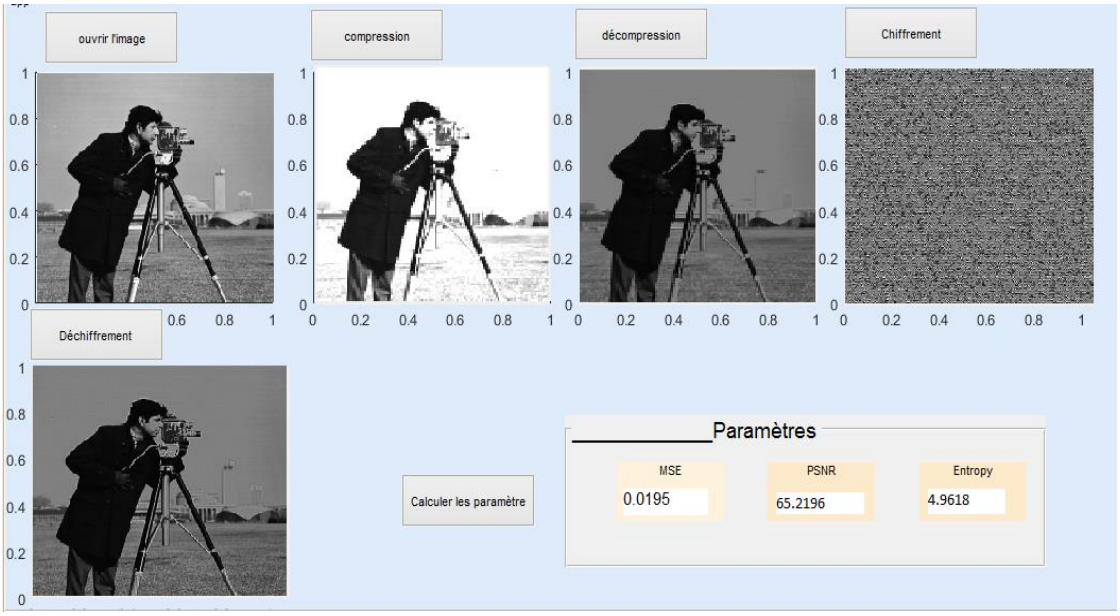

Figure 4. 3 : Application après exécution

- **Module 'lecture'** : Permet de charger une image à partir de n'importe qu'elle endroit du PC.

- **Module ‘compression’ :** Ce module est le plus important dans notre système ; il contient l’algorithme hybride « DWT, Huffman», il permet de faire la compression de l’image originale.

- **Module ‘chiffrement’ :** Ce module permet de crypter une image par le système de chiffrement AES.

- **Module ‘déchiffrement’ :** Ce module permet de décrypter l’image crypté.

- **Module ‘décompression’ :** Il permet de décompresser l’image et afficher l’image reconstruite.

- **Module ‘’calcule des paramètres’’** : Ce module est très important pour tester les performances de l’algorithme utilisé, en se basant sur trois paramètres, l'erreur quadratique moyenne ***(Mean Square Error, MSE)***, le rapport crête signal sur bruit ***(Peak Signal to Noise Ratio, PSNR)***, et l’entropie.

4.3.3 Bibliothèque d’images

Figure 4. 4 :Cameramn.tif 63.5ko

Figure 4. 5 : lina.bmp 245ko

Figure 4. 6 :Footbal.jepeg 11.2k

4.4 Tests expérimentaux

Nous présentons dans ce qui suit, les résultats issus de notre application, sur chacune des images abordées

4.4.1 Résultats du système

Notre travail est basé sur la compression et la sécurité d'image. En première partie nous allons tenter la compression de nos images avec différentes techniques afin de procéder à une comparaison entre elles en discutant sur les paramètres de performances de cette dernières, parmi ses paramètres : MSE, PSNR et le taux de compression.

Ensuite, nous appliquerons le chiffrement AES sur la DCT, DWT, Huffman et notre système hybride DWT-Huffman, avec deux façons distinctes (avant/après compression) pour enfin en conclure l'impact de chiffrement sur la compression.

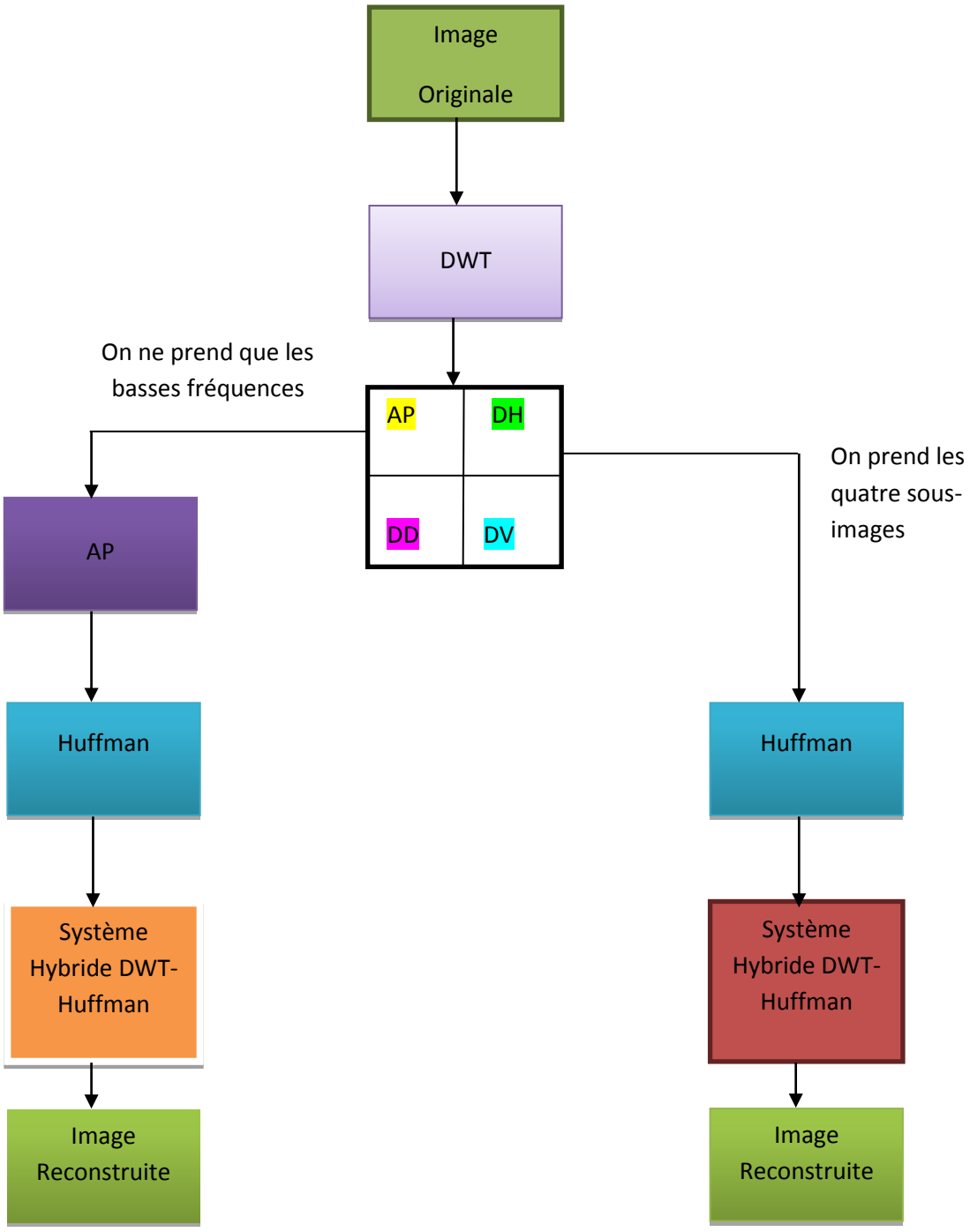

Figure 4. 7 : Schéma synoptique des cas de travail

4.4.2 Résultats de la compression

Les images de la collection étudiée, sont compressées suivant l'algorithme hybride DWT-Huffman.

La famille d'ondelette utilisée est Daubichies 1er niveau.

Dans la première partie de notre hybridation comme on voit dans la figure nous avons codé en Huffman l'image détail basse fréquence seulement.

Nom de l'image	**Notre approche Hybride**			**Notre approche Hybride (Pour l'image basse fréquence)**		
	Qc (%)	PSNR (dB)	MSE	Qc	PSNR	MSE
Lena.bmp **Dimension (500X500)** **Taille de l'image originale** **245 Kbytes**	96.74	65.2196	0.0195	98.76	24.5708	226.9876
cameraman.tif **Dimension (256X256)** **Taille de l'image originale** **67.2 Kbytes**	89.48	65.4289	0.0186	99.7	25.1451	198.8687

Figure application de notre système hybride sur les basses fréquences

- **Interprétation des résultats**

On remarque que PSNR est plus important dans l'approche hybride (appliqué que pour les basses fréquences) que l'approche hybride appliqué pour les quatre sous images détails avec un taux de compression moins important.

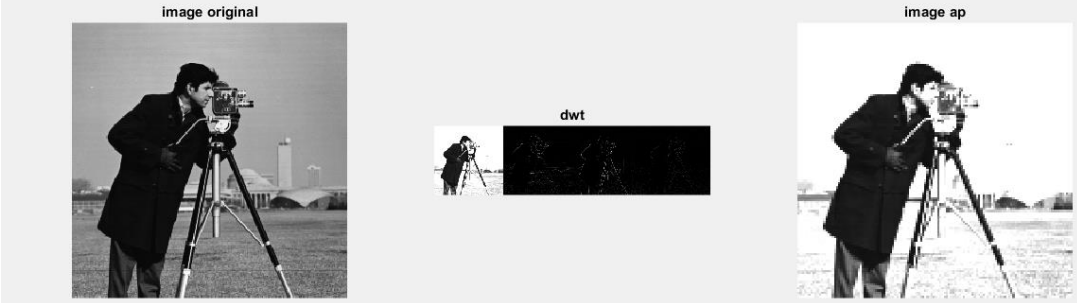

Figure 4. 8 : Transformer DWT

-L'image basse fréquence contient la plus grande partie d'information de l'image comme on voit dans la figure 4.9

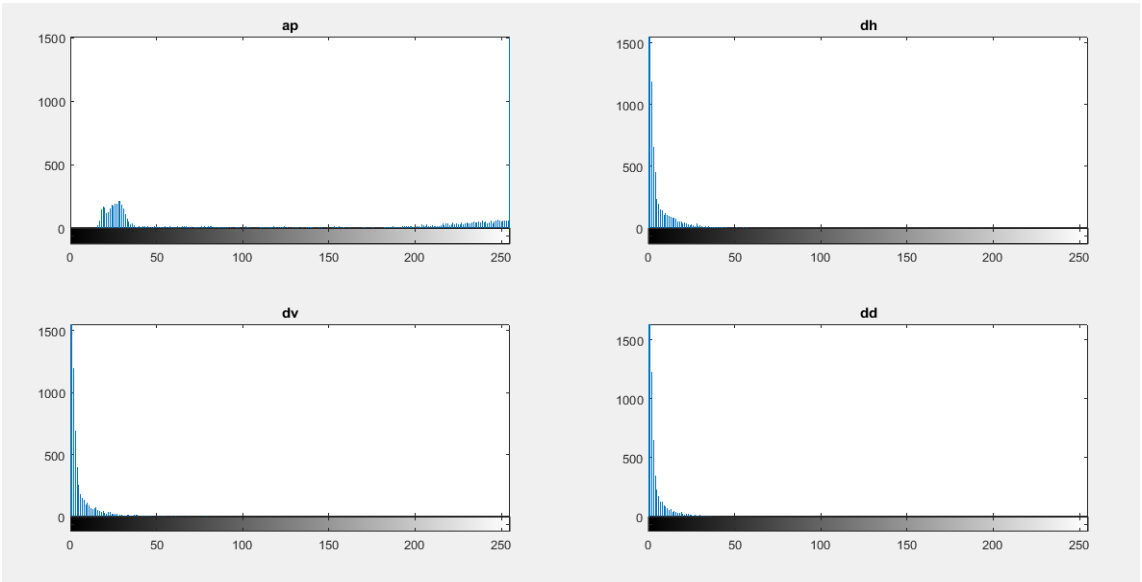

Figure 4. 9 : Histogramme DWT

En deuxième cas d'hybridation on va coder les quatre sous images détails en appliquant codage de Huffman et on obtient

Figure 4. 10 : Comparaison des images reconstruites

L'image reconstruite dans la figure est de meilleure qualité la compression des quatre images détails donne une meilleure qualité

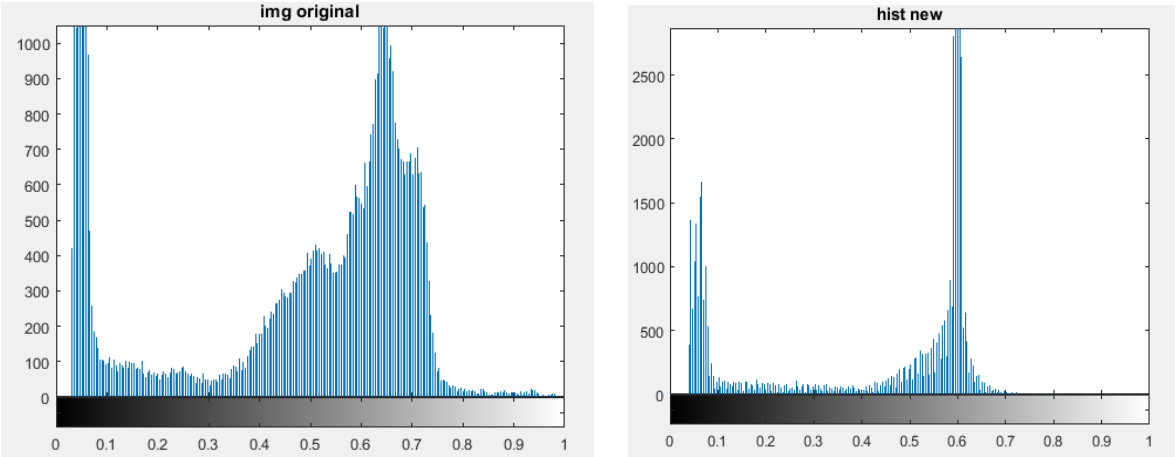

Figure 4. 11: Histogrammes des images reconstruites

- **Discussion**

Dans notre système on a pu fusionner les deux techniques en transformant les coefficients de matrice basse fréquence de DWT en vecteur pour calculer les fréquences d'apparition et construire l'arbre de Huffman, ceci dit les coefficients de la matrice DWT ne sont pas adaptés au codage Huffman.

On remarque que notre système hybride fonctionne, sauf que la perte de la qualité d'image est importante. Si on applique Huffman sur les quatre sous images détails l'erreur est plus élevé qu'en codant l'image basse fréquence.

4.4.3 Résultats de chiffrement

Nous allons ensuite appliquer le chiffrement avant et après pour notre système de compression, c'est pour cela qu'on procède à deux expériences afin d'extraire l'effet du chiffrement sur la compression en discutant sur MSE, le PSNR et Entropie.

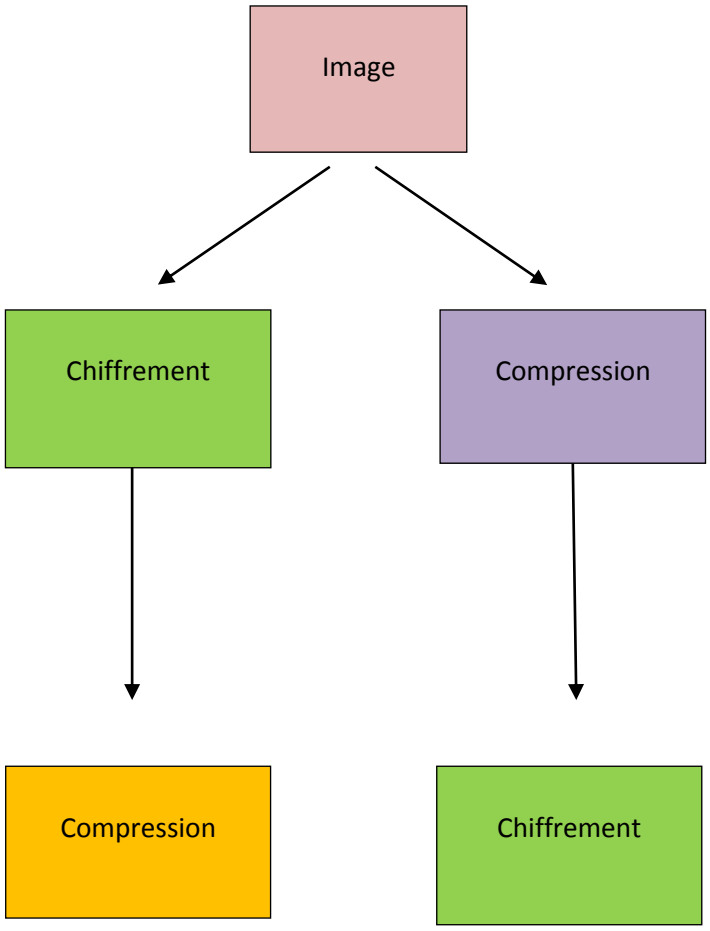

Figure 4. 12 : Synoptique chiffrement-de compression

Le tableau suivant englobe tous les résultats obtenus

	Compression hybride DWT-Huffaman					
	Sans cryptage			Avec cryptage		
	MSE	**PSNR**	**Entropie**	**MSE**	**PSNR**	**Entropie**
Lina .bmp	971.6733	27.558	18.2556	$1.0017^{e}03$	26.6024	7.0097
Cameraman. Tif	114.0776	27.558	7.4600	142.1808	18.1234	4.9618

Figure 1resultats de la compression hybride avant /après chiffrement

- **Interprétation des résultats**

L'effet d'AES sur la compression avec perte (DWT)

- Dans une première étude nous allons chiffrer l'image Lina basse fréquence issue de la transformé DWT puis la coder en Huffman et voir l'effet de chiffrement sur la compression

Figure 4.13 image basse fréquence chiffré

Image basse fréquence chiffré

Pour la transformer en ondelette on a pris seulement l'image basse fréquence, comme montre l'histogramme de chaque image détails de notre système hybride.

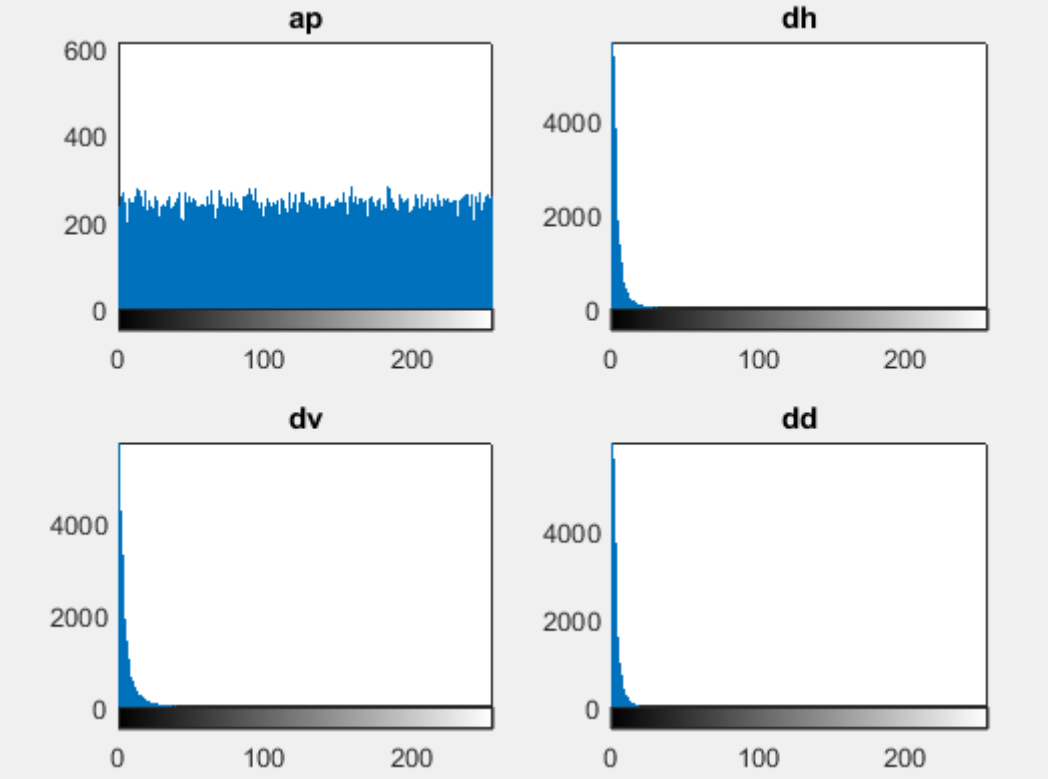

***Figure 4. 13*:** Histogramme d'image basse fréquence chiffré

A la sortie de notre système on a pu reconstruire l'image comme on voit dans la figure 4.14

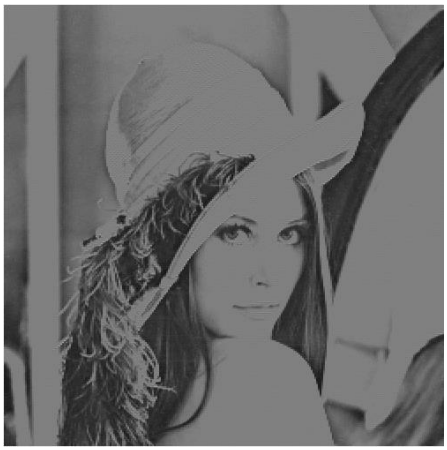

Figure 4. 14 : Résulta du premier cas

- dans un deuxième cas nous allons chiffrer l'image Lina par le système de cryptage AES et voir l'effet de chiffrement sur la compression.

L'image avant compression est complètement chiffrée

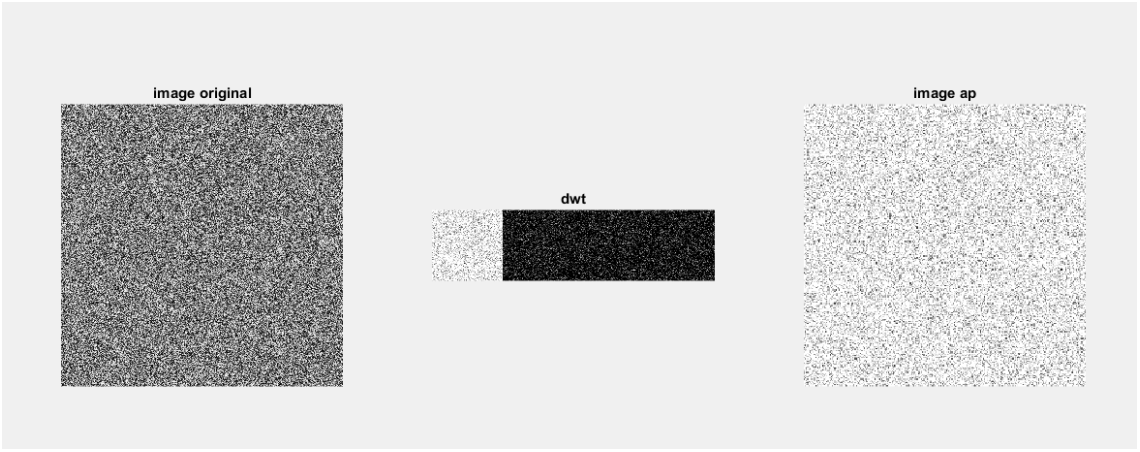

Figure 4. 15 :L'image originale chiffrée

L'image reconstruite est complètement bruitée

Figure 4. 16 : L'image reconstruite

- **Discussion**

➢ **Premier cas : chiffrement avant compression**

D'après les résultats obtenu le chiffrement AES a agi sur la compression, MSE (l'erreur quadratique moyenne augmente et PSNR diminue ce qui veux dire la fidélité de la compression il agit même sur l'entropie, et la reconstitution n'est pas fidèle.

➢ **Deuxième cas : chiffrement après compression**

D'après les résultats présentés on remarque bien que le chiffrement AES agit sur la robustesse des techniques de compressions qui donne des taux de compression de moins en moins faible et agis sur la reconstruction de l'image comme on à vue, si on applique le chiffrement avant compression on perd de la qualité de l'image et la compression et le MSE est très élevé.

Parmi les deux cas étudiés, si on applique le chiffrement sur la transformé DWT les informations de nos images seront déformés.

4.4.4 Comparaison de notre algorithme avec d'autres techniques de compression et chiffrement

Pour voir la robustesse de notre application de compression et chiffrement, on va comparer les résultats obtenus avec d'autres techniques de compression.

Nom de l'image	Global thresholding of coefficients and Huffman encoding *'gbl_mmc_h'*		Embedded Zerotree Wavelet *'ezw'*		Notre approche Hybride		Notre approche Hybride pour l'image basse fréquence	
	CR (%)	PSNR (dB)	CR (%)	PSNR (dB)	CR (%)	PSNR (dB)	CR	PSNR (dB)
Lena Dimension (500X500) Taille de l'image originale 254 Kbytes	82.91	28.24	95.97	16.53	38.29	65.21	23.57	24.57
Cameraman Dimension (256X256) Taille de l'image originale 67.6 Kbytes	78.02	31.13	97.72	19.86	29.35	65.42	18.89	25.14

Figure.2compraison entre notre système hybride proposé et d'autres systèmes

- **Discussion**

D'après les résultats présentés dans le tableau 3.4 on remarque bien que le taux de compression est très faible.

Notre technique de compression, nous a donné de bonnes reconstructions en utilisant la premier méthode (coder l'image basse fréquence en Huffman) mais notre système n'est pas assez faible par rapport aux deux méthodes présentés dans le tableau qui se base uniquement sur la transformées en ondelette, et lorsqu'on applique le chiffrement, le taux de compression deviens plus faible alors le chiffrement affecte la qualité de la compression.

Donc pour améliorer notre système il faut trouver un algorithme de crypto-compression et une adaptation du codage Huffman après transformé en ondelettes. En termes de paramètres distorsion ainsi le taux de compression.

4.5 Conclusion

Dans la première partie de ce chapitre, nous avons fusionné la transformé en ondelette et le codage de Huffman afin d'avoir un système hybride qui sert à compresser l'image. Ensuite, nous avons testé plusieurs types d'image à l'entrée de notre méthode et cela donne des images reconstruites après décompression de bonne qualité.

Dans la deuxième partie de ce chapitre, nous avons appliqué le chiffrement sur la compression. L'algorithme de chiffrement par un bloc appliqué aux images présentent deux inconvénients d'une part quand l'image contient des zones homogènes, tous les blocs identiques son identique après chiffrement pour cela l'image crypté contient des zones texturés et entropie de l'image n'est pas le même en plus les techniques de cryptage par blocs ne sont pas robustes au bruit.

D'après les résultats présentés, on remarque bien que le chiffrement AES agit sur la robustesse des techniques compressions, qui donne des taux de compression de moins en moins faible et agit sur la reconstruction de l'image comme nous avons vu, si on applique le chiffrement avant compression l'image obtenue est complètement bruité.

Si on crypte seulement les coefficients de la Matrice après DWT, l'image obtenu est parfaitement illisible, toutefois il est impératif de signaler qu'il est possible d'extraire l'image.

Conclusion générale

La compression des données est appelée à prendre un rôle encore plus important, en raison du développement des réseaux de télécommunications. Son importance est surtout due au décalage qui existe entre les possibilités matérielles des dispositifs que nous utilisons et les besoins qu'expriment les applications. De plus, cet échange grandissant des données fait appel à la cryptographie pour sécuriser les informations transférées. Dans ce mémoire, nous avons élaboré une technique de compression et sécurité d'images pour faciliter l'archivage et assurer la confidentialité d'images.

Pour ce faire, nous avons commencé par un état de l'art des méthodes et techniques de compression et de cryptage existantes. A partir de cette étude on a proposé un système hybride qui sert à compresser l'image en appliquant la transformée en ondelette et le codage de Huffman ; et pour le chiffrement on a utilisé l'AES.

Les résultats des simulations exhaustives de notre système hybride qui combine la DWT et Huffman, montrent une certaine discorde des performances pour le système hybride par rapport aux méthodes basées uniquement sur la DWT et Huffman.

Coté compression et cryptage, nous avons proposé deux méthodes de combinaison entre la compression et le chiffrement ; en appliquant l'AES avant /après compression pour conclure l'effet de chiffrement sur la compression.

Donc pour améliorer notre système il faut trouver un algorithme de crypto-compression et une adaptation du codage Huffman, après transformer en ondelettes. En termes de paramètres de distorsion ainsi que le taux de compression.

Bibliographie

[1] Z-E. BAARIR, A. OUAFI : « ETUDE DE LA TRANSFORMEE EN ONDELETTES DANS LA COMPRESSION D'IMAGES FIXES », Laboratoire de recherche LESIA, Département d'Electronique, Université Mohamed Khider, Biskra, Algérie, Juin 2004.

[2] Prof. Slimane Larabi : « Cours : SYSTEMES MULTIMIDIA », Master RSD, universitédes Sciences et de la Technologie Houari Boumediene, Alger, Algérie, 2014/2015.

[3] http://projet.eu.org/pedago/sin/C2i-Image.pdf

[4] MouradLAHDIR,SoltaneAMEUR, et Abd El Hamid ADANE : « ALGORITHME NON ITÉRATIF BASÉ SUR LES ONDELETTES BIORTHOGONALES ET LES FRACTALES POUR LA COMPRESSION D'IMAGES SATELLITAIRES », CONTEMPORARY PUBLISHING INTERNATIONAL Publié sous l'enseigne Éditions Scientifiques GB, 2006.

[5] D. Zeroual : '' IMPLEMENTATION D'UN ENVIRONNEMENT PARALLELE POUR LA COMPRESSION D'IMAGES AL'AIDEDES FRACTALES '', magistère en informatique, option : informatique industrielle, Batna, 2006.

[6] Brault &Dougherty : '' Les formats de compression d'image '', Institut Universitaire de Technologie de Tours Département Génie Électrique et Informatique Industrielle, Promotion 2002-2004.

[7] Atelier « Maths en Jeans », codage de Huffman, Année scolaire 2007-2008 Lycée Louis Lapicque – Épinal.

[8] Pierre Geurts : « cour : Structures de données et algorithmes », R 141 (Montefiore), Version du 6 février 2014.

[9] F. Davoine, "Compression d'images par fractales basée sur la triangulation de Delaunay," Institut National Polytechnique de Grenoble - INPG, France 1995.

[10]M.Lahdir : ”nouvelle approche de compression d’images basé sur les ondelettes et les fractales : application aux images météosat ”, Thèse de doctorat en électronique option : télédétection, université mouloud Mammeri, Tizi-Ouzou ,2010.
[11] http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html
[12]P. PLUME, " Compression de données ", Editions Eyrolles, 1993.
[13]Mme LAHDIR : « Compression d’images par SPIHT appliqué dans le domaine des ondelettes entières : Application aux images MSG », Mémoire de Magistère en Electronique, option Télédétection.
[14] M. BOUCHEMA :’’Exploitation des transformées paramétriques dans le cryptage des images fixes’ ’Mémoire de Magistère, Option : Communication, UNIVERSITE FERHAT ABBAS –SETIF 1- UFAS (ALGERIE), Faculté de Technologie Département d'Electronique, 2012.

[15]http://dspace.univ-tlemcen.dz/bitstream/112/1076/5/chapitre1.pdf

[16]Allal Mohamed Lamine : «Cryptologie appliquée», Mémoire Master, université des Sciences et de la Technologie Houari Boumediene, Alger, Algérie ,25 Juin 2016.

[17] http://igm.univ-mlv.fr/~dr/XPOSE2007/vma_PKI/concepts_de_base.html

[18] https://www.researchgate.net/figure/Principe-de-fonctionnement-dun-algorithme-de-cryptage-symetrique-Cryptographie_fig24_277474499

[19] http://dspace.univ-tlemcen.dz/bitstream/112/1046/8/chapitre2.pdf

[20]’’le cryptage’’ : RASIDY – DEBUYS, Quentin – Lionel, 6°U – Sciences informatiques, 2011-2012

[21]‘la cryptologie moderne’ AnneCanteaut, Françoise Levy-dit-Vehel, Ecole supérieur des techniques avancées.

[22] H. Benzenine et K. Amara : ‘’La cryptographie appliquée sur les fichiers audio (son)’’, Mémoire de Master en Informatique,Université Abou Bakr Belkaid– Tlemcen Faculté des Sciences Département d’Informatique, Option : Système d’Information et de Connaissances (S.I.C) ,2011.

[23]W. Puech, C. Gouenou : « Codage hybride cryptage-marquage-compression pour la sécurisation de l’information médicale »,08/02/2007

[24]http://www.cryptage.org/cle-publique html

[25] J. BLANC & A. DE GEORGES : « TECHNIQUES DE CRYPTOGRAPHIE », Licence Informatique, Année universitaire, 2003/2004

[26]http://dictionnaire.sensagent.leparisien.fr/Cryptographie%20asym%C3%A9trique/fr-fr/#Faiblesses

[27]M. Abo-Zahhad, R.RagabGharieb, S. M. Ahmed et M. Khaled Abd-Ellah : « Huffman Image Compression IncorporatingDPCM and DWT »,Journal of Signal and Information Processing, 2015, 6, 123-135,Published Online May 2015 in SciRes

[30]Mohammed ALDOSSARI : « Nouvelle méthode optique decompression et de cryptagesimultanés des images(fixes/vidéo) pour lessystèmes télécommunication »,Thèse soutenue le 15 Décembre 2014

[31]https://briot-jerome.developpez.com/matlab/tutoriels/introduction-programmation-interfaces-graphiques/