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

Mémoire de fin d'étude
Pour l'obtention du diplôme de MASTER
Domaine : Mathématiques et Informatique
Filière : Informatique

Un système de crypto-compression d'images basé sur le Block Cipher et la compression fractale

Option : Réseaux et Sécurité Informatique
Thème
Présenté Par :

BADAOUI Youcef Islem

Devant le jury :

Mr.BENNOUR Akram	*MCA*	*Univérsité Larbi Tébessi*	*Président*
Mr.GAHMOUS Abdellatif	*MAA*	*Univérsité Larbi Tébessi*	*Examinateur*
Mr.MENASSEL Rafik	*MCA*	*Univérsité Larbi Tébessi*	*Encadreur*

Date de soutenance : 13 Juin 2020

Dédicaces

A mes chers parents,

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes très chère, respectueux et magnifiques parents qui m'ont soutenus tout au long de ma vie, dont leurs mérites, leurs sacrifices, leurs qualités humaines m'ont permis de vivre ce jour : les mots me manquent pour exprimer toute la reconnaissance, la fierté et le profond amour que je vous porte pour les sacrifices qu'ils ont consenti pour ma réussite, qu'ils trouvent ici le témoignage de mon attachement ma reconnaissance, gratitude et respect, que Dieu leur préservent bonne santé et longue vie. Tous mes sentiments de reconnaissance pour vous.

A toute ma Famille « mes chères sœurs et mes frères, »

En témoignage de mes sincères reconnaissances pour les efforts qu'ils ont consenti pour l'accomplissement de mes études. Je leur dédie ce modeste travail en témoignage de mon grand amour et ma gratitude infinie.

A tous mes amis sans exception,

Pour leur aide et leur soutien moral durant l'élaboration du travail de fin d'études.

B.Youcef Islem

REMERCIMENT

C'est avec un grand plaisir que je réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail.

Je tiens tout d'abord à remercier Dr. BENNOUR Akram, maitre de conférences à l'université Larbi Tébessi, président du jury et Mr. GAHMOUS Abdellatif, maitre-assistant à l'université Larbi Tébessi examinateur, pour l'honneur qu'ils m'ont accordé en acceptant de juger mon travail

Je remercie sincèrement Dr. Menassel Rafik, maitre de conférences à l'université Larbi Tébessi, pour son encadrement, son assistance, son soutien, sa disponibilité et ses précieux conseils durant la période de ce stage.

Je m'acquitte, enfin, volontiers d'un devoir de gratitude et de remerciements à tous mes enseignants pour la qualité de l'enseignement qu'ils ont bien voulu me prodiguer durant mes études afin de me fournir une formation efficiente.

B.Youcef Islem

Table des matières :

Table des Figures :

Table des Tableaux :

INTRODUCTION GENERALE

Introduction générale :

- L'utilisation des technologies de l'information 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.

- Ensuite, avec la grande accélération dans le développement des technologies d'Internet et de la communication, la communication des images. Cependant, la sécurité de l'information est un sujet sensible pour la recherche, la discussion et le développement, et le cryptage est l'une des meilleures alternatives qui s'est avérée efficace tout au long de l'histoire pour assurer la confidentialité et la sécurité de l'information.

- Les algorithmes de chiffrement par blocs appliqués aux images représentent deux inconvénients. Premièrement, quand l'image contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement. Dans ce cas, l'image cryptée contient des zones texturées et l'entropie de l'image n'est pas maximale. Le second problème est que les méthodes de cryptage par blocs ne sont pas robustes au bruit. En effet, une erreur sur un bit chiffrer va propager des erreurs importantes dans tout le bloc courant.

- Dans ce travaille nous proposons un système de crypto-compression qui utilise les deux technique la compression fractale, et le chiffrement par bloc AES. Pour mener à bien notre travail, nous avons structuré notre mémoire en trois chapitres :
- Le premier chapitre c'est l'état de l'art se compose en deux partie ; la compression et la cryptographie :

- La première partie 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 et enfin en mettant le point sur la compression fractale.
- La deuxième partie nous amène dans le monde de la cryptographie ; en commençant par une définition détaillée suivi par une présentation des méthodes de cryptage parmi les plus utilisées, ensuit nous expliquons les deux types de la cryptographie (classique et moderne), et enfin nous mettons le point sur la cryptographie par bloc.

➢ Le deuxième chapitre qui introduit la notion de crypto-compression nous présentons une explication plus détaillés des deux techniques utilisées dans notre système, en commençant par la compression fractale par bloc et nous décrivons par la suite l'algorithme générique de compression et décompression fractale. Ensuite nous l'approche de cryptage à base de Block Cipher en utilisant l'algorithme AES.

➢ Le Troisième chapitre comprend la partie la plus importante de ce travail sera consacré l'approche de crypto-compression élaborée dans le cadre de ce travail ; en commençant par une présentation de l'environnement de travail et une aperçu générale de notre système, nous citons par la suite les tests expérimentaux sur des images réelles et les résultats obtenues.

PREMIER CHAPITRE
ETAT DE L'ART

Introduction :

➢ Les applications qui utilisent des données multimédias, comme les images, sont de plus en plus présentent dans notre vie quotidienne. Ainsi manipuler les images (stocker ou transmettre,…) devient un enjeu stratégique. Du 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 sont simples, rapides et efficaces [1].Ce chapitre expose le contexte de notre travaille. En effet dans cette partie nous avons présentés les deux technologies utilisées dans notre système la compression et le cryptage des données. La première partie est consacrée sur la compression d'images leur principe, les types de compressions, les caractéristiques des méthodes de compression et enfin, on met le point sur la compression fractale. La deuxième partie concerné le cryptage des données leur principe, leur objectifs, la cryptographie classique et moderne (symétrique et asymétrique) et enfin, on met le point sur le block Cipher.

I. La compression :

1. Principe :

➢ La compression des données, d'une maniéré générale, c'est l'ensemble des méthodes ou techniques que l'on utilise pour prendre un message long pour en faire un message courte, c'est-à-dire réduire le volume d'une donnée sans perdre les informations essentielles. Le but de la compression est de représenter les informations avec une forme plus compacte que l'original ou le résultat de la compression occupe moins d'espace que la donnée originale. Les données peuvent être compressées avec perte ou sans perte. [2]. La Figure 1 présente le principe de la compression des images :

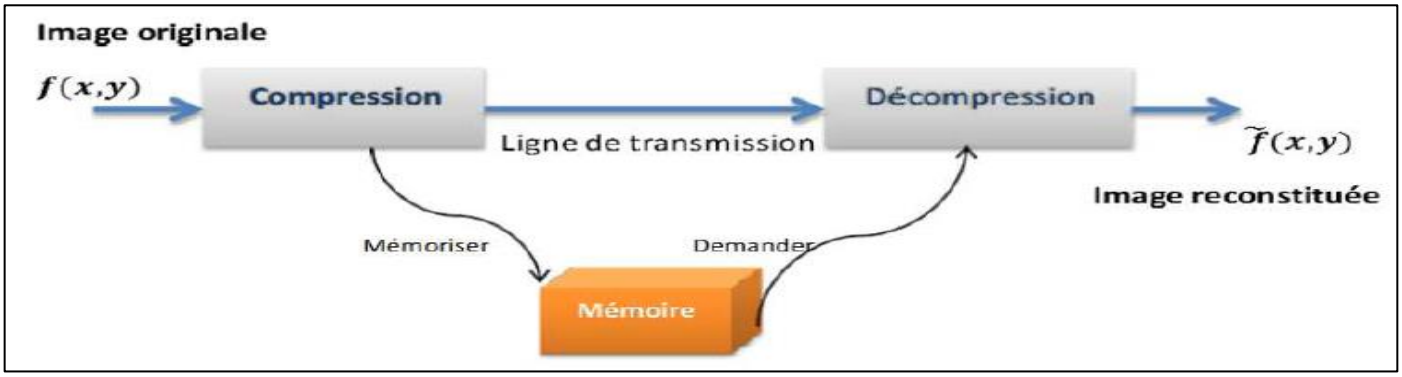


Figure 1 : Principe de Compression d'images [3]

- Les méthodes de compression d'images ont des critères d'évaluations, dont on peut citer : [4]
  - La qualité de reconstitution de l'image.
  - Le taux de compression.
  - La rapidité du codeur.
- Le schéma fonctionnel de la compression est présenté dans la Figure 2 ci-dessous :

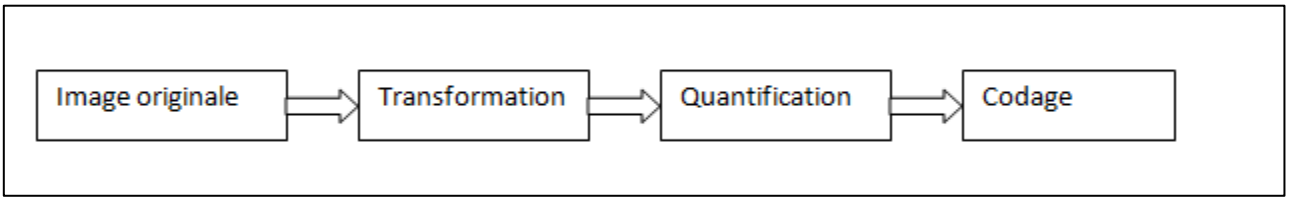


Figure 2 : Schéma d'un codeur d'image [4]

- A partir de ce schéma, nous allons revoir chacune de ses étapes à fin de préciser leur rôle.

  - **Transformation:**
    - La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduisent une corrélation très forte sur l'image.

  - **Quantification :**
    - La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression.

  - **Codage :**
    - Une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire a priori les deux conditions suivantes:
      - Unicité : deux messages différents ne doivent pas être codés de la même façon.
      - Déchiffrable : deux mots de codes successifs doivent être distingués sans ambiguïté. [4]

2. Compression physique et logique :

- **La compression physique :** s'applique uniquement aux données de l'image. Il s'agit de translater les trains de bit d'un motif à un autre.
- **La compression logique :** est effectuée par un raisonnement logique en substituant une information par une information équivalente. [5]

3. Compression symétrique et asymétrique :

- Dans le cas de la compression symétrique, la même méthode est utilisée pour compresser et décompresser l'information, il faut donc la même quantité de travail pour chacune de ces opérations.
- La compression asymétrique demande plus de travail pour l'une des deux opérations.

4. Caractéristiques des méthodes de compression :

4.1. Rapport et taux de compression :

- Le taux de compression est défini comme le rapport du nombre de bits utilisés par l'image originale et du nombre de bits utilisés par l'image compressée. [6]

$$\boldsymbol{CR}^{1} = \boldsymbol{rapportc} = \frac{\boldsymbol{nombre\ de\ bits\ avants\ compression}}{\boldsymbol{nombre\ bits\ aprés\ compression}}$$

- Le taux de compression est un pourcentage de l'espace obtenu après la compression par rapport à l'espace total requis par les données avant la compression. Qu'un fichier compressé indique à sa taille originale aura un taux de compression de 0 %. Un fichier réduit à 0 octet, aura un taux de compression de 100%. [7]

$$\boldsymbol{Tc} = \boldsymbol{TauxC} = \left(\mathbf{1} - \frac{\mathbf{1}}{\boldsymbol{CR}}\right) * \mathbf{100}$$

4.2. Mesure de distorsion

- Pour mesurer la distorsion entre l'image reconstruite et l'image originale (Mesure de la qualité visuelle de l'image reconstruite) on va utiliser l'Erreur Quadratique Moyenne [2]MSE (Mean Square Error) ou du rapport signal à bruit PSNR[3] (Peak Signal to Noise Ratio).
- Etant donnée une image originale composée de pixels αi(i=1...N) et l'image décodée composée de pixels άi(i=1...N) . Alors l'erreur quadratique moyenne est donnée par : [7]

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\mathbf{1}}^{\boldsymbol{N}} (\boldsymbol{\alpha i} - \acute{\alpha}\boldsymbol{i})^{\mathbf{2}} \text{ [7]}$$

- L'autre critère objectif cité plus haut est le rapport signal sur bruit de l'image reconstruite PSNR (en anglais Peak Signal to Noise Ratio). Il est défini par l'équation : [1]

$$\boldsymbol{PSNR} = \mathbf{10} \log \mathbf{10} \frac{(\mathbf{2}^{\boldsymbol{R}} - \mathbf{1})}{\boldsymbol{MSE}} \textbf{ DB}^{4} \textbf{ (décibels)} \text{ [7]}$$

$$\textbf{Ou: } \boldsymbol{PSNR} = \mathbf{10} \log \mathbf{10} \frac{\boldsymbol{d}^{\mathbf{2}}}{\boldsymbol{MSE}} \text{ [1]}$$

d représente la valeur d'intensité maximale de l'image. En utilisant des images en niveau de gris, les valeurs sont ainsi codées sur 8 bits et dans ce cas de figure $d \leq 2^8 - 1$. [1]

---

[1] **CR : En anglais « *Compression Ratio* » : c'est le rapport de compression**.
[2] **MSE** : En anglais « *Mean Square Error* » c'est l'Erreur Quadratique Moyenne**.**
[3] **PSNR**: En anglais « *Peak Signal to Noise Ratio* » c'est le rapport signal à bruit.
[4] **DB** : Le décibel (dB) est une unité définie comme dix fois le logarithme décimal du rapport entre deux puissances1, utilisée dans les télécommunications, l'électronique et l'acoustique.

4.3. Entropie:

➢ Est une métrique de « surprise » dans la mesure où, si l'entropie est élevé, les prédictions sont difficiles à faire et si l'entropie est faible, la séquence facilement prévisible si une séquence contient beaucoup de « surprise », elle contient beaucoup d'information. Si elle ne contient pas beaucoup de « surprise », elle ne contient pas beaucoup d'information Est définie par la formule suivante : [2]

$$H = \sum_{K=0}^{2^R-1} P(K) \log_2 P(K) \ \ bpp$$

➢ Avec : P(K) est la probabilité d'apparition des niveaux de gris dans l'image, K est la valeur de gris et R est le nombre de bits par pixels. [6]

4.4. 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. [8]

5. Les objectifs de la compression :

➢ De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante des réseaux en fonctionne de cette dernière nous citons les objectifs de la compression :

- La rapidité de la compression et la décompression.
- La compression d'images permet de réduire énormément la taille des images.
- Le temps de transmission gagné par une réduction de la taille des données.
- Garantie de non perdu de l'image entière grâce à la robustesse de l'algorithme de compression.
- Deux qualités le taux de compression et la qualité de l'image après un cycle de compression\décompression.

6. Type de compression :

➢ Les types de compression sont faits grâce aux redondances des données présentes sur l'image, ces redondances sont :

- **Redondance psycho visuel** : des détails non perceptible à l'œil humain qu'on peut d'éliminé (caractéristiques de l'œil humain).
- **Redondance inter** [5]**pixel** : la possible corrélation existante entre les pixels de l'image, on dit qu'une image a une redondance inter pixel si c'est possible de prédire la valeur d'un pixel en connaissance de la valeur des pixels voisin (suivant ou précédent), sachant que plus la résolution de l'image est grande plus la possibilité de rencontrer des redondances inter pixel est élevée.
- **Redondance de codage** : séquence de répétition des bis, en rencontre généralement à la fin de la compression, pendant l'étape de codage. [9]

➢ On peut distinguer deux grandes types de la compression (Figure 3) : 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. [7]

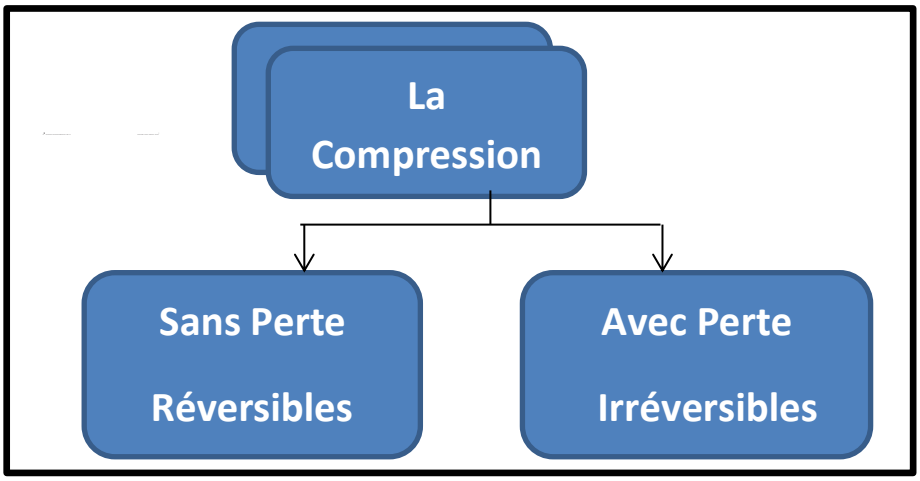


Figure 3 : Types de Compression

6.1. La compression sans perte :

➢ Comme son nom l'indique, ce type de compression n'occasionne aucune perte de données, appelle aussi compression non destructrice. La qualité de l'image après décompression est la même que celle de l'image original, le taux de compression de ce type est limite. [10]

➢ D'autre manière on donne un exemple plus simple comme il montre dans la (Figure 4), un ensemble de bits X a comme résultat de compression le compressé Y plus court que X, ce

[5] **Pixel :** Le pixel est souvent abrégé p ou px. Il est l'unité de base permettant de mesurer la définition d'une image numérique matricielle. Son nom provient de la locution anglaise picture element, qui signifie « élément d'image ».

résultat est stocké ou transmis. Lors de la décompression du résultat Y, on récupèrera par exactitude l'ensemble de bits de départ X. [2]

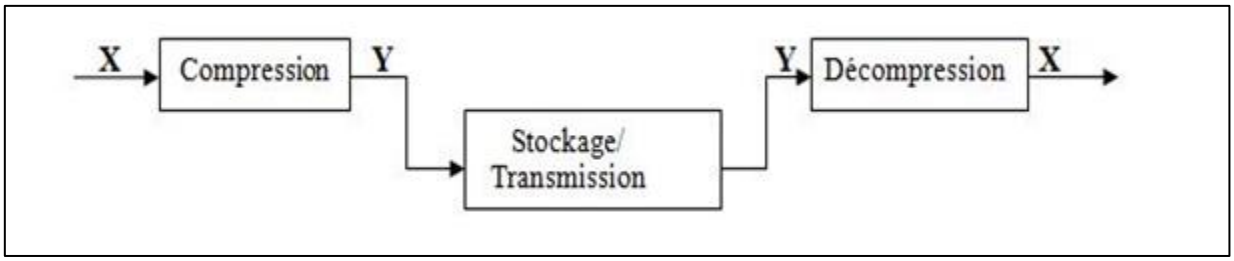


Figure 4 : Compression sans perte [2]

- Ce type de compression est nécessaire pour certaines applications ou la précision est majeure telles que les images médicales (**IRM**[6]) ou la télédétection (imagerie satellite).
- Les algorithmes de compression employée sont nombreux, les plus importants sont : [7]
    - Codage de Huffman.
    - Codage de Shannon-Fano.
    - Codage arithmétique.
    - Le codage par répétition ou "Run Length Coding" **(RLC)**[7]
    - Codage par dictionnaire adaptatif **(LZW)**[8] (Lempel-Ziv-Welch) ou LZ77

6.2. Compression avec perte :

- C'est une compression destructrice, les techniques de compression avec perte impliquent une certaine perte d'information ou d'avoir une permission de supprimer quelques données (qui sont inutiles) de l'information pour avoir une meilleure compression, ce qui signifie les données qui ont été compressées à l'aide de techniques à perte ne peuvent généralement pas être récupérées ou reconstruites exactement. [3]
- D'une façon simplifié on donne un exemple comme il montre dans la (Figure 5), On a une information X qu'on veut compresser, son résultat de compression est Y, après décompression du compressé Y on a un résultat Z qui est une approximation de X (Z=X). [2]

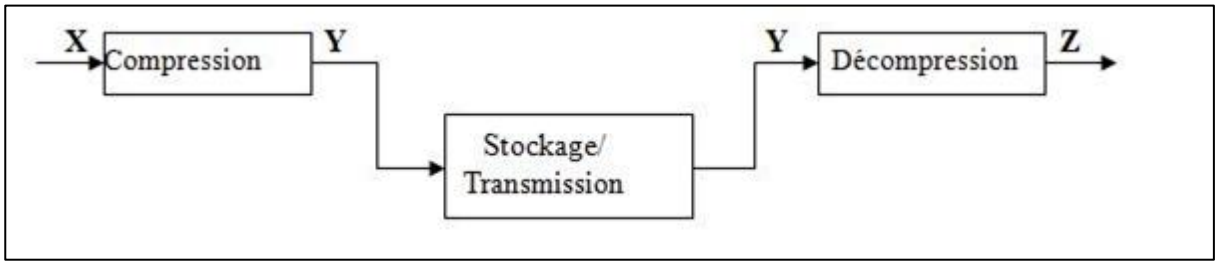


Figure 5 : Principe de compression avec Perte [2]

[6] **IRM :** L'imagerie par résonance magnétique (IRM) est une technique d'imagerie médicale permettant d'obtenir des vues en deux ou en trois dimensions de l'intérieur du corps de façon non invasive avec une résolution en contraste relativement élevée.
[7] **RLC :** Le run-length encoding, appelé en français le codage par plages, est un algorithme de compression de données sans perte en informatique.
[8] **LZW** : LZW (pour Lempel-Ziv-Welch) est un algorithme de compression de données sans perte. Il s'agit d'une amélioration de l'algorithme LZ78 inventé par Abraham Lempel et Jacob Ziv en 1978. LZW fut créé en 1984 par Terry Welch, d'où son nom.

➢ Ce type de compression on le trouve généralement dans le domaine-là où la réduction du poids de limages est très importante, comme le domaine multimédia par exemple (Web, photographie) où la fidélité envers l'image originale n'est pas très importante et le taux de compression sera plus grand que celui d'une compression sans perte du fait qu'on est juste limites par la qualité qu'on souhaite obtenir. [3]

➢ Les algorithmes de compression employée sont nombreux, les plus importants sont :

- Quantification.
- Codage par transformée.
- Le codage en sous-bandes.
- La compression par ondelettes.
- **La compression fractale.**

7. La compression fractale :

7.1. Pourquoi la compression par fractales ?

- La compression Fractale est une méthode récente, qui s'applique uniquement aux images. Le format des images compressé par ce procède n'est pas encore standardisé à ce moment. Elle a été proposée pour la première fois par Michael Barnsly 1988. [7]
- Les méthodes standard de compression d'images peuvent être évaluées par leur taux de compression: le quotient de la mémoire occupée par l'image comme collection de pixels par la mémoire nécessaire pour stocker l'image sous forme compressée.
- Le taux de compression d'images par fractales est élevé, ce qui justifie notre choix de cette méthode de compression. [11]

➢ Principe :

- La compression d'images fixes par une méthode fractale utilise les propriétés bien connues des fractales : La récurrence des motifs. Ce genre de compression tend à éliminer la redondance d'informations dans l'image, en recherchant tous les motifs, toutes les zones de l'image qui se répètent dans l'image. La récurrence des motifs s'effectue parfois de manière directe (seule l'échelle est différente), et parfois de manière indirecte (transformation, rotation, etc..). [12]

7.2. L'image Fractale :

➢ Une image fractale est une figure géométrique ou est un ensemble fini de transformations géométriques (rotation, translation, agrandissements, réductions) appliquées aux sous ensemble et motifs identiques et de tailles variables qui la composent. [13]

7.3. Objet fractal :

➢ C'est un objet naturel dont la forme est extrêmement irrégulière et de plus (éventuellement) interrompue et fragmentée. [13]

7.4. Auto- similarité :

➢ Un objet est dit auto - similaire si sa forme est la même à n'importe quel grossissement : un détail de l'objet une fois agrandie, reproduit exactement une partie plus grande de l'objet. Cependant il existe des objets fractals qui ne sont pas exactement auto- similaires, l'objet ne contient pas une transformation de lui-même, mais des parties de lui-même, c'est ce qu'on appelle l'auto- similarité locale. [13]

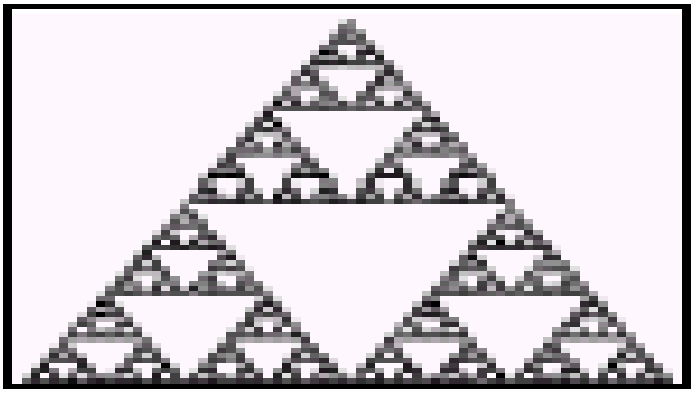


Figure 6 : Triangle de Sierprinski [7]

7.5. Les méthodes de compression fractale :

➢ Il existe plusieurs méthodes de compression fractale, et nous avons choisis de travailler sur l'un d'entre elles :

- La compression Jacquin.
- La compression par subdivisions successives de triangles.
- La compression de Delaunay.

➢ Ces trois algorithmes sont fondés sur une même idée :

- Définir une zone dans l'image (un carré ou un triangle suivant la méthode utilisée),
- Appliquer des transformations fondamentales (réduction, normalisation de la moyenne, rotation etc...)

- Réimplanter le résultat dans l'image dans une zone plus réduite (Destination). La zone de destination doit être la plus 'proche' possible du résultat.
- La répétition de ce modèle dans toute l'image, et ceci de manière itératif, implique une certaine convergence de l'image reconstruite par l'assemblage de transformations/réductions locales, vers l'image d'origine. [12]

7.5.1. Méthode Jacquin :

➢ Principe :

➢ Le principe de l'algorithme de compression par la méthode Jacquin est :

- Correspond parfaitement à l'algorithme générique présenté dans le deuxième chapitre.
- Utilise des régions carrées pour segmenter l'image.
- Cet algorithme de compression/décompression est le plus simple que les trois algorithmes présentés précédemment.
- Consiste a manipulé des régions carrées prédéfinies (tableaux 2D), et que cette dernière est beaucoup plus simple à implémenter que les régions triangulaires dynamiques.
- La méthode Jacquin consiste quatre étapes importantes nous verrons par la suite dans le chapitre 2. [12]
- La Figure 7 présente le principe fonctionnement de la compression fractale par la méthode de Jacquin.

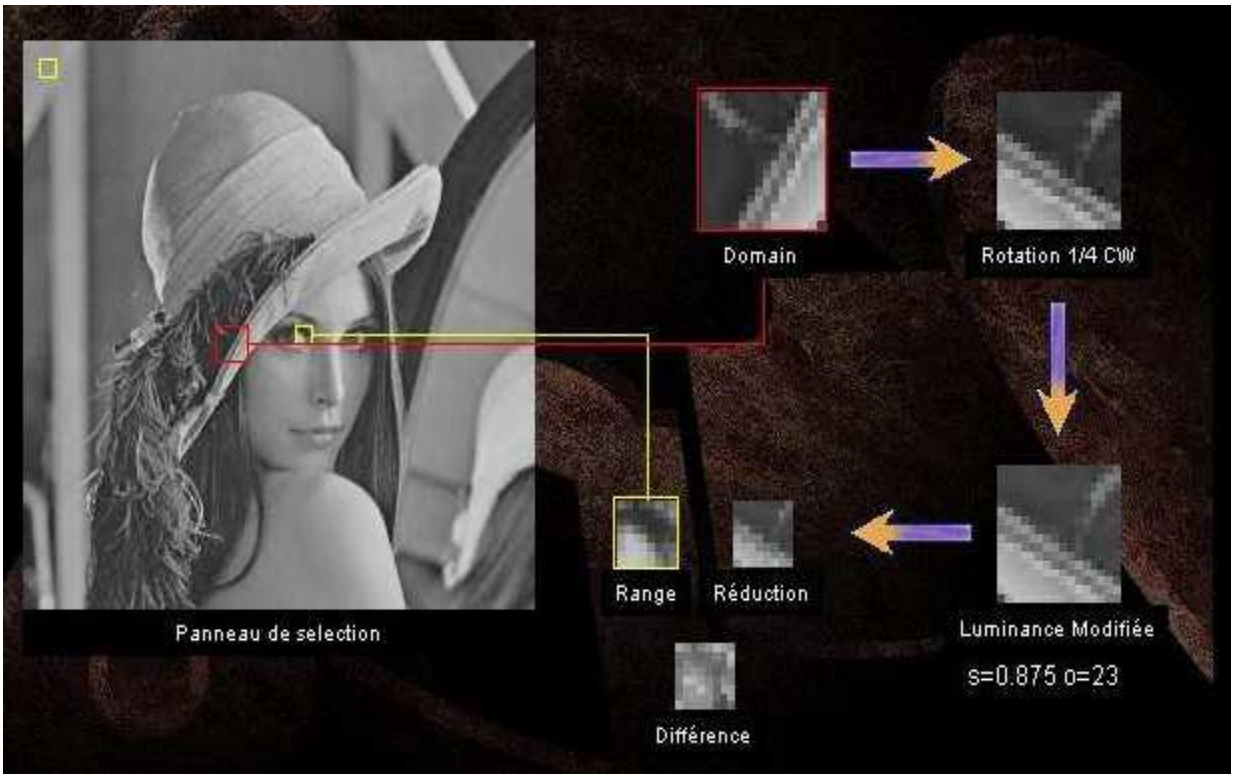


Figure 7: Principe de méthode Jacquin [12]

7.5.2. Méthode par subdivisions de triangles :

➢ Principe :

- La méthode par subdivisions de triangles est quasiment identique à la méthode Jacquin. Elles possèdent cependant deux différences majeures. D'une part, nous travaillons maintenant sur des triangles. Et d'autre part, un module de subdivision des triangles a été ajouté. Mais avant de présenter la subdivision de triangles en elle-même, il est n´nécessaire d'évoquer la méthode utilisée pour gérer les triangles.
- Travailler sur un carré est relativement simple, il suffit de définir la hauteur du carré, et un point de d´épart (en haut à gauche en général). Un triangle est plus complexe, il faut soit 3 points, soit 3 droites.
- Ainsi, pour savoir si un point (ou un pixel) appartient ou non à un triangle, nous avons choisi de modéliser un triangle à l'aide de 3 droites d1, d2 et d3, comme le montre la Figure 8. [12]

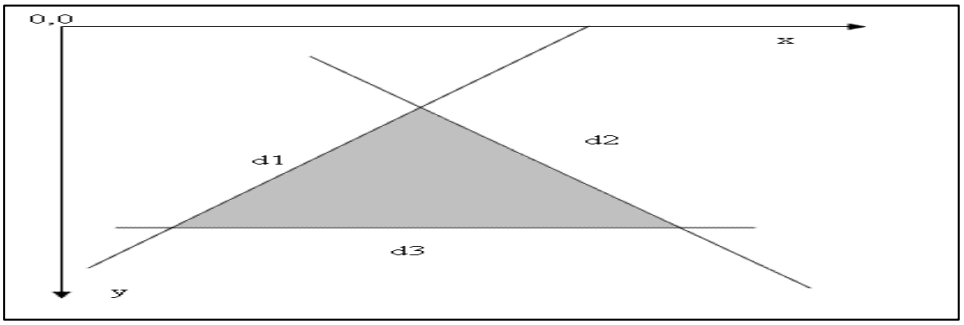


Figure 8 : Appartenance d'un point à un Triangle [12]

- Pour connaître l'appartenance d'un point à un triangle, il suffit de vérifier les équations définies ci-dessous :

$$P.x \geq d1 \quad P.y \geq d1$$
$$P.x \geq d2 \quad P.y \geq d2$$
$$P.x \geq d3 \quad p.y \geq d3$$

- Si un point P (de coordonnées (P.x,P.y)) appartient au triangle formé par les segments d1, d2 et d3, alors il vérifie ces inéquations.

7.5.3. La subdivision :

➢ Deux points essentiels pour lancé l'opération de subdivision :

- Découpé l'image à traiter en un pavage de triangles. Pour améliorer l'efficacité de l'algorithme de compression, une division des triangles ayant un nombre important de détails est ajoutée.
- Pour connaître le niveau de détails d'une zone de l'image (couverte par un triangle considéré), nous calculons simplement l´écart type des pixels de celui-ci. Plus cette valeur sera importante, et plus le triangle est loin d'être uniforme. [12]

7.5.3.1. Les étapes de subdivision de Triangles :

➢ Le triangle sera divisé en 6 plus petits triangles comme le montre la Figure suivante :

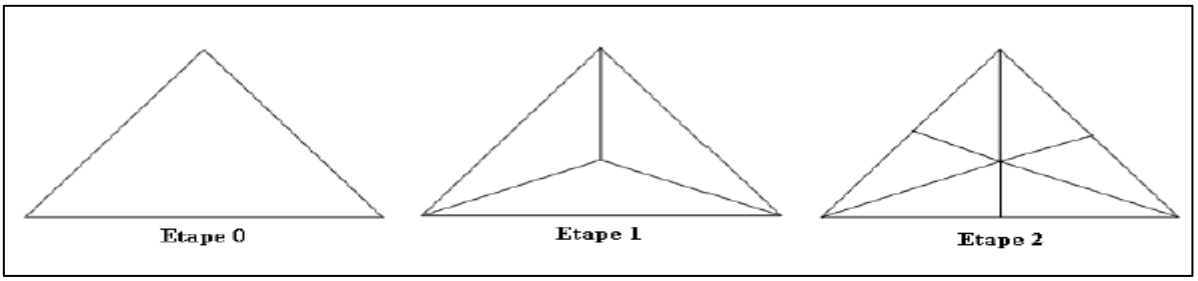


Figure 9 : Subdivisons de Triangles [12]

- **L'étape 1 :** nous division une première fois le triangle en 3 triangles, qui seront formés d'une part, avec les 3 points du triangle parent, et d'autre part avec son barycentre.
- **L'étape 2 :** va ensuite scinder les 3 nouveaux fils en 2, en utilisant la médiane (partant de l'ancien barycentre) de chacun.
- **L'étape 3 :** Cette subdivision amélioré considérablement le rendu de l'image compressée. La perte de qualité est diminuée, mais le temps de compression impose une profondeur de division restreinte. De plus, la taille du fichier de sortie augmente car il est n´nécessaire de stocker les différentes subdivisions. Un simple booléen suffit (1 : triangle divisé, 0 : triangle non subdivisé). [12]

➢ Transformation d'un triangle :

- Le processus discrétisé d'une réduction, qu'elle soit avec un carré (méthode de Jacquin) ou avec un triangle, reste identique. Il s'agit simplement de caractériser une matrice de transformation des coordonnées d'une figure de d´épart, vers les coordonnées d'une figure d'arrivée. Prenons l'exemple de deux triangles montré dans la Figure 10 : [12]

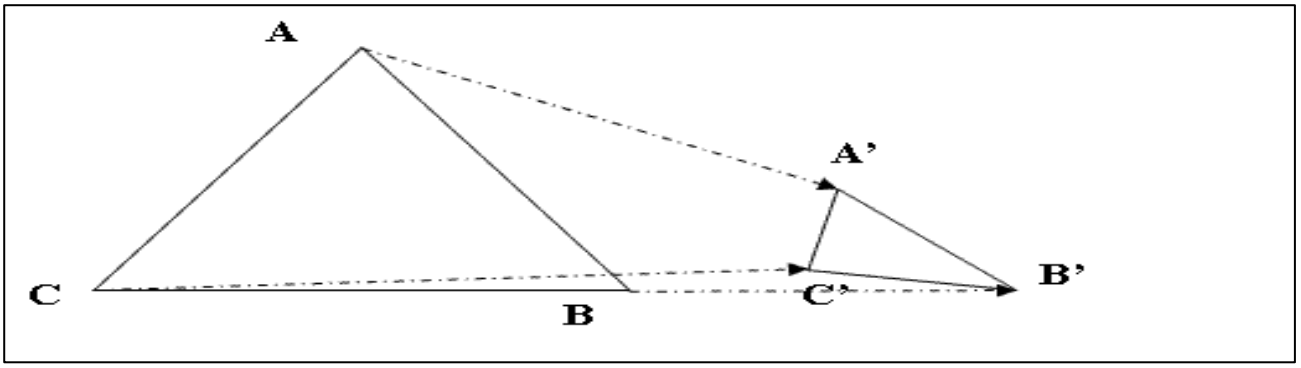


Figure 10 : Transformation d'un Triangle [12]

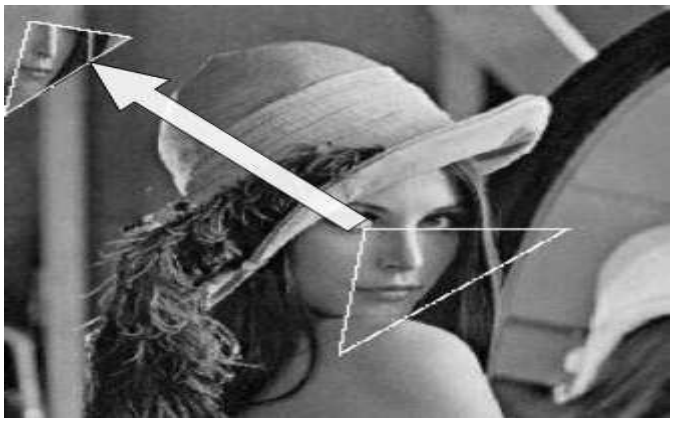


Figure 11 : Réduction d'un Triangle [12]

➢ Partitionnements :

- Le partitionnement des figures Sources et Destinations par la méthode de subdivision de triangles sont identiques à ceux de la méthode de Jacquin. Il s'agit ici, de réaliser un pavage de triangles (et non plus de carrés), avec un pas déterminée. Le pavage Destination est bien entendu plus compact que le pavage Source, pour conserver le caractère fractale de la compression. [12]
- La Figure suivante montre les deux pavages Source et Destination sur l'image lena.jpg.

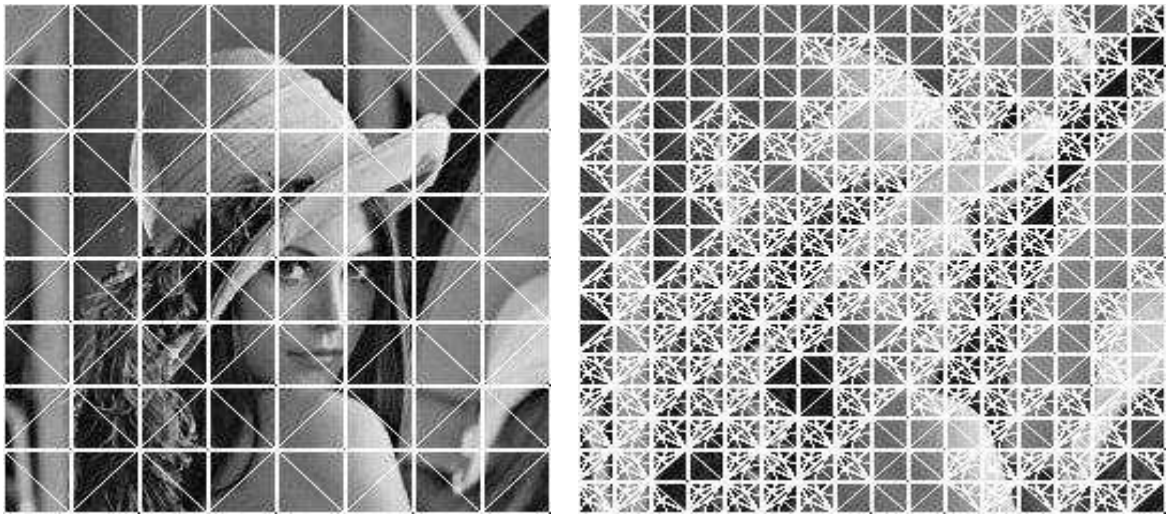


Figure 12 : Pavage Source et Destination, Méthode par subdivisions se Triangles [12]

7.5.4. Méthode de Delaunay :

➢ Principe :

➢ La dernière méthode de compression fractale, est la compression de Delaunay. Cette méthode manipule tout comme la Subdivision de triangles, des figures triangulaires pour segmenter l'image. Mais au lieu de partitionner suivant un pavage régulier de triangles, la méthode de Delaunay utilise comme son nom l'indique, un pavage formé par les triangles de Delaunay. Seul, le pavage Destination est formé ainsi, le partitionnement Source étant un simple pavage réguler.

➢ Pour augmenter encore la qualité de restitution des détails lors de la reconstruction de l'image, l'algorithme de compression de Delaunay segmente l'image en triangles de Delaunay. [12]

7.5.4.1. Triangulation de Delaunay :

➢ La construction des triangles de Delaunay à partir d'un diagramme de Voronoi est assez simple. En effet, l'algorithme de Delaunay recherche premièrement les germes voisins (donc séparés par un arrête de Voronoi), pour chaque germe du Diagramme, et les relie entre eux. Comme le montre la Figure suivante : [12]

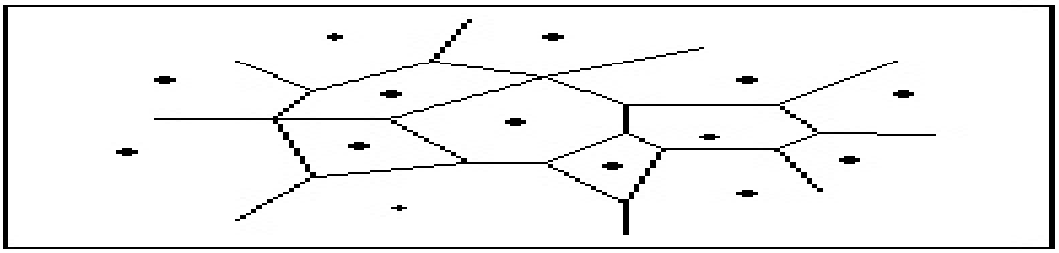


Figure 13 : Digramme de Voronoi [12]

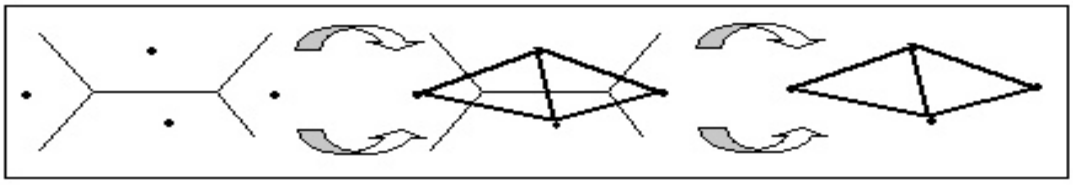


Figure 14 : Du diagramme de Voronoi aux Triangles de Delaunay [12]

➢ La conception de ce genre de segmentation étant assez complexe à réaliser.

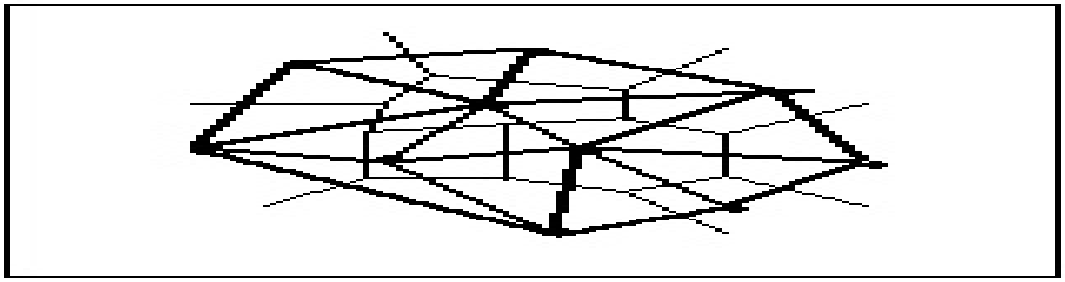


Figure 15 : Diagramme de Voronoi, Partitions de Delaunay [12]

- La Figure 16 présente la Pavage source et destination selon la méthode de Delaunay, ce que nous pouvons dire sur la figure de droite est que les zones homogènes sont composées de triangles assez grands (notamment au-dessus du chapeau de Lena, et à droite de l'image). Les détails sont quant à eux localisés par de plus petits triangles (les cheveux de Lena). [12]

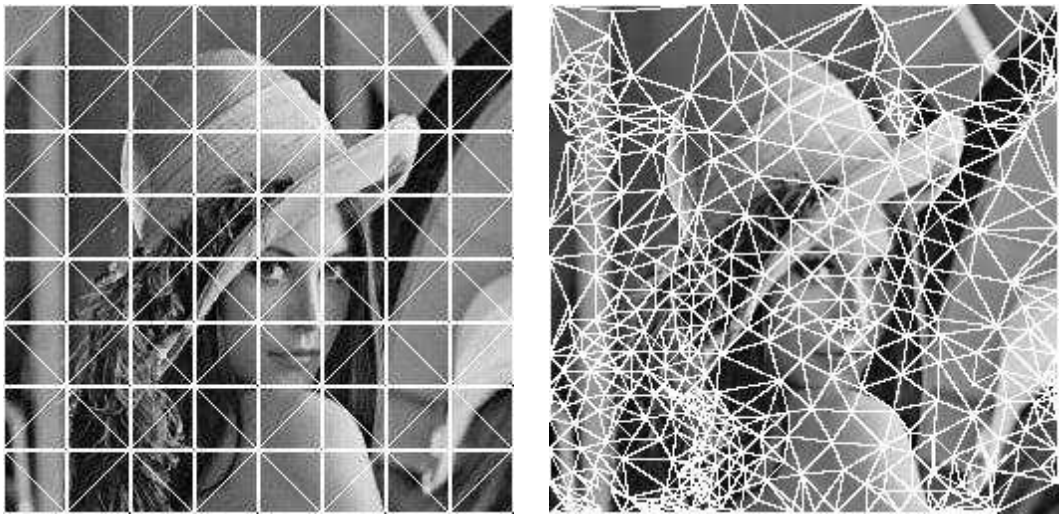


Figure 16 : Pavage Source et Destination, Méthode de Delaunay [12]

II. La Cryptographie :

1. Définition :

➢ La cryptographie est la science qui utilise les mathématiques pour le cryptage et le décryptage de données. Elle vous permet ainsi de stocker des informations confidentielles ou de les transmettre sur des réseaux non sécurisés (tels que l'Internet), afin qu'aucune personne autre que le destinataire ne puisse les lire.

➢ Alors que la cryptographie consiste à sécuriser les données, la cryptanalyse est l'étude des informations cryptées, afin d'en découvrir le secret. La cryptanalyse classique implique une combinaison intéressante de raisonnement analytique, d'application d'outils mathématiques, de recherche de modèle, de patience, de détermination et de chance. Ces cryptanalyses sont également appelés des **pirates**[9]**.** [14]

Cryptologie = Cryptographie + Cryptanalyse.

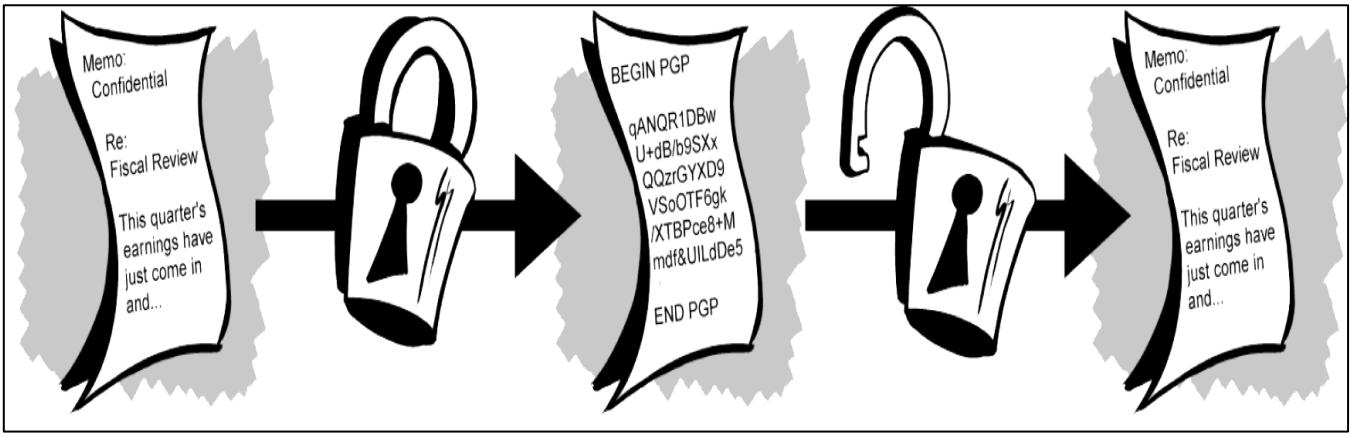


Figure 17 : Schéma générale de la Cryptographie [14]

[9] **Pirate :** un pirate ou hacker en sécurité informatique, un hacker, francisé hackeur ou hackeuse, est un spécialiste d'informatique, qui recherche les moyens de contourner les protections logicielles et matérielles.

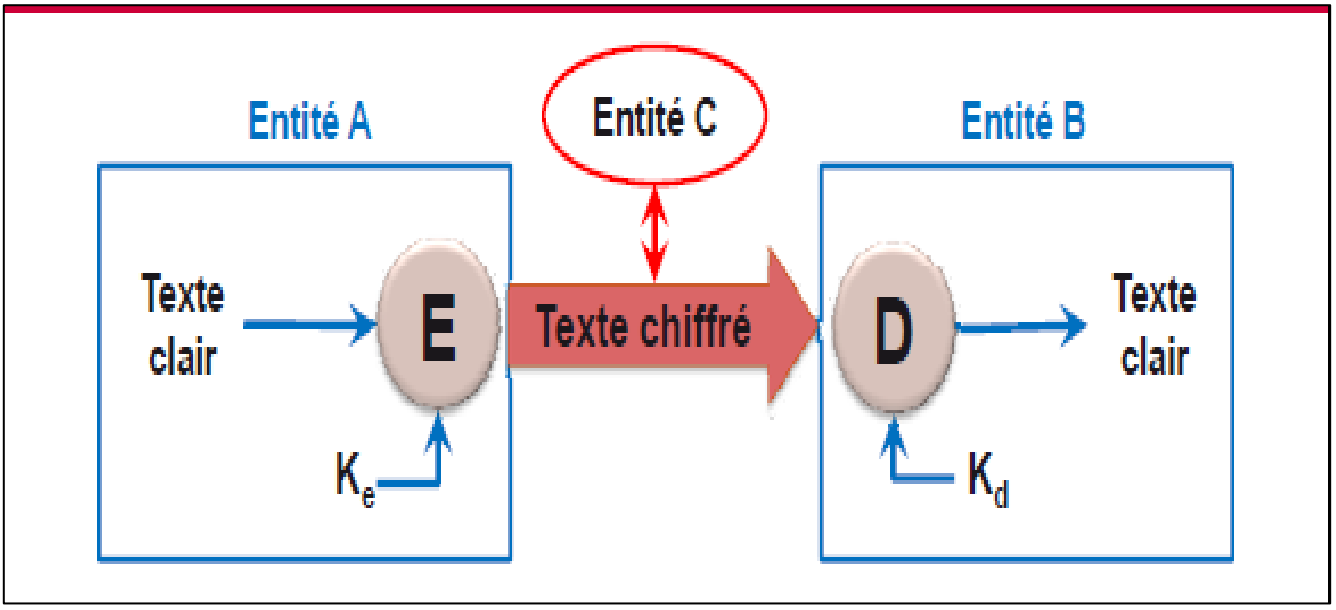


Figure 18 : Schéma de processus de cryptage et décryptage [15]

2. Les objectifs de la cryptographie

- Les principaux services à garantir par l'application de la cryptographie sont:
  - **Confidentialité de l'information** : le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé « Organisation internationale de normalisation (**ISO**[10]) »
  - **Intégrité de l'information** : L'information ne doit subir aucune altération ou destruction volontaire ou accidentelle, lors de son traitement, conservation ou transmission.
  - **Détection de l'altération de l'information par une entité non-autorisée.**
  - **Non-répudiation** : Empêcher qu'une entité réfute des actions ou engagements antérieures.
    - Vérifier que le expéditeur et le destinataire sont les entités qui ont envoyé ou reçu l'information. [15]

- **Les lois de Shannon** : Claude Elwood Shannon (1916-2001) est considéré comme le père de la théorie de l'information. Il a établi le lien entre l'algèbre de Boole et la commutation électronique, posant ainsi les bases des systèmes numériques actuels. C'est à lui que l'on doit la notion de (bit). Il s'est illustré dans de nombreux domaines, notamment la cryptographie. [16]

[10] **ISO** : International Organisation for Standardisation). Organisation internationale de standardisation regroupant les organismes similaires de 89 nations. L'ISO se charge des standards qui régissent l'Internet actuellement.

3. Cryptographie classique :

➢ La cryptographie classique a été conçue avant la création des ordinateurs et qui ont donné les concepts et les bases pour l'évolution de plusieurs algorithmes symétriques encore utilisés de nos jours.

➢ Les **crypto-systèmes** [11]classiques sont groupés en chiffrement mono alphabétique et poly alphabétique Le chiffrement mono alphabétique est très primaire, il s'agit d'une substitution simple. Chaque lettre est remplacée par une autre lettre ou symbole conformément à un certain algorithme. [17] [15]

3.1. Classification :

3.1.1. Chiffrement par décalage:

➢ Un des systèmes les plus anciens et simples est le chiffrement par décalage appel aussi le code de César. Ce code est un des plus anciens. Le principe est de remplacer dans un message une ou plusieurs entités (ex : lettres) par une ou plusieurs autres entités. [17] [15]

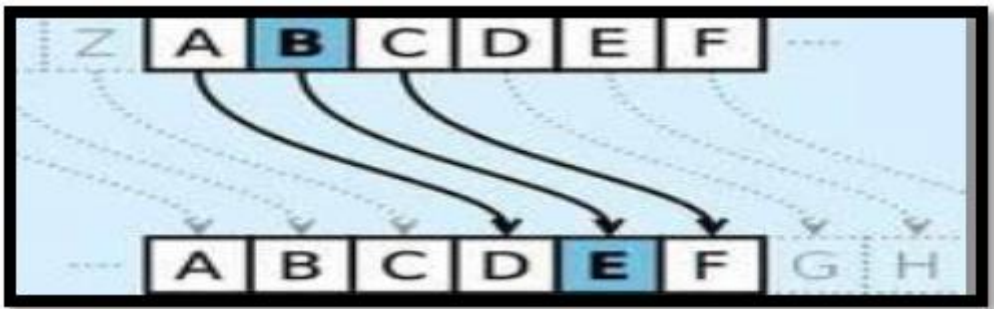


Figure 19 : Principe de code de César [17]

3.1.2. Chiffrement par substitution :

➢ Il s'agit d'une méthode plus générale qui englobe le chiffrement par décalage. En effet, à chaque lettre de l'alphabet on fait correspondre une autre, c'est-à-dire que l'on effectue une permutation de l'ensemble des lettres.

➢ Pour la première lettre a, on a 26 possibilité de substitutions. Pour la lettre b, on n'en a plus que 25 et ainsi de suite. [17] [15]

3.1.3. Le code de Vigenère :

➢ Le chiffrement de Vigenère est une amélioration décisive du chiffre de César. Il a été élaboré par Blaise de Vigenère (1523-1596), diplomate français du XVIe siècle. [14] [15]

- Substitution poly-alphabétique.
- Basé sur la table de Vigenère.
- La colonne correspondante à la lettre en clair.

[11] **Un crypto-système** est un terme utilisé en cryptographie pour désigner un ensemble composé d'algorithmes cryptographiques et de tous les textes en clair, textes chiffrés et clés possibles.

- La ligne correspondante à une lettre de la clé.
- la lettre chiffrée est le croisement de la ligne et de la colonne.
- La clé est répétée boucle autant que nécessaire.

➢ La Figure 20 explique le principe du fonctionnement du code Vigenére.

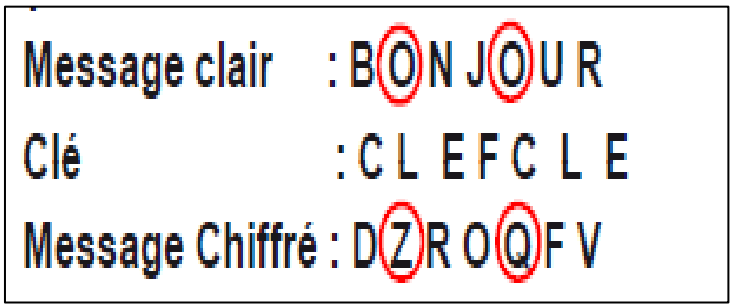


Figure 20 : Exemple sur le code Vigenére [15]

3.1.4. Chiffrement de Vernam :

➢ Une réalisation célèbre de la confidentialité parfaite est le chiffrement de Vernam (voir Figure 21), également connu sous les noms masque jetable ou one-time pad. Il fut inventé par Gilbert Vernam en 1917 pour chiffrer et déchiffrer des messages télégraphiques. [18]

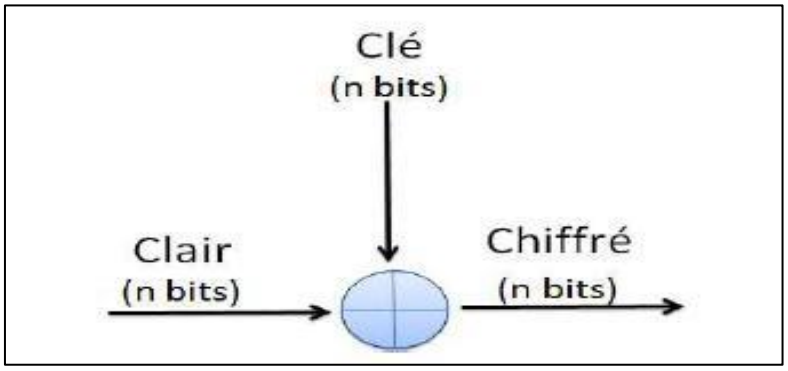


Figure 21 : Chiffrement de Vernam [18]

4. Cryptographie moderne :

➢ La cryptologie moderne a pour objet l'étude des méthodes qui permettent d'assurer les services d'intégrité, d'authenticité et de confidentialité dans les systèmes d'information et de communication. La cryptologie se partage en deux sous-disciplines :

- La cryptographie qui propose des méthodes pour assurer ces services.
- La cryptanalyse qui recherche des failles dans les mécanismes proposés.

➢ Pour assurer les objectifs de la cryptographie moderne nous pouvons utiliser des algorithmes basés sur des clés. Ces algorithmes sont définis par plusieurs types de cryptographie moderne, on distingue deux approches :

- La cryptographie symétrique.
- La cryptographie asymétrique.

4.1. La cryptographie Asymétrique :

➢ Le chiffrement asymétrique (ou chiffrement à clés publiques) consiste à utiliser une clé publique pour le chiffrement et une clé privée pour le déchiffrement. [15] [17]

4.1.1. Principe :

➢ Pour mieux comprendre le principe de la cryptographie asymétrique (voire la Figure 22).

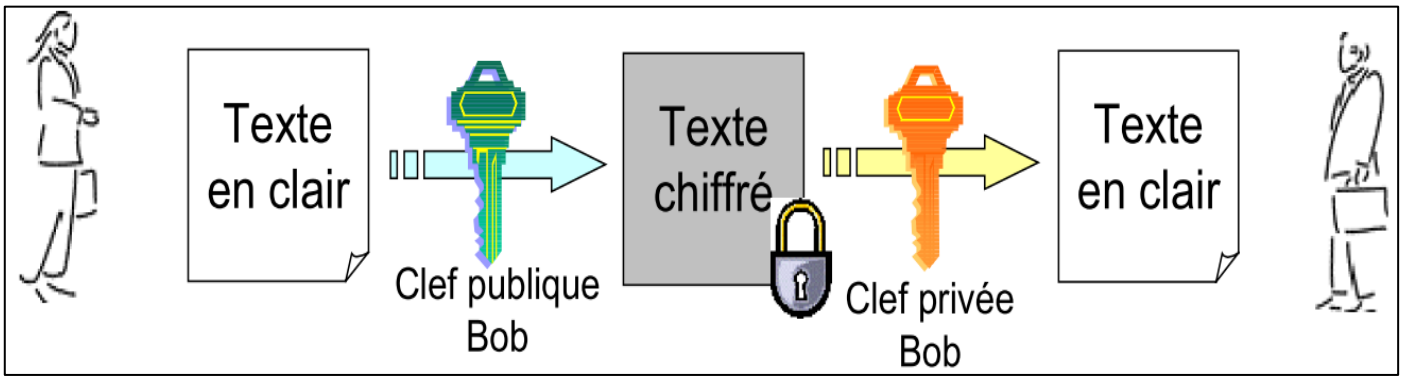


Figure 22 : Principe de chiffrement Asymétrique [17]

- Pas nécessaire d'échangé la clé secrète entre les deux interlocuteurs, seule la clé publique est partagé au travers d'un canal **non sécurisé**[12].
- Une paire de clés : clé publique connue de tous et clé privée connue que de son propriétaire: lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire.
- Chiffrement requiert beaucoup d'opérations et n'est pas recommandé pour de grande quantité d'informations.
- Algorithme de chiffrement **RSA**[13].

4.1.2. Les avantages du cryptage asymétrique :

- L'élimination de la problématique de la distribution de la clé privée.
- La possibilité d'utiliser la signature électronique.
- L'impossibilité de décrypter le message dans le cas de son interception par une personne non autorisé.
- Une paire de clés (publique/secrète) peut être utilisée plus longtemps qu'une clé symétrie.

[12] **Un canal non sécurisé** est non crypté et peut être sujet à l'écoute clandestine. Des communications sécurisées sont possibles sur un canal non sécurisé si à communiquer le contenu est crypté avant la transmission.

[13] **RSA** : Le chiffrement RSA est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman.

4.1.3. Les inconvénients du cryptage asymétrique :

- Le temps d'exécution : plus lent que le cryptage symétrique.
- Le danger des attaques par substitution des clés (d'où la nécessité de valider les émetteurs des clés).
- Taille des clés, plus grand que celle des systèmes symétriques (>512 bits).

4.2. La cryptographie symétrique :

➢ Le chiffrement symétrique (aussi appelé chiffrement à clé privée ou à clé secrète), ce type se base sur l'utilisation d'une clé pour crypter et décrypter les messages. La sécurité de cette solution repose sur le fait que la clé est connue uniquement par l'émetteur et le récepteur du message.

4.2.1. Principe :

➢ Pour mieux comprendre le principe de la cryptographie asymétrique (voire la Figure 23).

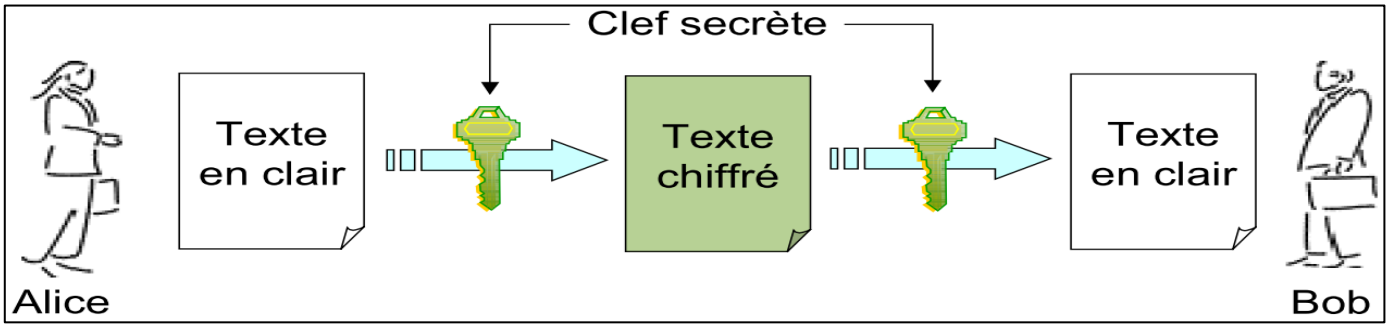


Figure 23 : Principe de chiffrement symétrique [17]

- L'échangé la clé secrète entre les deux interlocuteurs doit s'effectuer à travers un canal sécurisé ou sécuritaire
- Le chiffrement symétrique consiste à appliquer une opération (algorithme) sur les données à chiffrer à l'aide de la clé privée, afin de les rendre inintelligibles.
- On donne le message à quelqu'un et on lui fournit la clé privée, pour qu'il puisse déchiffrer le message.
- Le cryptage symétrique fonctionne selon deux procédés catégories : par Bloc/ par Flots.

4.2.2. Le cryptage par flot (Stream Cipher) :

➢ Les algorithmes de chiffrement de flux peuvent être définis comme étant des algorithmes de chiffrement par blocs, où le bloc a une dimension unitaire (1 bit, 1 octet, etc.) ou relativement petite. Leurs avantages principaux viennent du fait que la transformation (méthode de chiffrement) peut être changée à chaque symbole du texte clair et du fait qu'ils soient extrêmement rapides.

- Quelques algorithmes de cryptographie symétrique par flot:
  - A5: utilisé dans les téléphones mobiles de type GSM pour chiffrer la communication par radio entre le mobile et l'antenne-relais la plus proche.
  - RC4, le plus répandu, conçu en 1987 par Ronald Rivest l'un des inventeurs du RSA, pour les Laboratoires RSA, utilisé notamment par le protocole WEP, un algorithme récent de Eli Biham – E0 utilisé par le protocole Bluetooth. [15]

4.2.3. Le cryptage par bloc (Block Cipher) :

- C'est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique. Le principe de cryptage par Blocs est-il consiste de deviser le message en bloc de bits de longueur fixe (souvent une puissance de deux comprise entre 32 et 512 bits) (voir la Figure 24). Les blocs sont ensuite chiffré ses uns après les autres. [15]
- Deux algorithmes très connus DES et AES.

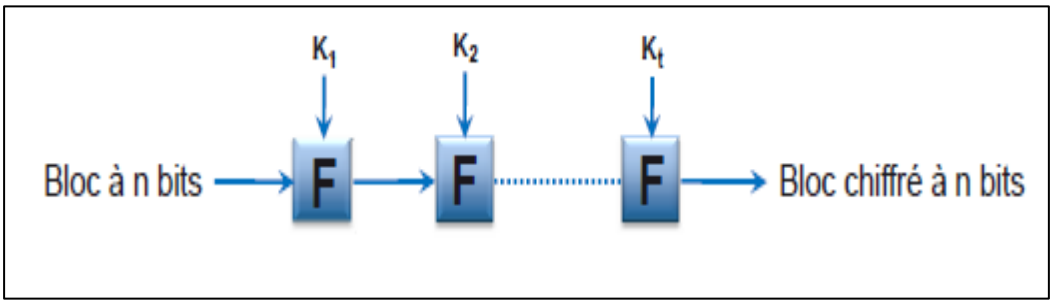


Figure 24 : Principe de chiffrement par Bloc [15]

- Le chiffrement par bloc utilise quatre modes opératoires :
  - Electronic Code Book(ECB)
  - Cipher Block Chaining (CBC).
  - Output Feedback (OFB).
  - Cipher Feedback(CFB).

- **Electronic Code Book(ECB) :** Dictionnaire de codes Un message réel en générale composé de nombreux blocs. La façon plus immédiate pour chiffrer un tel message est de chiffrer successivement chaque bloc, avec la même clé. Comme montrer dans la Figure 25.

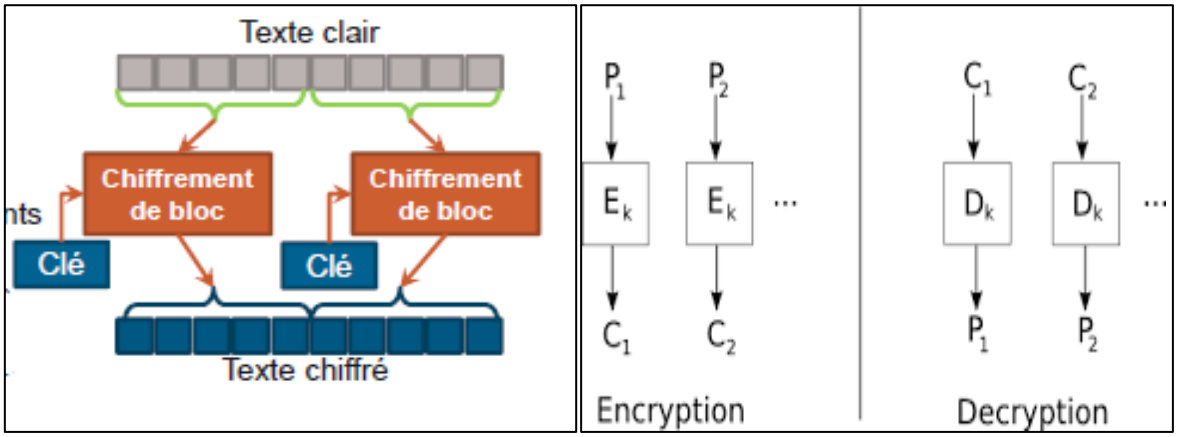


Figure 25 : Le chiffrement par Bloc mode ECB [15]

➢ **Cipher Block Chaining (CBC) :** L'enchaînement des blocs consiste, avant le chiffrement d'un bloc, à le masquer par le résultat du chiffrement du bloc précédent au moyen de l'opération XOR. Le premier bloc clair est lui aussi masqué, par une valeur habituellement notée IV (Initial Value) et de préférence variable (la date et l'heure peuvent faire une bonne (IV) pour que les chiffrements successifs du même message soient différents. La valeur initiale IV n'a pas besoin d'être secrète, et elle est en générale transmise en clair avant le message chiffré. Noter que si le destinataire reçoit un bloc chiffré avec des bits erronés, cela affecte le déchiffrement de ce bloc et du suivant mais pas des autres (Voir la Figure 26). [15]

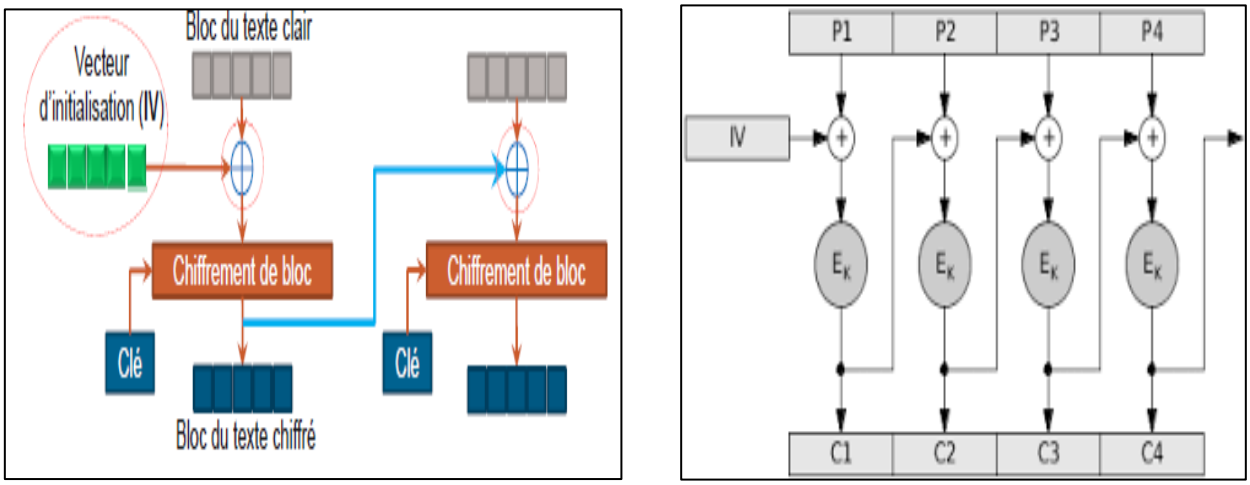


Figure 26 : Le Chiffrement par Bloc mode CBC [15]

Figure 27 : Le chiffrement par ECB et CBC [15]

4.2.4. Les avantages du cryptage symétrique :

- ✓ La rapidité d'exécution (jusqu'à 100 fois plus rapide que les solutions asymétriques)
- ✓ .La simplicité d'implémentation (gestion d'une seule clé).
- ✓ Clés relativement courtes (entre 64-128 bits).

4.2.5. Les inconvénients du cryptage symétrique :

- Gestion des clés difficiles (nombreuses clés).
- Point faible = l'échange de la clé secrète.
- Dans un réseau de N entités susceptibles de communiquer secrètement il faut distribuer N*(N-1)/2 clés.

III. Conclusion :

- Dans ce chapitre on a vu que la compression s'impose comme une étape incontournable pour optimiser l'utilisation des grands volumes d'informations dans les réseaux informatiques. L'objectif principal de la compression d'image est de réduire la quantité d'information nécessaire à une représentation visuelle fidèle à l'image originale. En générale on différencie les méthodes de compression selon la perte d'informations. Les méthodes réversibles, utilisent uniquement le principe de la réduction de la redondance et n'engendrent pas de perte. Celles irréversibles, définissent une représentation approximative de l'information. La compression fractale d'image est un outil assez récent pour le codage des images. Elle est basée sur l'auto similarité locale et la génération des copies de blocs sur la base des transformations mathématiques. La technique paraît intéressante dans théorie et l'application mais l'inconvénient majeur est dans l'usage du temps dû à la haute exigence de ressources quand on chiffre de grandes masses de données.
- D'autre part, on a survolé la cryptographie, ses différents types et les propriétés de chaque type, tout en mettant l'accent sur la cryptographie symétrique précisément le Block Cipher.
- La combinaison de ces deux techniques, à savoir la compression et le cryptage, et le choix de l'algorithme de compression et de cryptage pour en faire un système hybride de crypto-compression, tout ça sera détaillé dans le deuxième chapitre.

DEUXIEME CHAPITRE

CRYPTO-COMPRESSION

1. Introduction :

➢ La réduction de la taille des bits présentés sur l'image est devenue de plus en plus important dans le stockage ensuite la confidentialité de cette dernière, la cryptographie aussi est devenue un critère très important dans la transmission sécurisé des images. Pour assure ces deux critères on a besoin a des techniques qui combinent les deux technologies la compression et le cryptage des images qui ont signifié aux techniques de Crypto-compression qui ont pris de l'essor ces dernières années. Le concept vise à combiner à la fois les techniques de cryptage et de compression de manière jointe. L'objectif est de procurer un volume de données de taille réduite avec une confidentialité robuste. Le travail présenté dans ce chapitre porte sur une approche de crypto-compression à base de Block Cipher et d'un algorithme de compression fractale. Nous allons commencer tout d'abord par présenter l'algorithme générique de compression et décompression fractale. Ensuite nous présentons une approche de compression par la méthode de Jacquin et nous allons présenter l'approche de cryptage à base de Block Cipher en utilisant l'algorithme AES.

2. Compression fractale par blocs :

➢ Sur la base des IFS14, Jacquin a proposé en 1989 une approche fractale ne nécessitant pas d'intervention humaine et permettant de coder une image naturelle. La méthode était basé sur une série de transformation affines contractantes et ''locales'' définissant un opérateur contractant noté W. Elle a l'avantage d'être automatique mais ne résout pas directement le problème inverse des IFS puisque l'image n'est pas vue comme une union de transformation d'elle-même mais comme une union de sous-partie transformées d'elle-même. [19]

2.1. Transformation contractive :

➢ Une transformation est dite contractive si pour tous deux points P1 et P2 la distance :

$$d(w(P1), w(P2)) < S * d(P1, P2) \; tel \; que \; 0 < S < 1$$

➢ Cette transformation contractive appliquée à deux points va les rapprocher. Cette définition est absolument général, elle s'applique à tout espace métrique (espace sur lequel on peut définir une distance d (P1, P2)), si les points ont pour coordonnés P1=(x1, y1) et P2=(x2, y2) alors :

$$d(P1, P2) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad [13]$$

[14] **Les IFS (**Iterated Function System) ont été introduits en 1985 par M.F Barnsley et S.Denko.

2.2. La définition d'un IFS :

➢ Un IFS consiste en une collection de transformations contractantes$\{ \text{wi}: \mathrm{R}^2 \rightarrow \mathrm{R}^2 / \mathrm{i} = 1, \ldots, \mathrm{n}\}$, Cette collection de transformations définit une fonction :

$$W(.) = \cup_{i=1}^{N} W(.) \text{ [13]}$$

➢ La fonction W n'est pas appliquée au plan, mais à des ensembles de points du plan, Hutchinson a démontré un fait important pour cette fonction [5] est que, quand les wi sont contractantes dans le plan, alors W est contractant dans l'espace de l'ensemble de points (du plan). [13]

2.3. Théorème du point fixe :

➢ Ce théorème formalise une constatation intuitive, si une fonction est contractante, alors lorsqu'on l'applique de façon récurrente à partir d'un point initial, on converge vers un unique point fixe. [13]

2.4. Les IFS comme outils de compression :

➢ M. Barnsley suggéra que de stocker les images comme code IFS (collection de transformations affines contractantes) puisse amener la compression de l'image. Par exemple la feuille de fougère est générée par seulement quatre transformations affines chaque transformation affine est définie par 6 nombres (a, b, c, d, e, f) qui ne nécessitent pas beaucoup de mémoire pour être stockés, elles peuvent être stockées dans 4(transformations)×6(nombres)×32 bit (pour chaque nombre)=768 bits alors que l'image sous forme de pixels requiert beaucoup plus de mémoire au moins 65 536 bits.

➢ On voit que le stockage des images sous formes de collection de transformations contractantes amène directement la compression puisque le stockage d'une image sous formes de pixels nécessite beaucoup de place mémoire que de stocker l'image sous forme de collection de transformations contractantes. [13]

2.5. Codage :

➢ La compression d'une image par fractales repose sur une transformation fractale qui consiste à transformer l'image à l'aide d'un opérateur finalement contractant W, de manière à ce que son aspect visuel reste quasiment inchangé. Pour cela, l'opérateur W est constitué de N sous-transformation élémentaires $\mathrm{w_n}$, chacune opérant sur un bloc de l'image (Figure 28). [19]

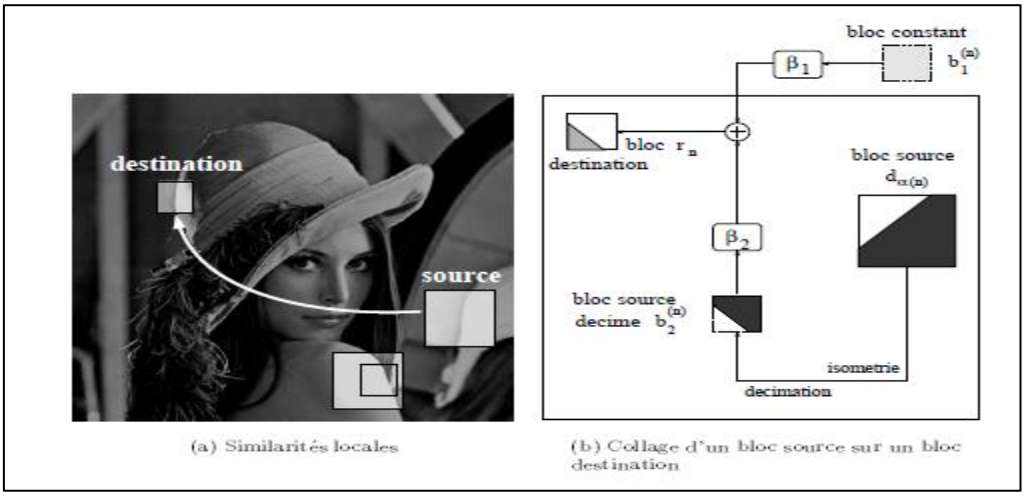


Figure 28 : Principe de codage par fractales [19]

- L'image A est partitionné en N blocs r_n appelés blocs destination. Chaque bloc destination est ensuite mis en correspondance avec un autre bloc $d_{\alpha(n)}$ au moyen d'une transformation locale w_n. Le bloc $w(d_{\alpha(n)})$ ressemble au bloc r_n au sens d'une mesure d'erreur (quadratique) sur les niveaux de gris. Le bloc $d_{\alpha(n)}$, appelé bloc source est recherché au travers d'une librairie composée de Q blocs ne forment pas nécessairement une partition de l'image mais sont représentatifs de toute l'image. La transformation W de l'image A est ainsi formulée de façon suivante :

$$\boldsymbol{W(A) = \cup_{n=1}^{N} W_n(d\alpha_n) = \cup_{n=1}^{N} \hat{r}_n \simeq \cup_{n=1}^{N} r_n = A}$$

- Où $\hat{r}_n$ est l'approximation du bloc destination r_n obtenue en transformation le bloc source $d_{\alpha(n)}$ par w_n . La transformation w_n a pour effet de coller le bloc source $d_{\alpha(n)}$ sur le bloc destination r_n, en déformant son support (isométrie, translation) es en modifiant sa fonction de luminance. L'approximation du bloc r_n, noté $\hat{r}_n$, est donnée par une combinaison linéaire de deux blocs, parmi lesquels le bloc $d_{\alpha(n)}$ est extrait de l'image elle-même.
- On comprend aisément que la difficulté réside dans la contraction, recherche des similarités et codage des transformations w_n, sur la base de partitions de l'image. [19]

3. Algorithme générique de compression et décompression :

Algorithm 1 Algorithme générique de la compression fractale.

```
1: Fonction COMPRESSION
2:     Créer un pavage de figures Sources
3:     Créer un pavage de figures Destinations
4:     Pour toutes les figures Destinations Faire
5:         Pour toutes les figures Sources  Faire
6:             Pour toutes les transformations définies Faire
7:                 Appliquer la transformation à la figure Source
8:                 Ajuster la moyenne des couleurs des pixels
9:                 Appliquer la réduction de la figure Source vers la figure Destination
10:                Calculer l'erreur entre le résultat et la figure de destination
11:                Si l'erreur est minimale pour la figure de destination Alors
12:                    Sauver les modifications effectuées
13:                FinSi
14:            Fin Pour
15:            Ecrire dans le fichier de sortie les valeurs sauvées
16:        Fin Pour
17:    Fin Pour
18: Fin Fonction
```

Figure 29 : Algorithme générique de compression [12]

Algorithm 2 Algorithme générique de la décompression fractale.

```
1: Fonction DÉCOMPRESSION
2:     Créer une image
3:     Créer un pavage de figures Sources
4:     Créer un pavage de figures Destinations
5:     Pour toutes les itérations désirées Faire
6:         Pour toutes les figures Destinations Faire
7:             Lire dans le fichier les transformations à appliquer
8:             Lire le numéro de la figure Source à manipuler
9:             Appliquer les transformations à la figure précédemment évoquée
10:            Appliquer la réduction de la figure Source vers la figure destination
11:            Inserer le résultat dans l'image
12:        Fin Pour
13:    Fin Pour
14: Fin Fonction
```

Figure 30 : Algorithme générique de décompression [12]

- Ces algorithmes sont génériques. Les trois méthodes que nous avons vu dans le chapitre précédente dans ce document, ajoutent leurs propres fonctionnalités afin d'améliorer le taux de compression. [12]

4. Méthode Jacquin :

4.1. Fonctionnement :

- ✓ [etape0] : On commence par générer sur l'image un pavage de carrés appelés Sources et un pavage de carrés Destinations. Puis, on cherche quelle est la meilleure combinaison Source/Destination pour chaque figure Source. Pour cela l'étape 01.
- ✓ [Étape 1] : On applique différentes transformations (flip et/ou rotation, réduction de l'intensité) sur le carré Source courant (comme le montre la figure 07 Chapitre 1).
- ✓ [Étape 2] : on réduit sa taille pour le rendre de la même hauteur que la figure destination.
- ✓ [étape 3] : on calcule l'erreur entre les deux entités.
    - L'erreur est évaluée grâce à l'´écart type des pixels des 2 régions. Si cette erreur est minimale, on sauvegarde le couple Source/Destination ainsi que les modifications apportées.
    - Pour recréer l'image à partir du fichier final, il suffit de recréer les 2 partitions source et destination, et d'appliquer plusieurs fois les transformations des couples sauvés dans le fichier.
    - Le gène fractal de l'algorithme de Jacquin garantit alors la convergence de l'image vers l'image de d´épart. [12]

4.2. Partitionnements :

➢ Le partitionnement est l'opération qui consiste à segmenter une image en régions. Dans compression par la méthode Jacquin, nous avons besoin de deux partitionnements : Source et destination. Comme nous l'avons vu précédemment, la méthode Jacquin utilise des figures carrées. Voici les pavages réalisés lors de la compression et décompression Jacquin montré dans la Figure 31 : [12]

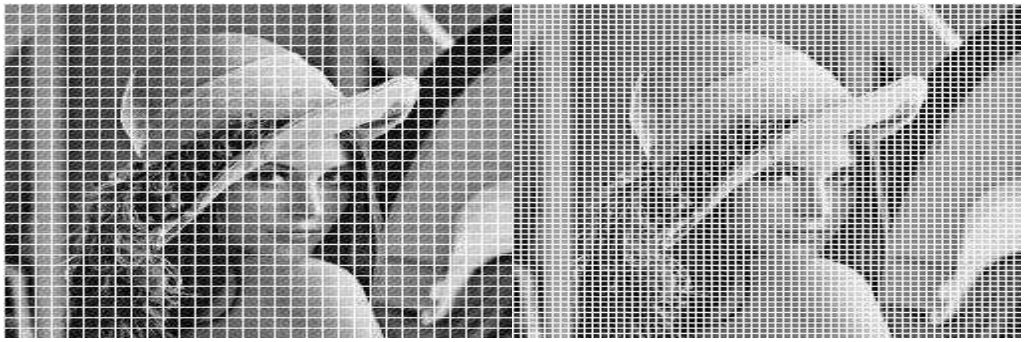


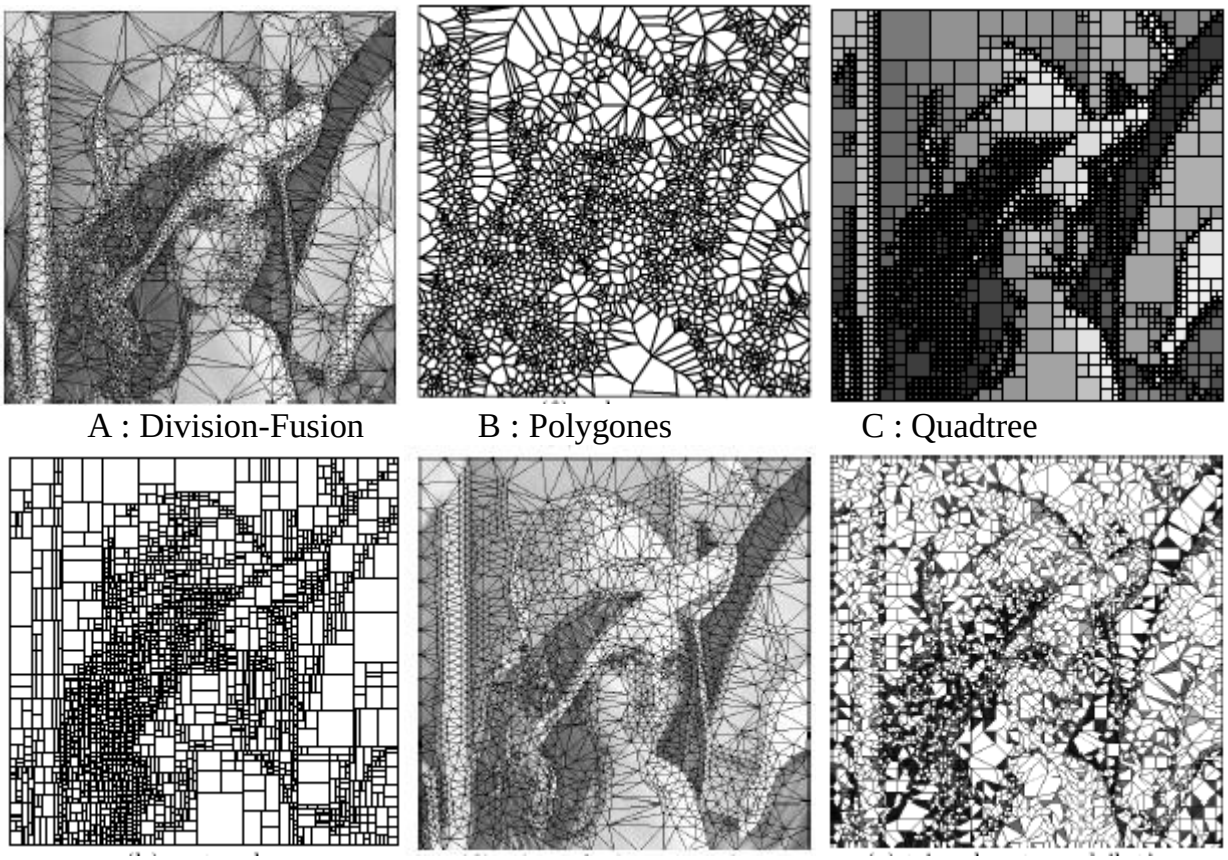
Figure 31 : Pavage Source et Destination, Méthode Jacquin [12]

➢ Un point essentiel dans les partitionnements Source et Destination est que le pavage destination doit être plus petit que le pavage source. En effet, dans le cas contraire, nous serions amenés à faire un agrandissement (et non une réduction) lors de la transposition des figures sources vers les figures destinations. [12]

4.2.1. Partitionnement adaptatifs :

➢ Différents partitionnements de l'image ont été étudiés de façon à minimiser le nombre de transformation locales, et à coder au mieux les similarités inter-blocs. Nous les énumérons ci-après selon une graduation du plus au plus souple :

- Partitionnement Carrés.
- Partitionnement Quadtree : La représentation d'une image à l'aide d'un quadtree est à résolution variable dans le sens où la partition exhibe des blocs de tailles différents. Le partitionnement est rigide car il est guidé par le processus de découpage récursif en blocs carrés. Il n'est pas adapté aux formes des objets de l'image, même s'il est adapté au contenue de celle –ci.
- Partitionnement Rectangulaire.
- Partitionnement Triangulaire.
- Partitionnement Triangulaire adaptif (Delaunay).
- Partitionnement à base de Quadrilatères. [19]



A : Division-Fusion B : Polygones C : Quadtree

D : Rectangle E : Triangulation Contrainte F : Triangulation quadrilatères

Figure 32 : Différentes Partitionnement allant d'une géométrie rigide (carrés) à une géométrie souple (triangle, Polygones) [19]

5. La Décomposition Quadtree :

- Le principal problème est que le codage fractal prend trop de temps. De nombreuses approches pour réduire le temps d'encodage ont une mauvaise affection sur l'image qualité après itération, donc la méthode de codage hybride de combiner le codage fractal et d'autres méthodes de codage deviennent une direction importante des méthodes fractales.
- L'approche Quadtree divise une image carrée en quatre blocs carrés de taille égale, puis teste chaque bloc pour voir si répond à un critère d'homogénéité. Si un bloc répond au critère, il n'est plus divisé, et le critère de test est appliqué à ces blocs. Ce processus est répété de manière itérative jusqu'à ce que chaque bloc réponde au critère.
- Le résultat peut avoir des blocs de plusieurs tailles différentes. [20]

6. Cryptage a clé privé :

6.1. L'AES (Advanced Encryption Standard) :

- Est un algorithme cryptographique par bloc à clé secrète.
- C'est le successeur du DES (Data Encryption Standard) qui a été implémenté dans un grand nombre de modules cryptographiques à une échelle mondiale depuis son apparition en 1977.
- L'AES conserve toujours le haut niveau de sécurité proposé par le DES
- L'innovation apportée par l'AES est notée au niveau de la clé secrète ainsi que la taille des données traitées en entrée.
- Nous passons d'une clé de chiffrement de 64 bits (8 octets) pour le DES vers une clé de taille double 128 bits (16 octets) pour l'AES.
- La taille des données à crypter est notablement plus grande puisque nous passons de 56 bits vers 128 bits. Ce qui permet d'exploiter l'algorithme dans des applications supportant des fichiers de données de taille grande.
- L'AES utilise un bloc d'une longueur fixée à 128 bits et une clé d'une longueur de 128, 192, et 256 bits.
- Comme tout système cryptographique symétrique, l'AES dispose de deux entrées, à savoir le texte clair à chiffrer (plaintext) ainsi que la clé de cryptage (key) (Figure 33). [21]

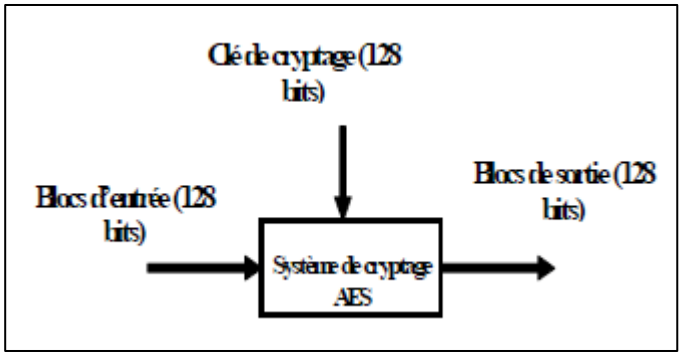


Figure 33 : Système de cryptage AES [21]

6.2. AES : Algorithme

- Bloc divisé en octets répartis dans des matrices 4x4 (1 octet = 8 bits) (Figure 34).
- **Séquence de 4 transformations répétées :**
  - ✓ 1ère étape : substitution (confusion)
  - ✓ 2ème étape : décalage des lignes (diffusion)
  - ✓ 3ème étape : brouillage des colonnes (diffusion)
  - ✓ 4ème étape : addition des sous-clés.
- **Les tours :**
  - ✓ Tour initial: addition XOR des sous-clés aux blocs
  - ✓ Tours similaires itérés: les 4 étapes sont répétées.
  - ✓ Dernier tour: transformation sans la 3ème étape.
- Étapes de chiffrement sont inversées et réordonnées pour produire un algorithme de déchiffrement. [22]

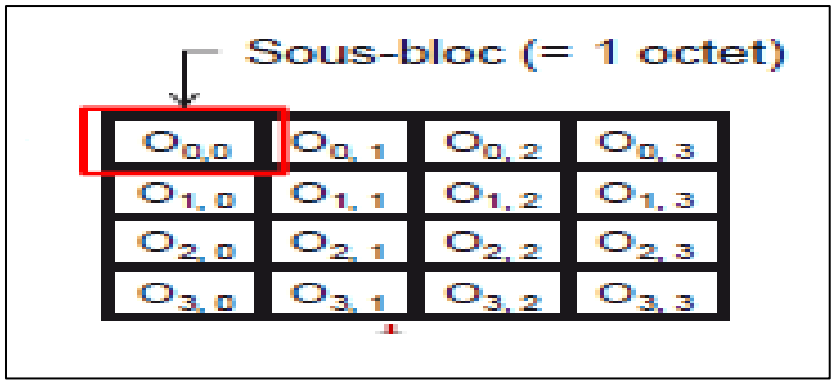


Figure 34 : Le bloc présenté par l'algorithme l'AES [22]

7. L'algorithme proposé pour la compression :

- Les étapes de l'algorithme sont les suivantes.
  a. Diviser l'image originale en utilisant la décomposition Quadtree du seuil est de 0.2, minimum La dimension minimale et la dimension maximale sont respectivement 2 et 64.
  b. Enregistrez les valeurs des coordonnées x et y, la valeur moyenne et la taille du bloc de Quadtree Décomposition.
  c. Enregistrez les informations du codage fractal pour terminer le codage de l'image à l'aide du codage Huffman et calculer le taux de compression.
  d. Pour l'image codée, on applique le décodage Huffman pour reconstruire l'image et calculer le PSNR. La Figure 35 montre la technique de compression fractale utilisée.

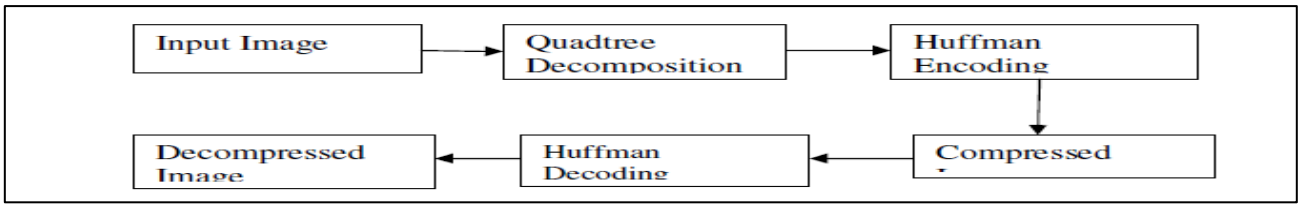


Figure 35: Technique de compression Fractale basée sur la décomposition en Quadtree [20]

7.1. Le codage de Huffman :

➢ L'algorithme de codage Huffman commence par construire une liste de tous les symboles de l'alphabet dans ordre décroissant de leurs probabilités.

➢ Il construit ensuite, de bas en haut, un arbre binaire avec un symbole à chaque feuille. Cela se fait par étapes, où à chaque étape deux symboles avec le plus petit les probabilités sont sélectionnées, ajoutées en haut de l'arborescence partielle, supprimées de la liste et remplacées avec un symbole auxiliaire représentant les deux symboles originaux. Lorsque la liste est réduite à un seul symbole auxiliaire (représentant tout l'alphabet), l'arbre est complet. L'arbre est alors traversé pour déterminer les mots de code des symboles. [20]

7.2. Décodage Huffman :

➢ Avant de commencer la compression d'un fichier de données, l'encodeur doit déterminer les codes. Il fait ça sur la base des probabilités de fréquences d'occurrence des symboles. Les probabilités ou les fréquences doivent être écrites, comme information latérale, sur la sortie, de sorte que tout décodeur Huffman pourra décompresser les données. C'est facile, car les fréquences sont des nombres entiers et les probabilités peuvent être écrites sous forme d'entiers mis à l'échelle.

➢ Il ajoute normalement quelques centaines d'octets à la production. Il est également possible d'écrire les codes de longueur variable eux-mêmes sur la sortie, mais cela peut être maladroit, car les codes ont des tailles différentes. Il est également possible d'écrire l'arbre Huffman sur la sortie, mais cela peut nécessiter plus d'espace que les seules fréquences. Dans tous les cas, le décodeur doit savoir ce qui se trouve au début du fichier compressé, le lire et construire le Huffman arbre pour l'alphabet. Ce n'est qu'alors qu'il peut lire et décoder le reste de son entrée.

➢ L'algorithme pour le décodage est simple. Commencez à la racine et lisez le premier bit de l'entrée (le fichier compressé). Si c'est zéro, suivez le bord inférieur de l'arbre; s'il en est un, suivez le bord supérieur. Lisez le bit suivant et déplacez un autre bord vers les feuilles de l'arbre. Lorsque le décodeur arrive sur une feuille, il y trouve le symbole original, non compressé, et ce code est émis par le décodeur. Le processus commence à nouveau à la racine avec le bit suivant. [20]

8. L'AES algorithme de cryptage d'image :

- **1ère phase ("Byte Sub") :**
  - ✓ Substitution de chaque sous-bloc avec des fonctions non linéaires ("S-boxes") : transformation non linéaire.
- **2ème étape ("Shift Row") :**
  - ✓ Décalage des lignes de chaque sous-bloc: les 3 dernières lignes sont décalées cycliquement vers la gauche avec des décalages différents : la deuxième ligne est décalée de trois octets, la troisième ligne de deux octets et la première ligne d'un octet.
- **3ème étape ("Mix Column") :**
  - ✓ Brouillage des colonnes : multiplication d'une matrice aux sous-blocs : Chaque colonne du bloc est transformée linéairement par la multiplication d'une matrice dont les coefficients sont constants.
  - ✓ Transformation linéaire.
- **4ème étape ("Add Round Key"):**
  - ✓ Addition des sous-clés aux sous-blocs avec XOR : Une clé, différente à chaque tour et de même longueur que le bloc, est ajoutée bit à bit au bloc par un XOR. Cette clé de tour est dérivée de la clé de chiffrement par un sous-algorithme, nommé algorithme de cadencement de clé (Figure 36). [21]

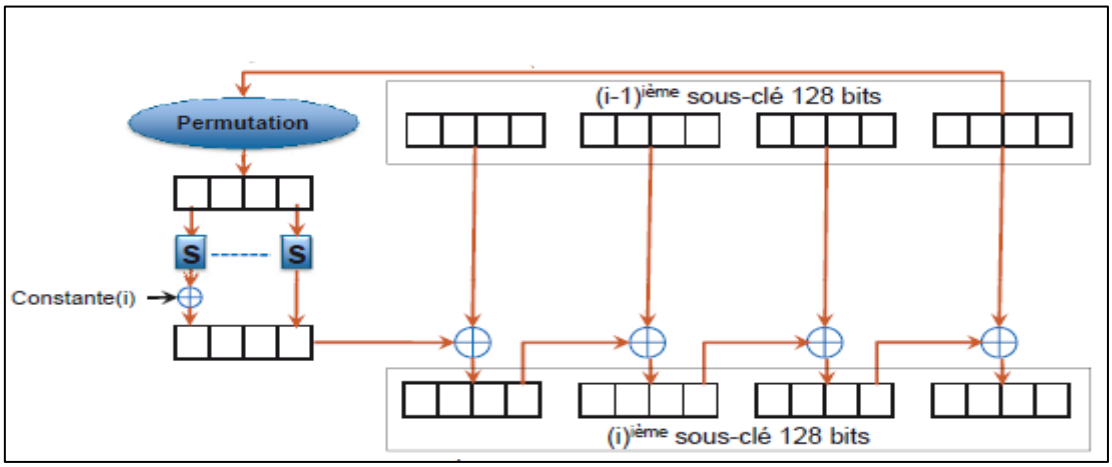


Figure 36 : Algorithme de génération de clés. [23]

8.1. L'algorithme de cryptage :

➢ L'implémentation de l'algorithme de chiffrement et de déchiffrement AES-128 à l'aide du logiciel MATLAB est effectuée. Dans laquelle l'entrée est une image et la clé au format hexadécimal et la sortie est la même que celle de l'image d'entrée. Pour le processus de cryptage, divisez d'abord l'image et rendez-la en 4 * 4 octets, c'est-à-dire au format matriciel. Calculez le nombre de tours en fonction de la taille de la clé et développez la clé à l'aide de notre clé. Et il y a (n-1) tours effectués qui sont des octets de substitution, décalent les lignes, mélangent les colonnes et ajoutent une clé. Le dernier tour «n» ne comprend pas de colonne de mélange dans l'itération. La Figure 38 montre le flux de l'algorithme. [24]

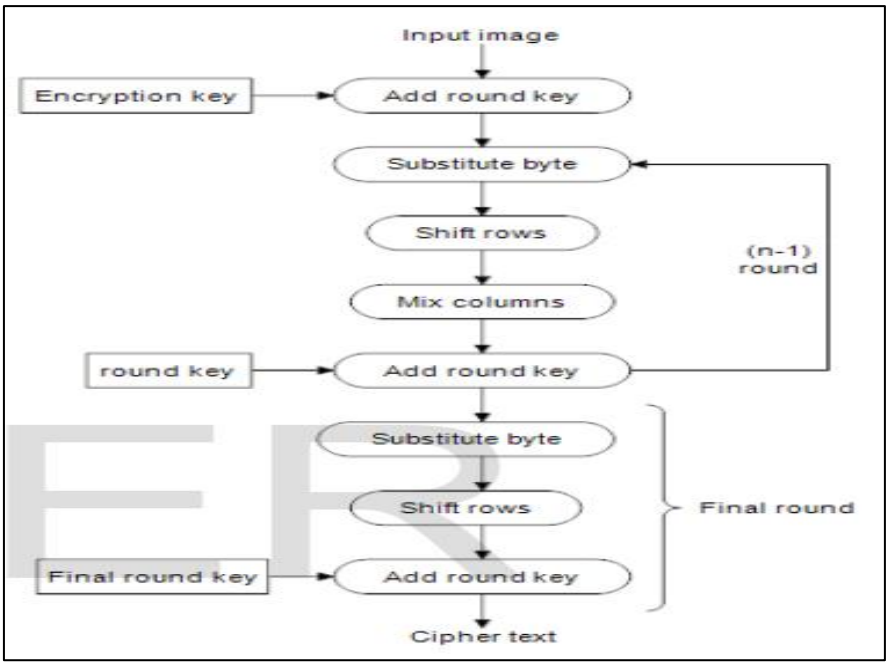


Figure 37 :L'algorithme proposé pour le Cryptage AES [24]

8.2. L'algorithme de décryptage :

➢ Le processus de décryptage AES est le processus inverse de celui du processus de cryptage. La figure 37 ci-dessus montre le flux de l'algorithme de décryptage AES. Composé de texte chiffré comme entrée, la clé est la même pour le processus de déchiffrement que pour le chiffrement. En cas de déchiffrement, l'octet de substitution inverse, les lignes de décalage inverses et les colonnes de mélange inverses doivent être implémentés. Alors que la touche d'ajout de rond reste la même. [24]

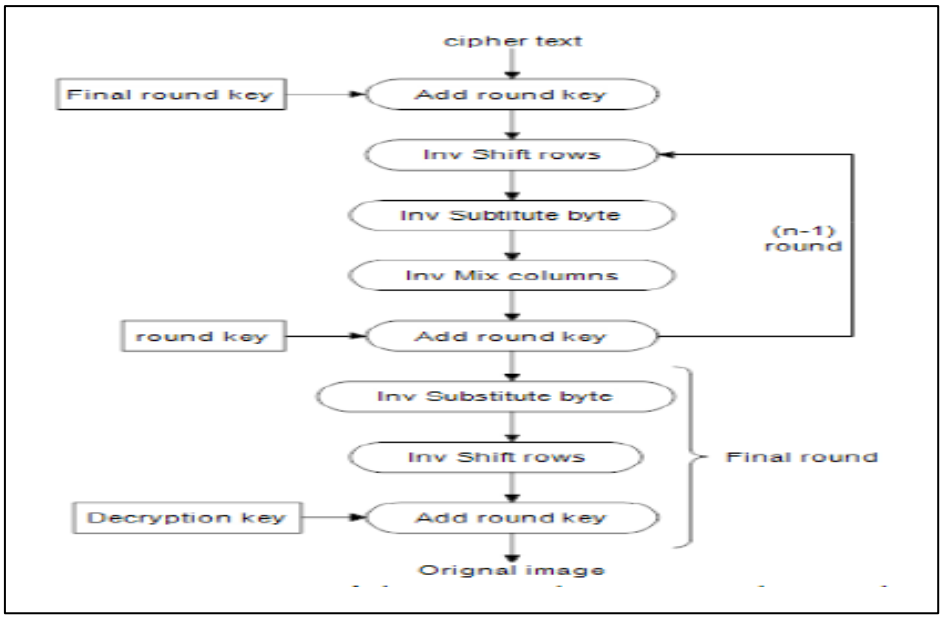


Figure 38 : L'algorithme proposé pour le Décryptage AES [24]

9. Système de Crypto-Compression proposé :

➢ L'approche introduite est basée sur deux algorithmes: un pour compresser et crypter l'image et l'autre pour reconstruire l'image. Ces deux algorithmes sont présentés comme montre la Figure 39.

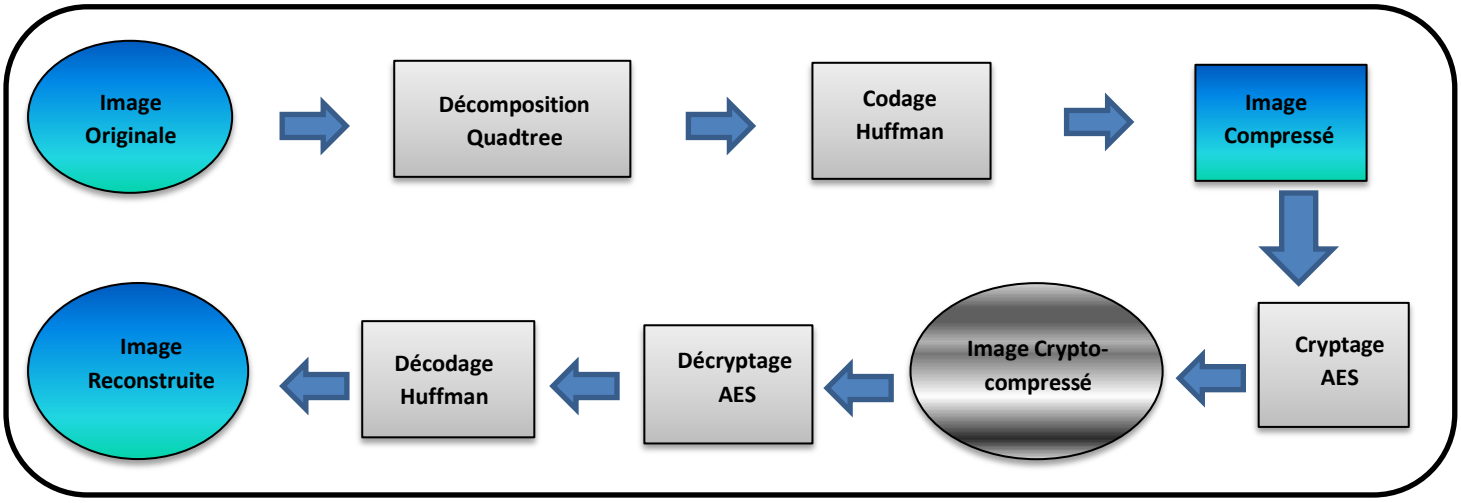


Figure 39: Système de Crypto-Compression proposé

10. Conclusion :

- Dans ce chapitre, nous avons commencé par une présentation détaillée sur la compression fractale ainsi qu'une étude de compression fractale par bloc, le théorème du point fixe, les IFS comme outils de compression fractale, comment ça marche le codage fractale et enfin, la décomposition Quadtree. Ensuite, nous avons présenté l'approche de cryptage AES.
- Enfin on a terminé avec la présentation du système de crypto-compression proposé. L'implémentation de l'algorithme du système proposé sera présentée dans le chapitre suivant avec comparaison des différents résultats.

TROISIEME CHAPITRE

RESULTAT ET DISCUSSION

1. Introduction :

➢ Notre but consiste à la réduction de la taille des bits présentés sur l'image (la taille de l'image), au même temps assurer leur transmission sécurisé sur le réseau, pour cela le travail de ce mémoire, il consiste à la réalisation d'un système hybride de crypto-compression, utilisant une méthode de compression fractal et un chiffrement par bloc, ces deux techniques sont appliqués l'un après l'autre dans un seul système, sur des images réelles.

➢ Plusieurs résultats sont obtenues grâce à l'exécution de notre système sur différentes images, ensuite on a changé la façon de l'application de ces deux approches sur les mêmes images pour procéder une étude comparative avec notre technique utilisé dans notre système et d'autre technique ; se basant sur des paramètres calculés : le MSE, PSNR, CR, Taux de compression, temps de compression et cryptage, temps de décompression et décryptage.

➢ Pour la réalisation de l'interface de travaille, 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.

➢ 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.

2. Environnement de travail :

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 10 dont les performances sont les suivantes :

- Processeur : Intel core (TM) i3-5010u CPU 2.10Ghz.
- Fréquence de 2.10 GHz.
- Mémoire installé (RAM) : RAM de 4 Go DDR3.
- Type de système : système d'exploitation 64 bits, processeur x64.
- Disque 500 Go HGST HTS545050A7E680.

2.2. Langage de programmation :

2.2.1. 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**[15] 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 coeur 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.

2.2.2. 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.

2.2.3. 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 40 illustre l'organigramme du logiciel élaboré.

➢ 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

[15] **HMI :** Les interfaces homme-machine ou IHM sont les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine.

décomposition Quadtree et le codage Huffman, et pour le chiffrement nous appliquerons l'algorithme d'AES.

- En deuxième partie nous faisant le déchiffrement et la décompression, pour bien valider la qualité de l'image reconstruite on calcul les paramètres de la distorsion à savoir le PSNR et MSE.

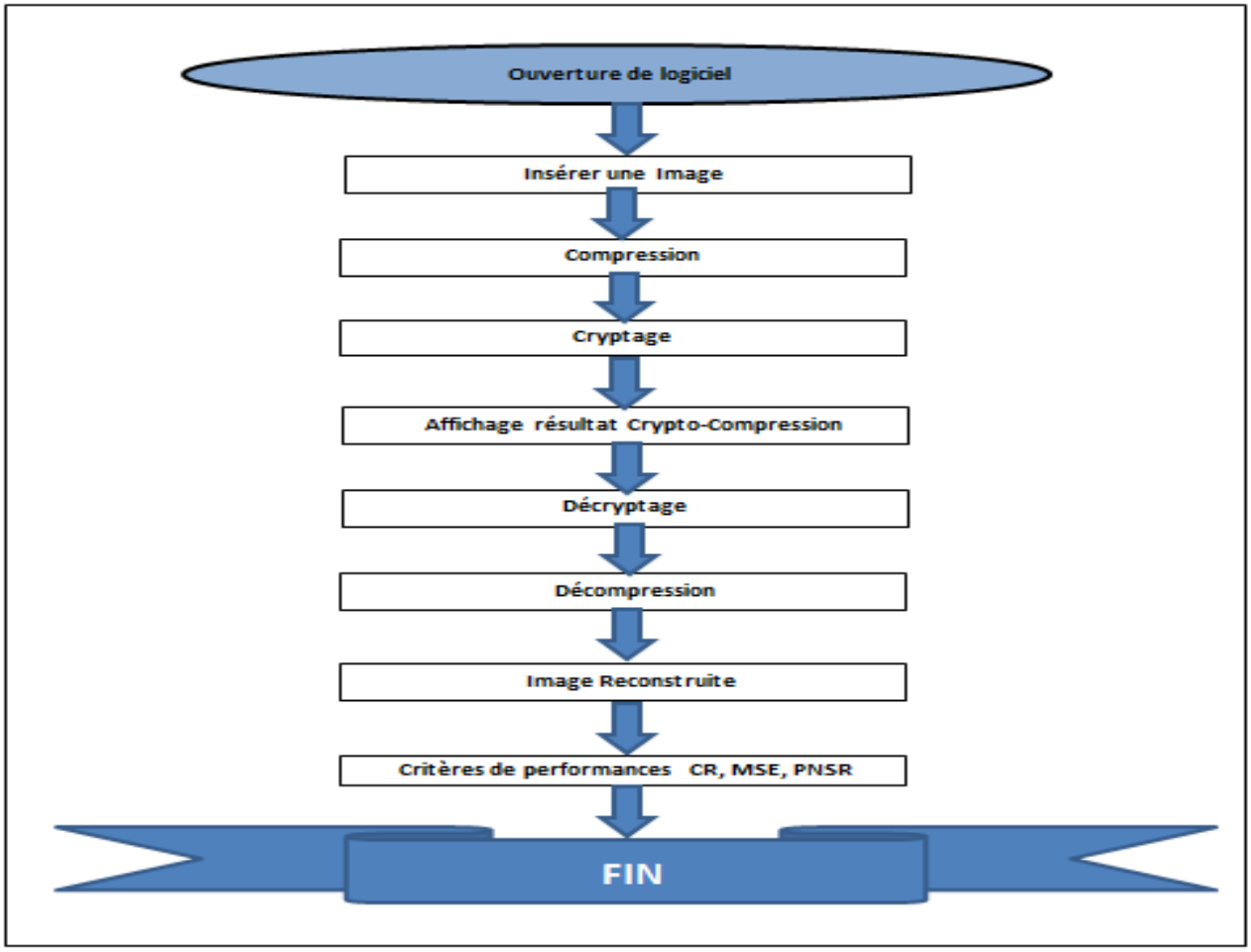


Figure 40 : Organigramme de logiciel élaboré

3. Principe de fonctionnement de l'application :

➢ La figure ci-dessous présente l'interface de l'application qui s'intitule « Image crypto-compression System ».

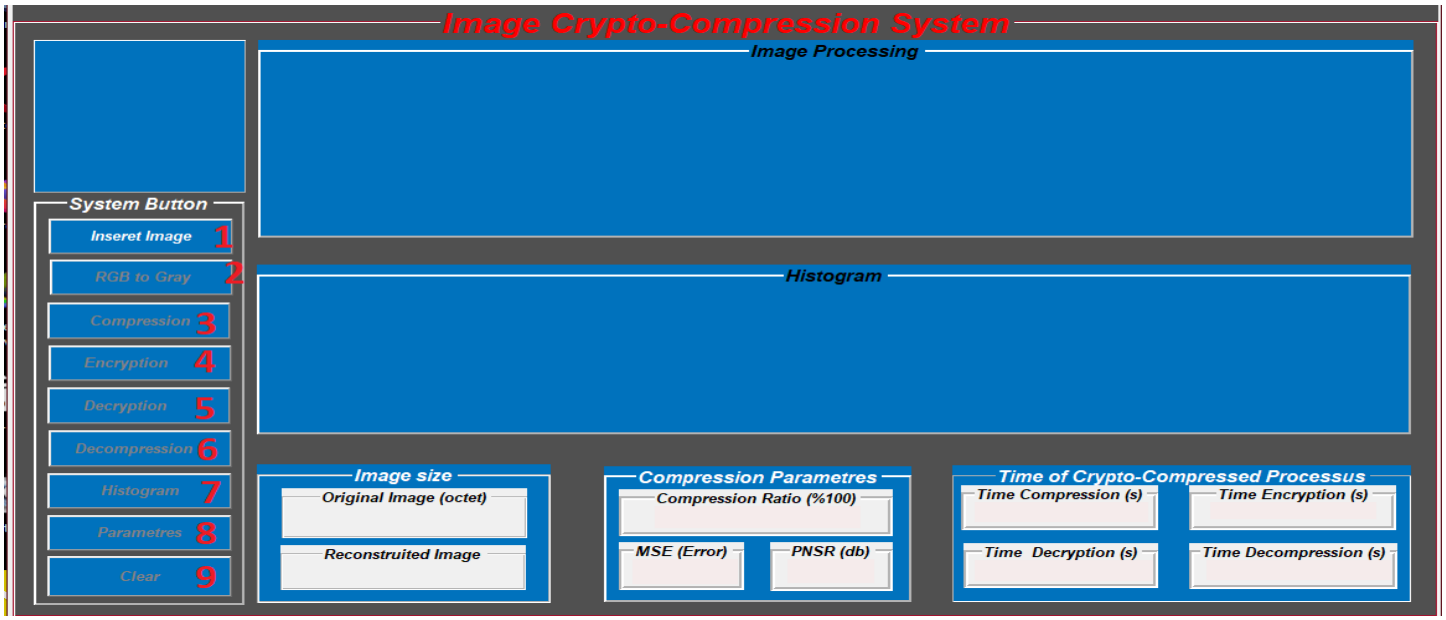


Figure 41 : Interface du système

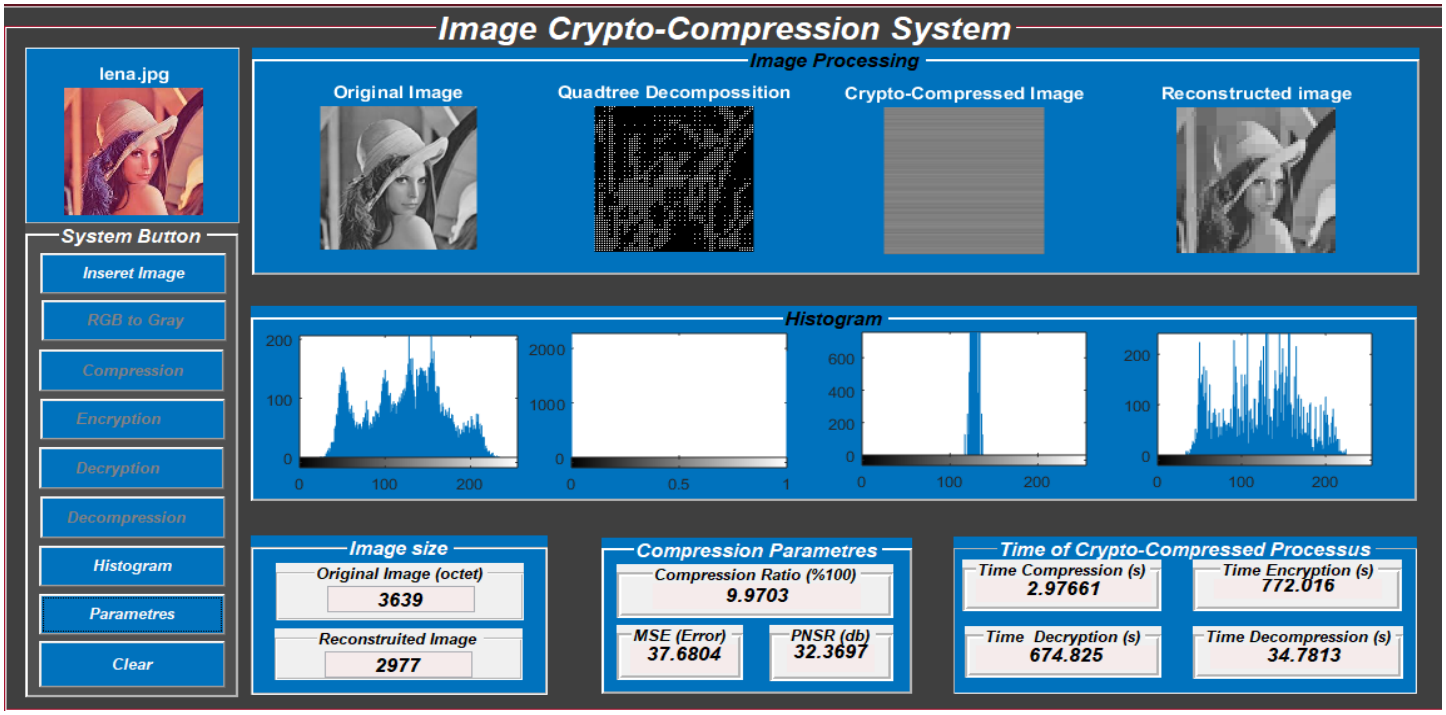


Figure 42 : Principe de fonctionnement de l'application

3.1. Description des modules de système :

1. Module ‘Lecture’ : Permet de charger une image à partir de n’importe qu’elle endroit du PC.
2. Module de ‘Transformation’ : Permet de changer la couleur de l’image de RGB au gris, d’autre façon l’image **RGB**[16] est une matrice compose de trois matrice uint8.
3. Module ‘compression’ : Ce module est le plus important dans notre système ; il contient l’algorithme hybride «décomposition Quadtree, et le codage de Huffman», il permet de faire la compression fractale de l’image originale, l’algorithme est réalisé on Matlab pour coder et décoder les images « La valeur de Threshold est 0.2, maximum dimension est 64 et minimum dimension est 2 pour la décomposition Quadtree.
4. Module ‘chiffrement’ : Ce module permet de crypter une image par le système de chiffrement AES, l’entré de ce module est le résultat obtenue par le module de compression ‘’matrice uint8 après la transformation d’un vecteur double qui contient des informations compressé’’. La clé utilisé est une hexadécimale clé de 16 bits c’est la même clé pour le module de décryptage.
5. Module ‘déchiffrement’ : Ce module permet de décrypter l’image crypté. L’entrée de ce module est la sortie du module précèdent est une image crypter « matrice uint8 », la sortie de ce module est une image « matrice uint8 » décrypter avec la même clé utilisé dans le module de décryptage, on le transforme cette dernier on vecteur double pour la préparer au module de décompression.
6. Module ‘décompression’ : Il permet de décompresser l’image et afficher l’image reconstruite. Ce module c’est la dernière étape dans notre système de Crypto-Compression, d’autre façon c’est la sortie de notre système qui donne l’image reconstruite.
7. Module de ‘’calcule l’histogramme’’ : Ce module permet de calculer l’histogramme de chaque étape du processus de crypto-compression.
8. 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.
9. Module de ‘’Clear’’ ou ‘’Reset’’ : Ce module permet de supprimer toutes les champs de l’interface utilisateur les variables utilisé pendant l’exécution du processus de notre système.

[16] Rouge, vert, bleu, abrégé en RVB ou en RGB est un système de codage informatique des couleurs, le plus proche du matériel

4. Bibliothèque d'images :

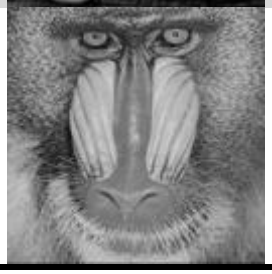
Bibliothèque des images			
Nom : **Lena** Taille Physique : **3.55 ko** Dimension : **128 X 128** Type de fichier : **JPG** Type de fichier après transformation : **JPG**		**Nom :** Avion **Taille Physique :** 3.55 ko **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	
Nom : **Barbara** Taille Physique : **3.63 ko** Dimension : **128 X 128** Type de fichier réel : **BMP** Type de fichier après transformation : **JPG**		**Nom :** Clown **Taille Physique :** 3.58 **ko** **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	
Nom : **Fruit** Taille Physique : **3.58** ko Dimension : **128 X 128** Type de fichier réel : **BMP** Type de fichier après transformation : **JPG**		**Nom :** Pimen **Taille Physique :** 3.76 **ko** **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	
Nom : **Mandr** Taille Physique : **4.11 ko** Dimension : **128 X 128** Type de fichier réel : **BMP** Type de fichier après transformation : **JPG**		**Nom :** House **Taille Physique :** 3.78 **ko** **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	

Tableau 1: Bibliothèques des images utilisées

5. Tests expérimentaux :

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

5.1. 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, ensuit nous appliquons le cryptage et décryptage de ces images et enfin la décompression (comme le montre la Figure 43), en discutant sur les paramètres de performances de cette dernières, parmi ses paramètres : MSE, PSNR, le taux de compression, le temps de cryptage et décryptage et le temps de compression décompression. Le Tableau 02 présente les résultats obtenus.

➢ La clé utilisé dans les tests qui suit est en hexadécimale sur 16 bits :

```
Key='000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f';
```

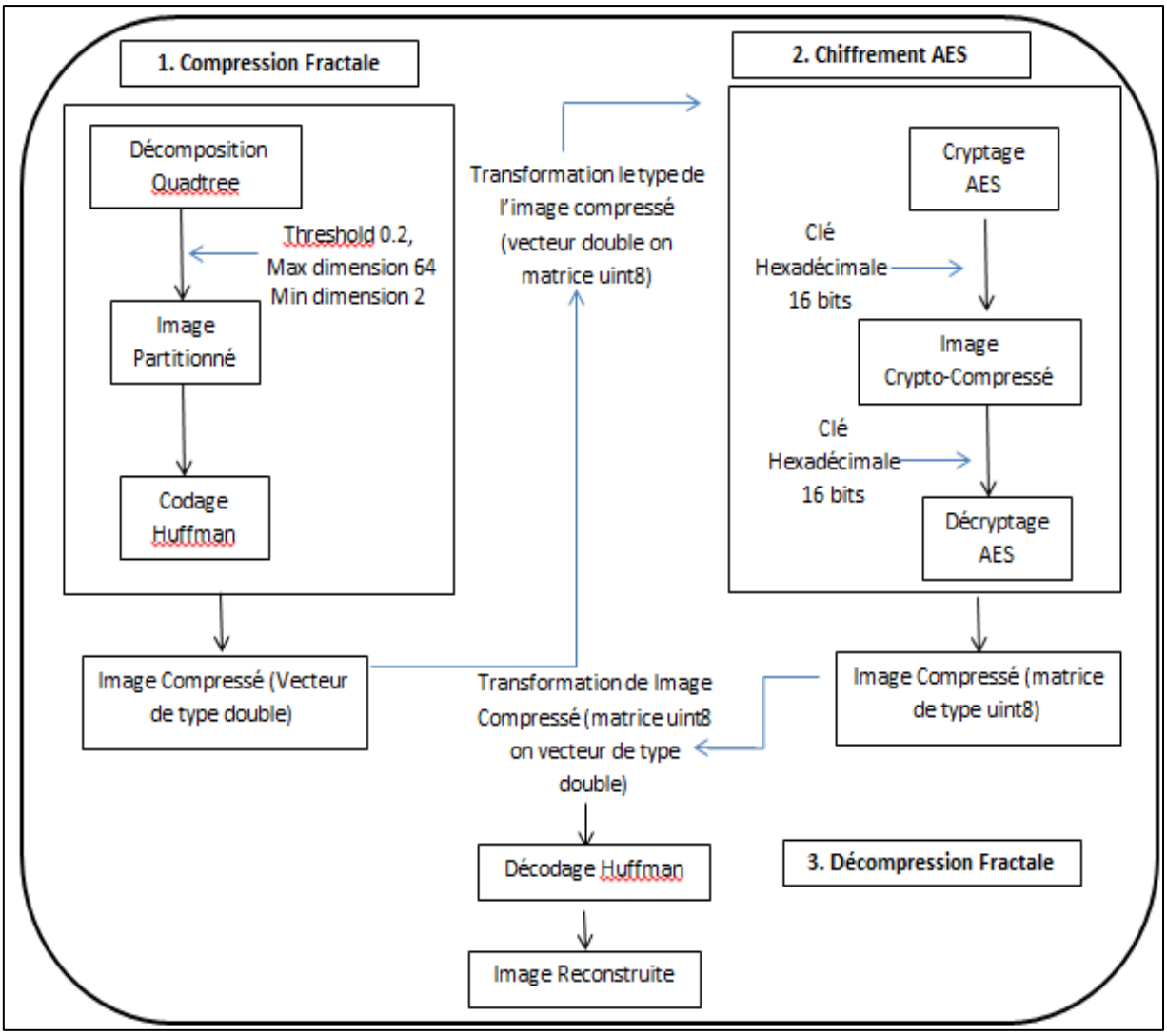


Figure 43 : Schéma synoptique de notre système

5.1.1. Testes et résultats :

➢ Les images de la collection étudiée, sont compressées avec la compression Fractale suivant l'algorithme hybride Décomposition Quadtree -Codage Huffman, et chiffré suivant l'algorithme AES -256.

5.1.1.1. Premier Cas :

➢ Comme nous avons montré dans la figure 40 et 43 notre système applique la compression comme une première étape ensuite le cryptage et le décryptage et enfin la décompression, alors c'est notre premier test, et voilà le Tableau 02 qui présente les résultats obtenus.

L'image			Test	Paramètres						
Nom Taille O Taille R	**Dimension**	**Type**		**CR%**	**MSE (Err)**	**PNSR (db)**	**Temps compression (S)**	**Temps Cryptage (S)**	**Temps Décryptage (S)**	**Temps Décompression (S)**
Lena **3.55 ko** **2.97 ko**	128 X 128	JPG	01	9.97	37.68	32.369	2.97	772.015	674.825	34.78
Barbara **3.71 ko** **3.01 ko**	128 X 128	BMP	02	9.55	43.18	31.77	2.84	808.556	698.263	32.75
Clown **3.66 Ko** **2.99 Ko**	128 X 128	BMP	03	9.86	40.97	32.005	3.61	790.324	660	30.86
Avion **3.63 ko** **2.68 ko**	128 X 128	BMP	04	11.25	36.45	32.513	2.78	770.299	704.179	43.065
Mandr **4.21 ko** **2.92 ko**	128 X 128	BMP	05	8.67	51.11	31.043	3.17	921.731	758.229	32.692
Fruit **4.27 ko** **3.30 ko**	128 X 128	BMP	06	8.08	48.02	31.310	3.71	969.586	790.309	42.01
House **3.87 ko** **2.94 ko**	128 X 128	BMP	07	9.10	44.67	31.630	3.33	861.691	709.709	36.49
Boat **3.30 ko** **2.36 ko**	128 X 128	BMP	08	12.09	35.14	32.599	4.31	690.886	525.823	24.484
Isabe **3.60 ko** **2.42 ko**	128 X 128	BMP	09	12.90	42.18	31.87	3.31	761.956	608.786	23.278
Pimen **3.86 ko** **3.23 ko**	128 X 128	BMP	10	9.62	40.73	32.03	3.15	1019.04	745.595	33.81

Tableau 2 : Résultat d'application de notre système sur différentes images

5.1.1.2. Processus de traitement et Histogramme :

- Test 01 : Lena.jpg :

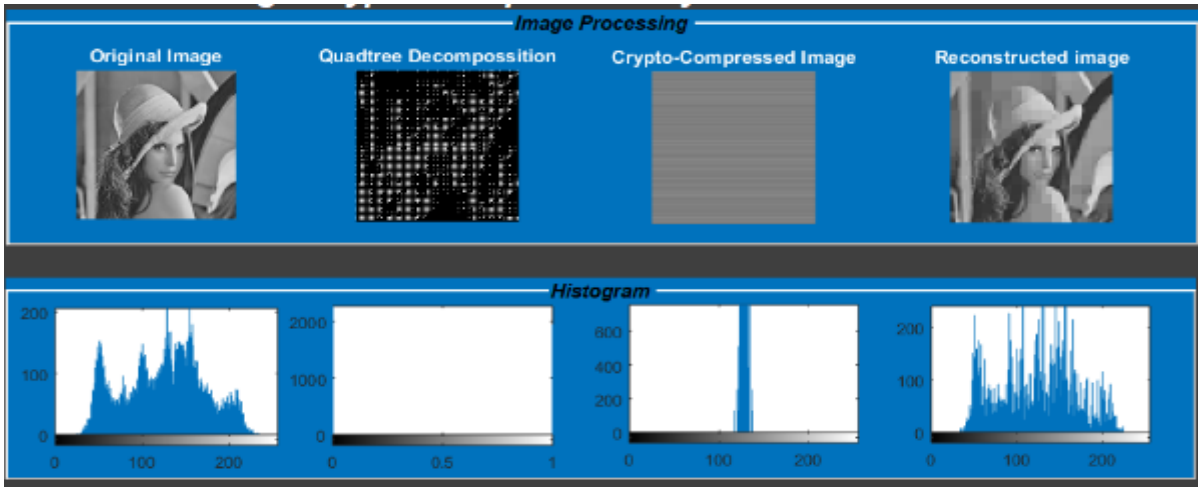


Figure 44 : Processus de traitement et l'histogramme Lena.jpg

- Test 02 : Brabara.bmp :

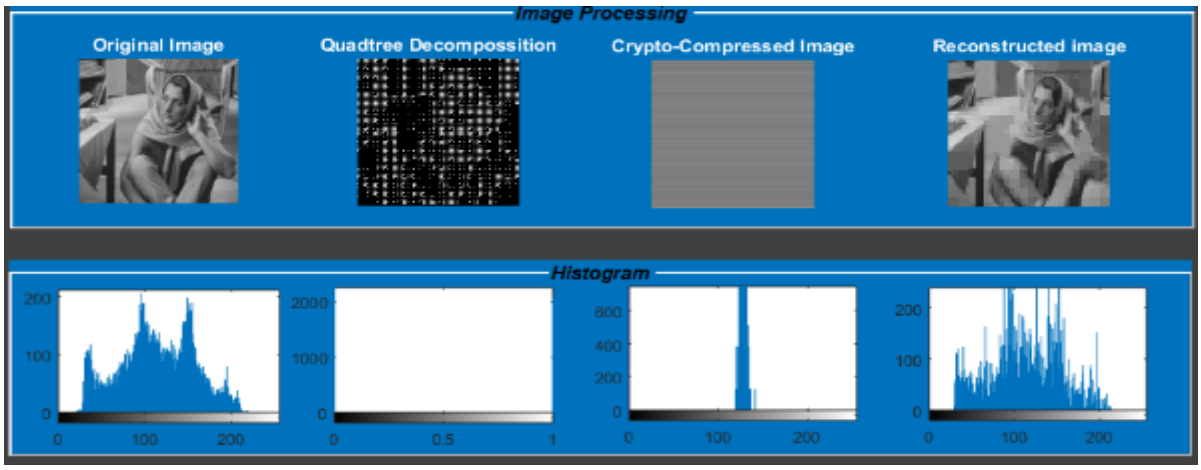


Figure 45: Processus de traitement et l'histogramme Brabara.bmp

- Test 03 : Clown.bmp :

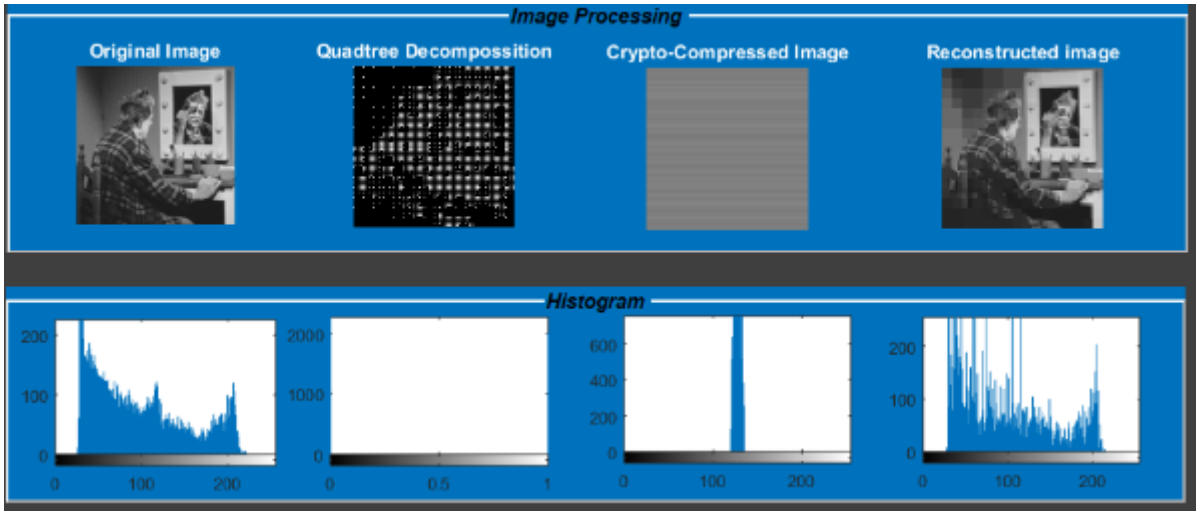


Figure 46: Processus de traitement et l'histogramme Clown.bmp

- Test 04 : Avion.bmp :

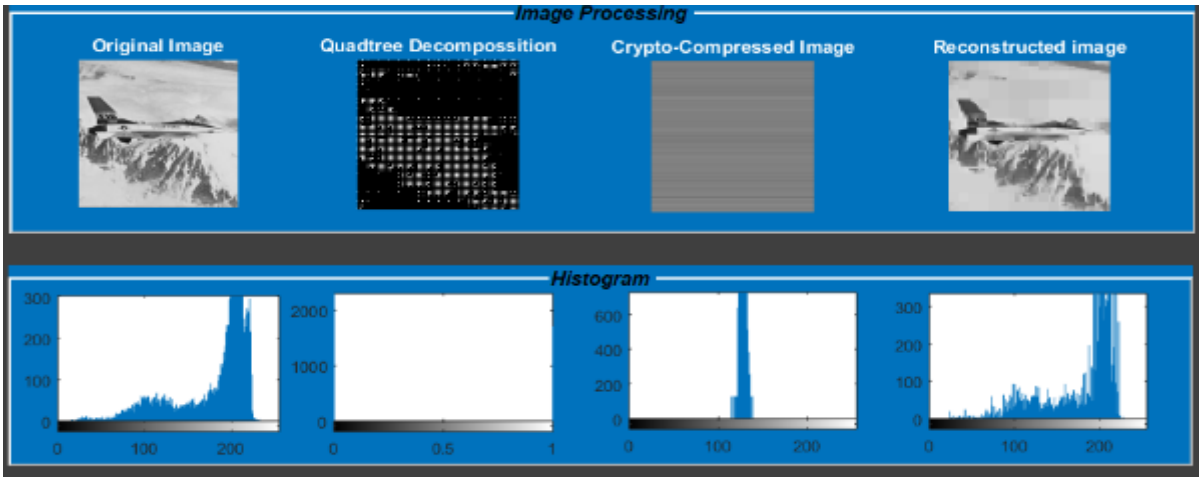


Figure 47: Processus de traitement et l'histogramme Avion.bmp

- Test 05: Mandr.bmp:

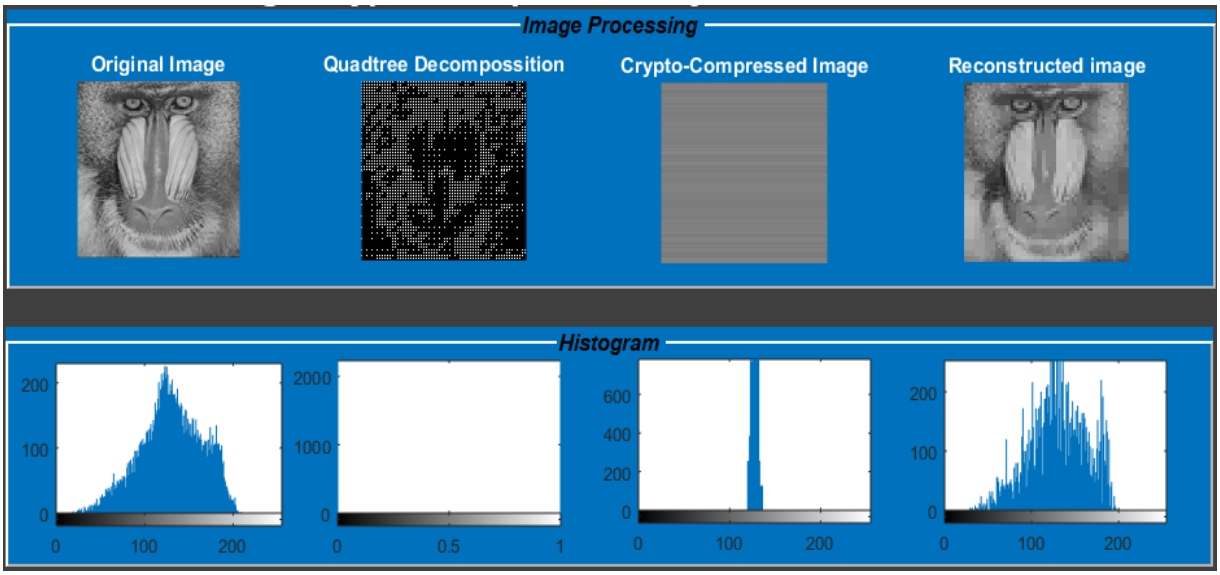


Figure 48 : Processus de traitement et l'histogramme Mandr.bmp

- Test 06: Fruit.bmp:

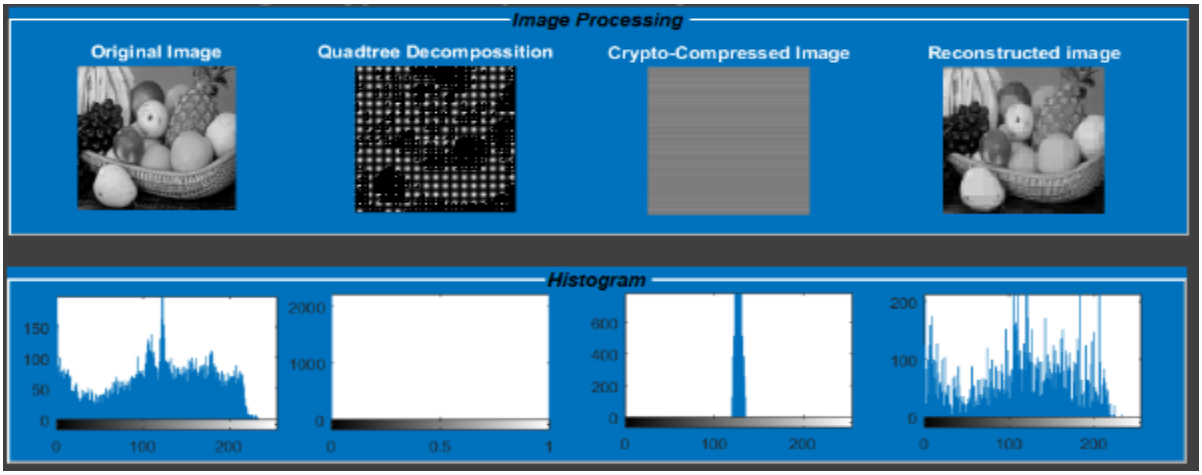


Figure 49: Processus de traitement et l'histogramme Fruit.bmp

- Test 07: House.bmp:

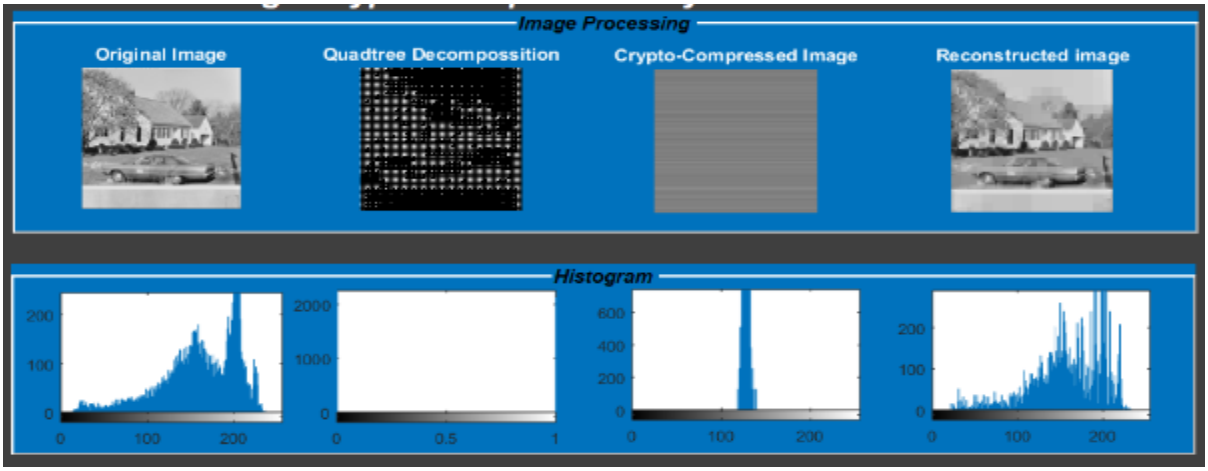


Figure 50: Processus de traitement et l'histogramme House.bmp

- Test 08: Boat.bmp:

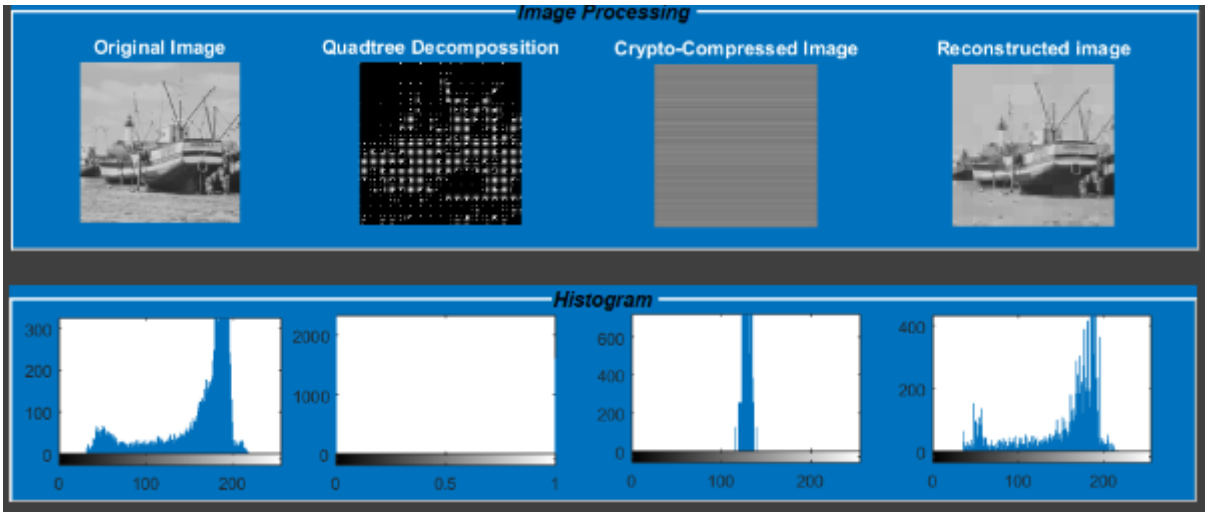


Figure 51: Processus de traitement et l'histogramme Boat.bmp

- Test 09: Isabe.bmp:

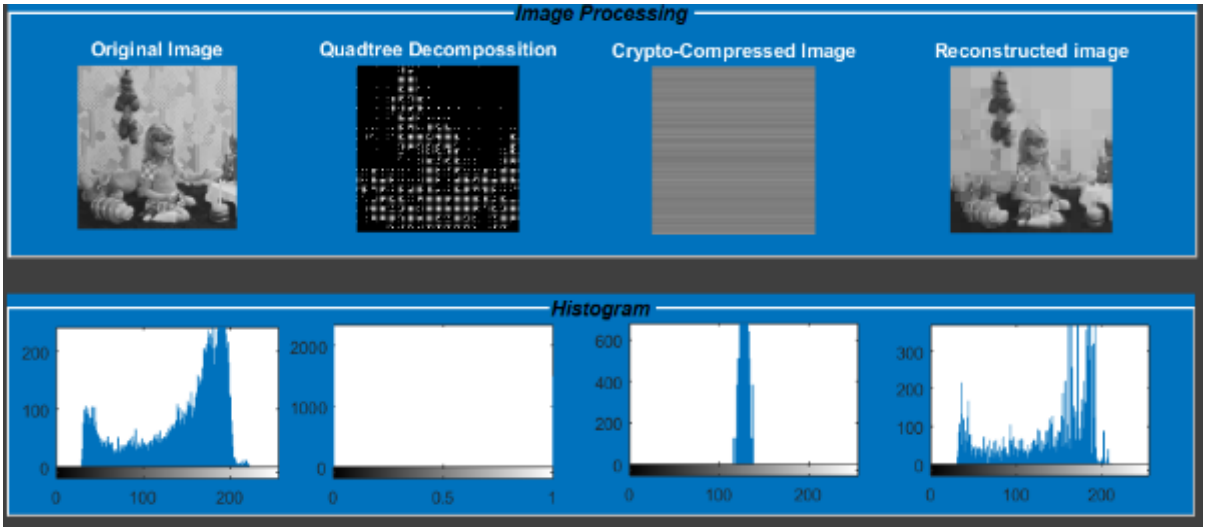


Figure 52: Processus de traitement et l'histogramme Isabe.bmp

- Test 10: Pimen.bmp:

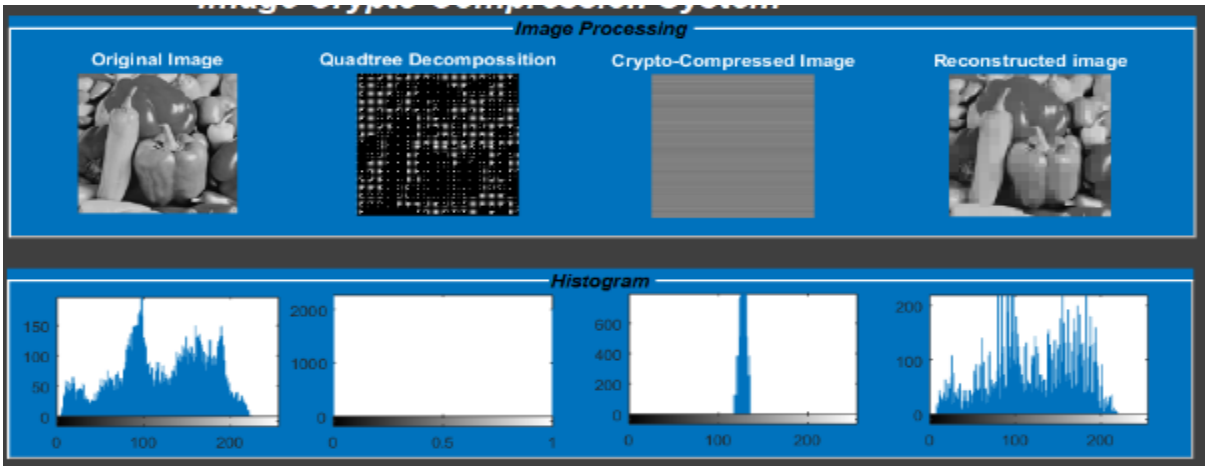


Figure 53: Processus de traitement et l'histogramme Pimen.bmp

5.1.1.3. Deuxième Cas :

- Nous essayons maintenant de changer l'ordre d'appliquer les étapes de notre système c'est-à-dire nous allons appliquer le cryptage comme une première étape ensuite la compression et décompression et enfin le décryptage sur les mêmes images utilisées dans le premier cas, et on va calculer les paramètres de performances tel que le CR, MSE, PNSR, Temps de Cryptage, Temps de Compression, Temps de Décompression et Temps de Décryptage.

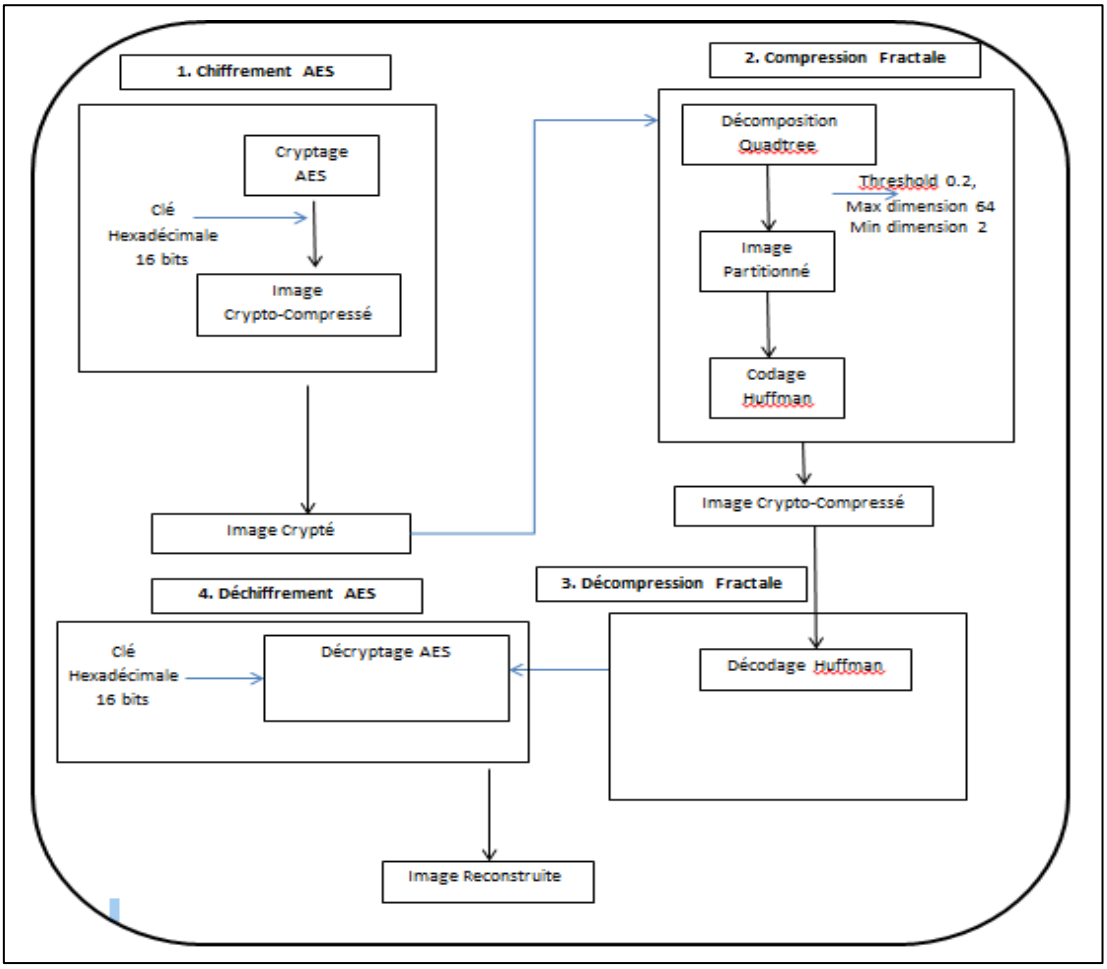


Figure 54 : Schéma synoptique du deuxième cas

➢ L'exécution d'enchainement des étapes dans ce cas se fait parfaitement sans aucune erreur d'exécution mais les résultats obtenues ne le sont pas, comme le montrent les Figures suivantes :

Workspace

Name	Value	Size	Class	Min	Max
blkcount	4096	1x1	double	4096	4096
CompRatio	5.0822	1x1	double	5.0822	5.0822
Data	*16387x1 single*	16387x1	single	0	246
Dict	*217x2 cell*	217x2	cell		
Image_crypt_comp	*99689x1 double*	99689x1	double	0	1
Image_crypte	*128x128 uint8*	128x128	uint8	0	255
Image_Recons	*128x128 uint8*	128x128	uint8	0	255
Img	*128x128 uint8*	128x128	uint8	21	239
Img_crypte_dcmp	*128x128 uint8*	128x128	uint8	22	246
key	'000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f'	1x64	char		
lena	*220x220x3 uint8*	220x220x3	uint8	0	255
Quadtre	*128x128 double*	128x128	double	0	2
Tcmp	6.1721	1x1	double	6.1721	6.1721
Tcrp	128.7782	1x1	double	128.7782	128.7782
TDcomp	82.6942	1x1	double	82.6942	82.6942
TDcrypt	93.3261	1x1	double	93.3261	93.3261

Figure 55: Résultat du deuxième Cas Lena.jpg

Workspace

Name	Value	Size	Class	Min	Max
barba	*512x512x3 uint8*	512x512x3	uint8	<Too man...	<Too man...
blkcount	4096	1x1	double	4096	4096
CompRatio	5.0796	1x1	double	5.0796	5.0796
Data	*16387x1 single*	16387x1	single	0	237
Dict	*219x2 cell*	219x2	cell		
Image_crypt_comp	*99711x1 double*	99711x1	double	0	1
Image_crypte	*128x128 uint8*	128x128	uint8	0	255
Image_Recons	*128x128 uint8*	128x128	uint8	0	255
Img	*128x128 uint8*	128x128	uint8	21	232
Img_crypte_dcmp	*128x128 uint8*	128x128	uint8	20	237
key	'000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f'	1x64	char		
Quadtre	*128x128 double*	128x128	double	0	2
Tcmp	5.9425	1x1	double	5.9425	5.9425
Tcrp	117.4616	1x1	double	117.4616	117.4616
TDcomp	83.1003	1x1	double	83.1003	83.1003
TDcrypt	186.2532	1x1	double	186.2532	186.2532

Figure 56:Résultat du deuxième Cas Barbara.jpg

- Les Figures 55 et 56 montrent qu'à cause du type de l'algorithme de compression Fractale (avec perte d'information), l'algorithme de Cryptage AES qui se base sur le block Cipher, trouve que les informations Cryptées dans la première étape ne sont pas les mêmes, alors Il n'a pas pu décrypter ces informations.
- Dans chaque résultat obtenu nous remarquons que l'image originale « Img» est de type ''uint8'' avec min valeur 21 et max 232, et l'image reconstruite «Image_Recons » est aussi de type ''uint8 '', mais les valeurs min et max ne sont pas les mêmes ce qui implique que l'algorithme de décryptage n'a pas pu de décrypter les informations cryptées auparavant.

6. Interprétation des résultats :

- Un algorithme performant de compression possède un gain de compression maximale et une erreur quadratique moyenne minimale, pour cela nous calculons le Taux de compression et le gain de compression en utilisant les résultats précédents obtenus dans le Tableau 02 (le Tableau suivant présente ces deux paramètres).

- Taux de compression : $\boldsymbol{T = \frac{1}{CR}}$
- Gain de compression : $\boldsymbol{G = 1 - T}$

L'image	Test	CR %	T	G
Lena	01	9.97	**0,1003**	**0,8997**
Barbara	02	9.55	**0,1047**	**0,8953**
Clown	03	9.86	**0,1014**	**0,8986**
Avion	04	11.25	**0,0888**	**0,9112**
Mandr	05	8.67	**0,1114**	**0,8886**
Fruit	06	8.08	**0,1237**	**0,8763**
House	07	9.10	**0,1098**	**0,8902**
Boat	08	12.09	**0,0827**	**0,9173**
Isabe	09	12.90	**0,0775**	**0,9225**
Pimen	10	9.62	**0,1039**	**0,8961**

Tableau 3: Taux et Gain de Compression

- Dans le Tableau précèdent on remarque que les valeurs des Gain de compression sont dans l'intervalle [0.89, 0.92], c'est-à-dire sont proches du « un », ce qui implique une compression performante.
- Comme on a vu dans le premier chapitre, pour mesurer la distorsion entre l'image reconstruite et l'image originale (Mesure de la qualité visuelle de l'image reconstruite) on va utiliser l'Erreur Quadratique Moyenne MSE (Mean Square Error) ou du rapport signal à bruit PSNR (Peak Signal to Noise Ratio). Dans le Tableau 02, les valeurs des MSE et PNSR dans l'intervalle [35,51], [30, 32], sont un peu élevés.
- L'histogramme d'une image mesure la distribution des niveaux de gris dans l'image. Pour un niveau de gris x, l'histogramme permet de connaitre la probabilité de tomber sur un pixel de valeur x en tirant un pixel au hasard dans l'image.
    - Concrètement, l'histogramme d'une image à valeurs entières est construit de la manière suivante: pour chaque niveau de gris x, on compte le nombre de pixels ayant la valeur x.
    - L'histogramme permet d'obtenir rapidement une information générale sur l'apparence de l'image. Une image visuellement plaisante aura généralement un histogramme équilibré (proche d'une fonction plate).
    - On prend notre résultat obtenues sur les différentes images par notre système nous avons remarqué que l'histogramme est tassé sur le centre dans la plupart des tests (01, 02, 05, 06, 10), mais dans les tests (04, 07, 08, 09) l'histogramme est tassé sur la droite, et le test 03 l'histogramme est tassé sur la gauche.
    - On remarque que la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse, dans l'image originale n'est pas la

même dans l'image reconstruite ce qui implique la perde d'information grâce à l'algorithme de compression.

6.1. Discussion :

6.1.1. Premier cas :

- Dans notre système de crypto-compression on a appliqué le chiffrement AES après la compression Fractale, D'après les résultats obtenus 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 a vu, on perd de la qualité de l'image et la reconstitution n'est pas fidèle.

6.1.2. Deuxième Cas :

- Nous avons essayé d'appliquer le système d'une autre façon où on a appliqué le chiffrement avant la compression : D'après les résultats obtenus on a remarqué, qu'on a on a des problèmes dans la reconstitution de l'image puisque les données ou les informations cryptées au début du processus (étape de chiffrement avant la compression) au niveau d'image, ne sont pas les mêmes (Dernière étape du processus décryptage après compression et décompression). Cela est causé par l'algorithme de compression fractale (avec perte d'information).

7. Conclusion :

- Dans la première partie de ce chapitre, nous avons présenté l'environnement de travail et le langage de programmation que nous avons utilisé, ainsi que notre système de crypto-compression et les modules de fonctionnement de ce système.
- Dans la deuxième partie de ce chapitre nous avons testé plusieurs types d'image à l'entrée de notre système. Ensuite nous avons appliqué le chiffrement sur la compression et nous avons enregistré les résultats.
- Dans la dernière partie nous avons discuté sur les résultats obtenus, et d'après les résultats présentés, on remarque bien que le chiffrement AES agit sur la robustesse des techniques compressions ainsi sur la reconstruction de l'image comme.
- On a vu aussi, que si on applique le chiffrement avant compression. l'image obtenue est complètement bruitée, cela est dû à la perte d'information causée par l'algorithme de compression utilisé qui est avec perte.

CONCLUSION GENERALE

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 compression fractale en utilisant le partitionnement Quadtree et le codage de Huffman ; et pour le chiffrement on a choisi le système de chiffrement par bloc, et on a utilisé l'algorithme de cryptage AES-256.

- 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 la compression sur le chiffrement et l'inverse.

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

Liste des abréviations :

A:

- AES : Advanced Encryption Standard

B:

- BMP : BitMaP

C:

- CR : En anglais « Compression Ratio » : Rapport de compression.
- Cipher Block Chaining (CBC).
- Cipher Feedback(CFB).

D:

- DES : Data Encryption Standard
- DB : Le décibel ou (dB)

E:

- Electronic Code Book(ECB)

G:

- GSM : Global System for Mobile Communications
- GUI (pour Graphical User Interface

I:

- IFS (Iterated Function System)
- ISO : International Organisation for Standardisation).
- IRM : L'imagerie par résonance magnétique (IRM)

J:

- JPG : Joint Photographic Group

L :

- LZW : LZW (pour Lempel-Ziv-Welch)

M :

- MSE : En anglais « Mean Square Error » L'Erreur Quadratique Moyenne.

O :

- Output Feedback (OFB).

P :

- PSNR: En anglais « Peak Signal to Noise Ratio » Rapport signal à bruit.

R :

- RLC : Le run-length encoding, appelé en français le codage par plages.
- RSA : **R**onald Rivest, Adi **S**hamir et Leonard **A**dleman.
- RGB : Rouge, vert, bleu, abrégé en RVB ou en RGB
- RC4 (Rivest Cipher 4)

Résumé:

- La compression et le cryptage des données représentent deux technologies dont l'importance est en croissance exponentielle et ce dans une multitude d'applications. De plus, l'utilisation excessive de réseaux informatiques pour le transfert de données doit évidemment obéir à un double objectif : la réduction du volume de données afin d'encombrer le moins possible les réseaux de communication publics et la confidentialité afin d'assurer un niveau optimal de sécurité.
- D'une part, la compression d'images a pour but de réduire la taille d'une image afin de faciliter son stockage aussi bien que son transfert. Ainsi on distingue deux grandes familles de méthodes compression, à savoir celles qui provoquent des pertes d'information causant une image reconstruite non fidèle à l'originale mais de taille très réduite. Les autres méthodes ne provoquent pas de perte d'information mais présentent des taux de compression réduits.
- D'autre part, le cryptage des données est généralement décrit à partir d'une communication secrète d'informations entre deux interlocuteurs. Dans un système informatique, cette confidentialité intervient dans plusieurs formes, en particulier dans la protection du stockage, de l'accès et transmission de l'information.

Abstract :

- Data compression and encryption are two technologies that are growing exponentially in a multitude of applications. In addition, the excessive use of computer networks for data transfer must obviously obey a double objective: the reduction of the volume of data in order to clutter the public communication networks as little as possible and confidentiality in order to ensure a level optimal security.
- On the one hand, image compression aims to reduce the size of an image in order to facilitate its storage as well as its transfer. Thus there are two main families of compression methods, namely those which cause information loss causing a reconstructed image not faithful to the original but of very small size. The other methods do not cause loss of information but have reduced compression rates.
- On the other hand, data encryption is generally described on the basis of a secret communication of information between two interlocutors. In a computer system, this confidentiality intervenes in several forms, in particular in the protection of storage, access and transmission of information.

BIBLIOGRAPHIE

Bibliographie :

[1] M. ALDOSSARI, «Nouvelle méthode optique de compression et de cryptage simultanés des images (fixes/vidéo) pour les systèmes télécommunication,» *" HAL" Thèse de Doctorat UNIVERSITÉ DE BRETAGNE OCCIDENTALE* , p. 28, 02 Février 2006.

[2] S. Pigeon, *Contribution à la compression de données,Thèse présentée à la Faculté des arts et sciences en vue de l'obtenation du grade Philosophie Doctor en Informatique*, Montréal,Canada: Département d'informatique et de recherche opérationnelle, 2001.

[3] A. Mourad, «Crypto compression d'image par cryptage partiel En vue de l'obtention d'un Master II en Réseau et télécommunication,» Université Mouloud Mammeri de Tizi-Ouzou, 2015.

[4] Z. Athmane, «Ondelettes et techniques de compression d'images numérique,» *THESE pour l'obtention du Diplôme de Doctorat en Sciences en Electronique*, pp. 5-16, 2012/2013.

[5] S. Renard, «La compression des données,» chez *Club Photoshop de Nantes*, 14 Octobre 1999.

[6] L. Diane, Rapport de recherche "Cours de traitements d'images" , Centre National de la Recherche Scientifique,, ISRN I3S/RR–2004-05–FR, 22 Janvier 2004.

[7] C. TAOUCHE, «Implémentation d'un Environnement Parallèle pour la Compression d'Images à l'aide des Fractales , Memoire Pour l'obtention du diplôme de Magister en Informatique Option Information & Computation,» Université Mentouri Faculté des Sciences de l'Ingénieur Département d'Informatique, Constantine, 2005.

[8] P. Plumé, «Techniques de compression de données,» *Edition EYROLLES et la revue « PC EXPERT »*, pp. 1-6, Janvier 1995..

[9] C. Wagner, «De l'image vers la compression. [Rapport de recherche] RR-2035, INRIA.1993,» INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (INRIA), Rennes, 24 May 2006.

[10] M. B. Mme Dougherty, «Les formats de compression d'image,» Département Génie Électrique et Informatique Industrielle, Institut Universitaire de Technologie de Tours, Promotion 2002-2004.

[11] Y. Fisher, «FRACTAL IMAGE COMPRESSION,» Super Computer Center, University of California, San Diego.

[12] J. M. Florian AGEN, «La Compression Fractale, Méthodes de Jacquin, Subdivisions de triangles et Delaunay,» UNIVERSITE François Rabelais TOURS Polytech'Tours-Département Informatique, Juin 2005.

[13] N. H.-S. Adda ALI-PACHA, «Compression des Images Fixes par Fractale : Partitionnement Quadtree,» 2007.

[14] «Guide de Network Associates International BV sur la cryptographie».

[15] S. Tag, Écrivain, *Support de Cours de 1er Année Master Sécurité Informatique.* [Performance]. 2018-Octobre.

[16] C. E. SHANNON, «A Mathematical Theory of Communication,» *The Bell System Technical Journal,*, p. 55, July, October, 1948..

[17] A. L. DAHMANE Zouhir, «Implémentation d'un algorithme de cryptage sur un circuit FPGA,» chez *MEMOIRE DE MASTER*, Mai 2017, pp. 8-12.

[18] B. Rabab, «THESE DE DOCTORAT 3ème Cycle Sécurité des images Numériques compressées JPEG,» Université Djilal Liebes– Sidi Bel Abbes.

[19] J.-M. C. Franck Davoine, «COMPRESSION D'IMAGES PAR FRACTALES,» Laboratoire TIMC-IMAG Institut Albert Bonniot, LA TRONCHE Cedex France.

[20] a. A. Veenadevi, «FRACTAL IMAGE COMPRESSION USING QUADTREE DECOMPOSITION AND HUFFMAN CODING,» *An International Journal (SIPIJ) Vol.3, No.2,* n° %1209, pp. 209-210, April 2012.

[21] Z. B. K. A. S. A. Z. Hamid Meraoubi, «Un système de crypto-compression des images médicales basé sur la DCT 2x2-IDS et l'AES,» Centre de Développement des Technologies Avancées, Baba Hassen, Alger.

[22] S. Guillem-Lessard, «Tutoriel de Cryptographie,,» 2002.

[23] A. C. e. F. Lévy-dit-V'hel, «La cryptographie moderne, Revue Armement,» 2001.

[24] P. Deshmukh, «An image encryption and decryption using AES algorithm,» *International Journal of Scientific & Engineering Research,* Vols. %1 sur %2Volume 7, Issue 2, n° %1212, pp. 210-213, February-2016.

Dédicaces

A mes chers parents,

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes très chère, respectueux et magnifiques parents qui m'ont soutenus tout au long de ma vie, dont leurs mérites, leurs sacrifices, leurs qualités humaines m'ont permis de vivre ce jour : les mots me manquent pour exprimer toute la reconnaissance, la fierté et le profond amour que je vous porte pour les sacrifices qu'ils ont consenti pour ma réussite, qu'ils trouvent ici le témoignage de mon attachement ma reconnaissance, gratitude et respect, que Dieu leur préservent bonne santé et longue vie. Tous mes sentiments de reconnaissance pour vous.

A toute ma Famille « mes chères sœurs et mes frères, »

En témoignage de mes sincères reconnaissances pour les efforts qu'ils ont consenti pour l'accomplissement de mes études. Je leur dédie ce modeste travail en témoignage de mon grand amour et ma gratitude infinie.

A tous mes amis sans exception,

Pour leur aide et leur soutien moral durant l'élaboration du travail de fin d'études.

B.Youcef Islem

REMERCIMENT

C'est avec un grand plaisir que je réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail.

Je tiens tout d'abord à remercier Dr. BENNOUR Akram, maitre de conférences à l'université Larbi Tébessi, président du jury et Mr. GAHMOUS Abdellatif, maitre-assistant à l'université Larbi Tébessi examinateur, pour l'honneur qu'ils m'ont accordé en acceptant de juger mon travail

Je remercie sincèrement Dr. Menassel Rafik, maitre de conférences à l'université Larbi Tébessi, pour son encadrement, son assistance, son soutien, sa disponibilité et ses précieux conseils durant la période de ce stage.

Je m'acquitte, enfin, volontiers d'un devoir de gratitude et de remerciements à tous mes enseignants pour la qualité de l'enseignement qu'ils ont bien voulu me prodiguer durant mes études afin de me fournir une formation efficiente.

B.Youcef Islem

Table des matières :

Table des Figures :

Table des Tableaux :

Résumé:

La compression et le cryptage des données représentent deux technologies dont l'importance est en croissance exponentielle et ce dans une multitude d'applications. De plus, l'utilisation excessive de réseaux informatiques pour le transfert de données doit évidemment obéir à un double objectif : la réduction du volume de données afin d'encombrer le moins possible les réseaux de communication publics et la confidentialité afin d'assurer un niveau optimal de sécurité.

D'une part, la compression d'images a pour but de réduire la taille d'une image afin de faciliter son stockage aussi bien que son transfert. Ainsi on distingue deux grandes familles de méthodes compression, à savoir celles qui provoquent des pertes d'information causant une image reconstruite non fidèle à l'originale mais de taille très réduite. Les autres méthodes ne provoquent pas de perte d'information mais présentent des taux de compression réduits.

D'autre part, le cryptage des données est généralement décrit à partir d'une communication secrète d'informations entre deux interlocuteurs. Dans un système informatique, cette confidentialité intervient dans plusieurs formes, en particulier dans la protection du stockage, de l'accès et transmission de l'information.

C'est dans ce contexte que inscrit notre travaille et qui consiste à combiner à la fois des techniques de compression et de cryptage des images, où on va appliquer un algorithme de compression fractale et un algorithme de cryptage basé sur le block Cipher. Pour la compression nous avons utilisés une approche de compression hybride basé sur la décomposition Quadtree et le codage de Huffman pour terminer le codage et calculer les paramètres de performances du compression, tel que (CR, MSE, PNSR, l'histogramme), pour le chiffrement à base de block Cipher nous appliqué l'algorithme de cryptage L'AES, et enfin nous avons calculés les temps d'exécution pour chaque opération tel que (temps de compression/décompression, temps de cryptage/décryptage).

Abstract:

Data compression and encryption are two technologies that are growing exponentially in a multitude of applications. In addition, the excessive use of computer networks for data transfer must obviously obey a double objective: the reduction of the volume of data in order to clutter the public communication networks as little as possible and confidentiality in order to ensure a level optimal security.

On the one hand, image compression aims to reduce the size of an image in order to facilitate its storage as well as its transfer. Thus there are two main families of compression methods, namely those which cause information loss causing a reconstructed image not faithful to the original but of very small size. The other methods do not cause loss of information but have reduced compression rates.

On the other hand, data encryption is generally described on the basis of a secret communication of information between two interlocutors. In a computer system, this confidentiality intervenes in several forms, in particular in the protection of storage, access and transmission of information.

It is in this context that our work takes place, which consists in combining both image compression and encryption techniques, where we will apply a fractal compression algorithm and an encryption algorithm based on the Cipher block. For compression we used a hybrid compression approach based on Quadtree decomposition and Huffman coding to complete the coding and calculate the compression performance parameters, such as (CR, MSE, PNSR, histogram), for the Cipher block-based encryption we applied the AES encryption algorithm, and finally we calculated the execution times for each operation such as (compression / decompression time, encryption / decryption time).

INTRODUCTION GENERALE

Introduction générale :

L’utilisation des technologies de l’information 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.

Ensuite, avec la grande accélération dans le développement des technologies d'Internet et de la communication, la communication des images. Cependant, la sécurité de l'information est un sujet sensible pour la recherche, la discussion et le développement, et le cryptage est l'une des meilleures alternatives qui s'est avérée efficace tout au long de l'histoire pour assurer la confidentialité et la sécurité de l'information.

Les algorithmes de chiffrement par blocs appliqués aux images représentent deux inconvénients. Premièrement, quand l’image contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement. Dans ce cas, l’image cryptée contient des zones texturées et l’entropie de l’image n’est pas maximale. Le second problème est que les méthodes de cryptage par blocs ne sont pas robustes au bruit. En effet, une erreur sur un bit chiffré va propager des erreurs importantes dans tout le bloc courant.

Dans ce travaille nous proposons un système de crypto-compression qui utilise les deux techniques la compression fractale, et le chiffrement par bloc AES. Pour mener à bien notre travail, nous avons structuré notre mémoire en trois chapitres :
Le premier chapitre est l’état de l’art est divisé en deux partie ; la compression et la cryptographie :

- La première partie 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 et enfin ont mettant le point sur la compression fractale.
- La deuxième partie nous amène dans le monde de la cryptographie ; en commençant par une définition détaillée suivi par une présentation des méthodes de cryptage parmi les plus utilisées, ensuit nous expliquons les deux types de la cryptographie (classique et moderne), et enfin nous mettons le point sur la cryptographie par bloc.

Le deuxième chapitre introduit la notion de Crypto-Compression, nous présentons une explication plus détaillés des deux techniques utilisées dans notre système, en commençant par la compression fractale par bloc et nous décrivons par la suite l'algorithme générique de compression et décompression fractale, ensuite nous l'approche de cryptage à base de Block Cipher en utilisant l'algorithme de chiffrement AES.

Le Troisième chapitre comprend la partie la plus importante de ce travail et qui sera consacré à l'approche de crypto-compression élaborée dans le cadre de ce travail ; en commençant par une présentation de l'environnement de travail et un aperçu générale de notre système, nous citons par la suite les tests expérimentaux sur des images réelles et les résultats obtenues.

PREMIER CHAPITRE NOTION THEORIQUE ET ETAT DE L'ART

Introduction :

Les applications qui utilisent des données multimédias, comme les images, sont de plus en plus présentent dans notre vie quotidienne. Ainsi manipuler les images (stocker ou transmettre,…) devient un enjeu stratégique. Du 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 sont simples, rapides et efficaces [1]. En effet dans cette partie nous avons présentés les deux technologies utilisées dans notre système la compression et le cryptage des données. La première partie est consacrée sur la compression d'images, leur principe, les types de compressions, les caractéristiques des méthodes de compression et enfin, on met le point sur la compression fractale. La deuxième partie concerne le cryptage des données, leur principe, leur objectifs, la cryptographie classique et moderne (symétrique et asymétrique) et enfin, on met le point sur le block Cipher.

I. La compression :

1. Principe :

La compression des données, d'une maniéré générale, est l'ensemble des méthodes ou techniques que l'on utilise pour prendre un message long pour en faire un message court, c'est-à-dire réduire le volume d'une donnée sans perdre les informations essentielles. Le but de la compression est de représenter les données avec une forme plus compacte que l'original où le résultat de la compression occupe moins d'espace que la donnée originale. Les données peuvent être compressées avec perte ou sans perte. [2]. La Figure 1 présente le principe de la compression des images, de manière nous supposons que l'opération de compression est une fonction f (x, y), alors que l'opération de décompression c'est l'opération inverse que celui de l'opération de compression (f prime de x, y). :

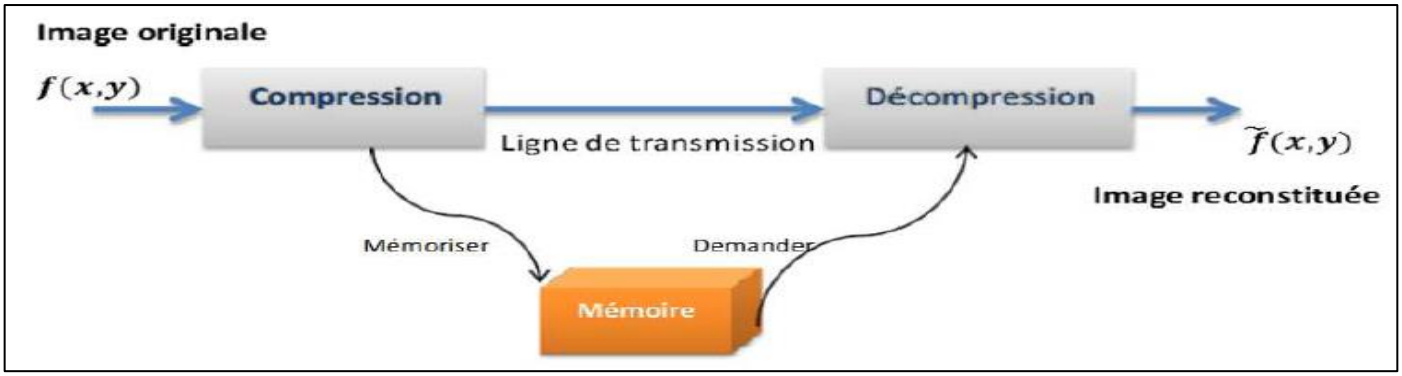


Figure 1 : Principe de Compression d'images [3]

Les méthodes de compression d'images ont des critères d'évaluations, dont on peut citer : [4]

- La qualité de reconstitution de l'image.
- Le taux de compression.
- La rapidité du codeur.

Le schéma fonctionnel de la compression est présenté dans la Figure 2 ci-dessous, ce schéma présente le processus de codage de donnée et qui se par la transformation ensuite la quantification et enfin le codage :

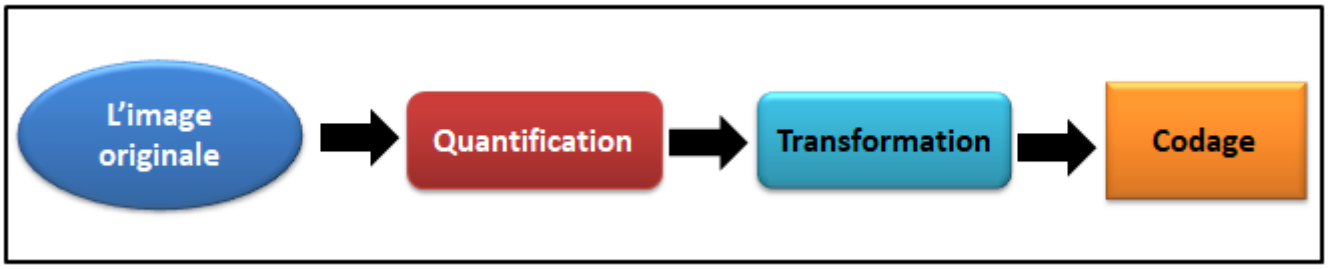


Figure 2 : Schéma d'un codeur d'image [4]

A partir de ce schéma, nous allons revoir chacune de ses étapes à fin de préciser leur rôle.

- **Transformation:**
  - La dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un [1]**pixel** à un pixel voisin) traduisent une corrélation très forte sur l'image.

- **Quantification :**
  - La quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression.

- **Codage :**
  - Une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire a priori les deux conditions suivantes:
    - Unicité : deux messages différents ne doivent pas être codés de la même façon.
    - Déchiffrable : deux mots de codes successifs doivent être distingués sans ambiguïté. [4]

[1] **Pixel :** Le pixel est souvent abrégé p ou px. Il est l'unité de base permettant de mesurer la définition d'une image numérique matricielle. Son nom provient de la locution anglaise picture element, qui signifie « élément d'image ».

2. Compression physique et logique :

- **La compression physique :** s'applique uniquement aux données de l'image. Il s'agit de translater les trains de bit d'un motif à un autre.
- **La compression logique :** est effectuée par un raisonnement logique en substituant une information par une information équivalente. [5]

3. Compression symétrique et asymétrique :

Dans le cas de la compression symétrique, la même méthode est utilisée pour compresser et décompresser l'information, il faut donc la même quantité de travail pour chacune de ces opérations.

La compression asymétrique demande plus de travail pour l'une des deux opérations.

4. Caractéristiques des méthodes de compression :

4.1. Rapport et taux de compression :

Le rapport de compression est le nombre de bits utilisés par l'image originale et du nombre de bits utilisés par l'image compressée. [6]

$$CR^{2} = rapportc = \frac{nombre\ de\ bits\ avants\ compression}{nombre\ bits\ aprés\ compression}$$ Équation 1

Le taux de compression est un pourcentage de l'espace obtenu après la compression par rapport à l'espace total requis par les données avant la compression. Qu'un fichier compressé indique à sa taille originale aura un taux de compression de 0 %. Un fichier réduit à 0 octet, aura un taux de compression de 100%. [7]

$$Tc = TauxC = \left(1 - \frac{1}{CR}\right) * 100$$ Équation 2

4.2. Mesure de distorsion

Pour mesurer la distorsion entre l'image reconstruite et l'image originale (Mesure de la qualité visuelle de l'image reconstruite) en va utiliser l'Erreur Quadratique Moyenne [3]MSE (Mean Square Error) ou du rapport signal à bruit PSNR[4] (Peak Signal to Noise Ratio).

Etant donnée une image originale composée de pixels αi(i=1...N) et l'image décodée composée de pixels άi(i=1...N) . Alors l'erreur quadratique moyenne est donnée par : [7]

$$MSE = \frac{1}{N}\sum_{1}^{N}(\alpha i - \acute{\alpha} i)^{2}$$ [7] Équation 3

L'autre critère objectif cité plus haut est le rapport signal sur bruit de l'image reconstruite PSNR (en anglais Peak Signal to Noise Ratio). Il est défini par l'équation : [1]

[2] **CR : En anglais « *Compression Ratio* » : c'est le rapport de compression**.
[3] **MSE** : En anglais « *Mean Square Error* » c'est l'Erreur Quadratique Moyenne**.**
[4] **PSNR**: En anglais « *Peak Signal to Noise Ratio* » c'est le rapport signal à bruit.

$$PSNR = 10 \log \frac{(2^8-1)}{MSE} \text{ DB}^5 \text{ (décibels) [7]} \quad \text{Équation 4}$$

$$\text{Ou: } PSNR = 10 \log \frac{d^2}{MSE} \text{ [1]} \quad \text{Équation 5}$$

d représente la valeur d'intensité maximale de l'image. En utilisant des images en niveau de gris, les valeurs sont ainsi codées sur 8 bits et dans ce cas de figure $d \leq 2^8 - 1$. [1]

4.3. Entropie:

Est une métrique de « surprise » dans la mesure où, si l'entropie est élevé, les prédictions sont difficiles à faire et si l'entropie est faible, la séquence facilement prévisible si une séquence contient beaucoup de « surprise », elle contient beaucoup d'information. Si elle ne contient pas beaucoup de « surprise », elle ne contient pas beaucoup d'information Est définie par la formule suivante : [2]

$$H = -\sum_{K=0}^{2^R-1} P(K) \log_2 P(K) \ bpp \quad \text{Équation 6}$$

Avec : P(K) est la probabilité d'apparition des niveaux de gris dans l'image, K est la valeur de gris et R est le nombre de bits par pixels. [6]

4.4. 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. [8]

5. Les objectifs de la compression :

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante des réseaux en fonctionne de cette dernière, nous citons les objectifs de la compression : [8]

- La rapidité de la compression et la décompression.
- Réduire énormément la taille des images.
- Le temps de transmission gagné par une réduction de la taille des données.
- Garantie de non perdu de l'image entière grâce à la robustesse de l'algorithme de compression.

[5] **DB** : Le décibel (dB) est une unité définie comme dix fois le logarithme décimal du rapport entre deux puissances1, utilisée dans les télécommunications, l'électronique et l'acoustique.

- Deux qualités le taux de compression et la qualité de l'image après un cycle de compression\décompression.

6. Type de compression :

Les types de compression sont faits grâce aux redondances des données présentes sur l'image, ces redondances sont :

- **Redondance psycho visuel** : des détails non perceptible à l'œil humain qu'on peut d'éliminé (caractéristiques de l'œil humain).
- **Redondance inter pixel** : la possible corrélation existante entre les pixels de l'image, on dit qu'une image a une redondance inter pixel si c'est possible de prédire la valeur d'un pixel en connaissance de la valeur des pixels voisin (suivant ou précédent), sachant que plus la résolution de l'image est grande plus la possibilité de rencontrer des redondances inter pixel est élevée.
- **Redondance de codage** : séquence de répétition des bits, on rencontre généralement à la fin de la compression, pendant l'étape de codage. [9]

On peut distinguer deux grandes types de la compression (Figure 3) : 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. [7]

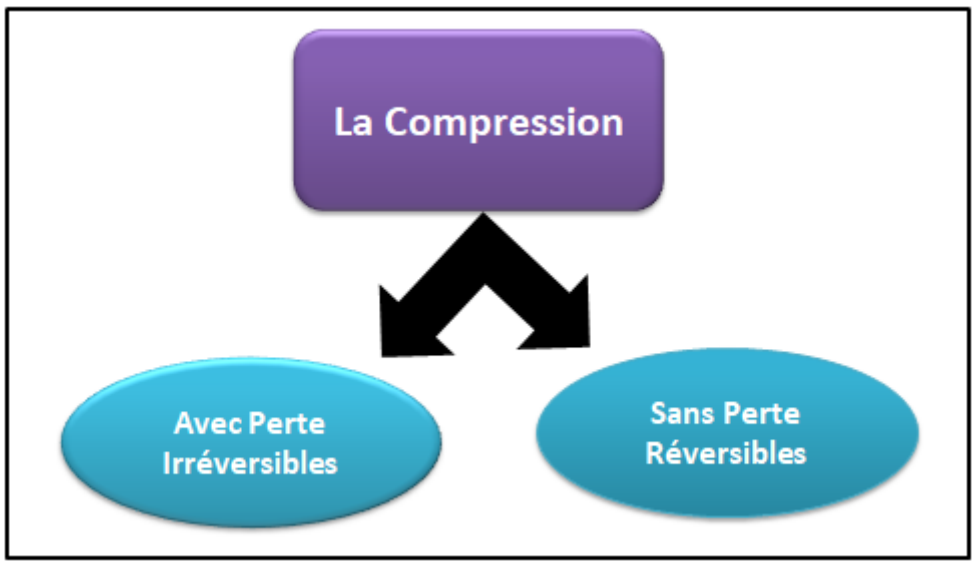


Figure 3 : Types de Compression

6.1. La compression sans perte :

Comme son nom l'indique, ce type de compression n'occasionne aucune perte de données, appelle aussi compression non destructrice. La qualité de l'image après décompression est la même que celle de l'image original, le taux de compression de ce type est limite. [10]

D'autre manière on donne un exemple plus simple comme il montre dans la (Figure 4), un ensemble de bits X a comme résultat de compression le compressé Y plus court que X, ce résultat est stocké ou transmis. Lors de la décompression du résultat Y, on récupèrera par exactitude l'ensemble de bits de départ X. [2]

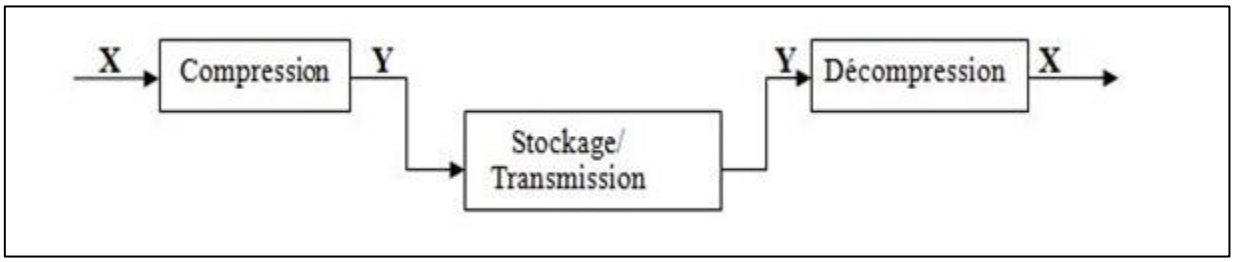


Figure 4 : Compression sans perte [2]

Ce type de compression est nécessaire pour certaines applications ou la précision est majeure telles que les images médicales (**IRM**[6]**)** ou la télédétection (imagerie satellite).

Les algorithmes de compression employée sont nombreux, les plus importants sont : [7]

- Codage de Huffman.
- Codage de Shannon-Fano.
- Codage arithmétique.
- Le codage par répétition ou "Run Length Coding" **(RLC)**[7]
- Codage par dictionnaire adaptatif **(LZW)**[8] (Lempel-Ziv-Welch) ou LZ77

6.2. Compression avec perte :

C'est une compression destructrice, les techniques de compression avec perte impliquent une certaine perte d'information ou d'avoir une permission de supprimer quelques données (qui sont inutiles) de l'information pour avoir une meilleure compression, ce qui signifie les données qui ont été compressées à l'aide de techniques à perte ne peuvent généralement pas être récupérées ou reconstruites exactement. [3]

D'une façon simplifié on donne un exemple comme il montre dans la (Figure 5), On a une information X qu'on veut compresser, son résultat de compression est Y, après décompression du compressé Y on a un résultat Z qui est une approximation de X (Z=X). [2]

[6] **IRM :** L'imagerie par résonance magnétique (IRM) est une technique d'imagerie médicale permettant d'obtenir des vues en deux ou en trois dimensions de l'intérieur du corps de façon non invasive avec une résolution en contraste relativement élevée.

[7] **RLC :** Le run-length encoding, appelé en français le codage par plages, est un algorithme de compression de données sans perte en informatique.

[8] **LZW** : LZW (pour Lempel-Ziv-Welch) est un algorithme de compression de données sans perte. Il s'agit d'une amélioration de l'algorithme LZ78 inventé par Abraham Lempel et Jacob Ziv en 1978. LZW fut créé en 1984 par Terry Welch, d'où son nom.

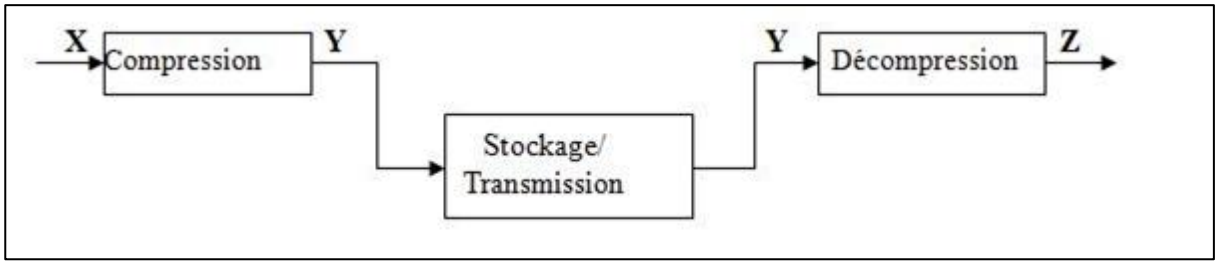


Figure 5 : Principe de compression avec Perte [2]

Ce type de compression on le trouve généralement dans le domaine-là où la réduction du poids de limages est très importante, comme le domaine multimédia par exemple (Web, photographie) où la fidélité envers l'image originale n'est pas très importante et le taux de compression sera plus grand que celui d'une compression sans perte du fait qu'on est juste limites par la qualité qu'on souhaite obtenir. [3]

Les algorithmes de compression employée sont nombreux, les plus importants sont :

- Quantification.
- Codage par transformée.
- Le codage en sous-bandes.
- La compression par ondelettes.
- **La compression fractale.**

7. La compression fractale :

7.1. Pourquoi la compression par fractales ?

- La compression Fractale est une méthode récente, qui s'applique uniquement aux images. Le format des images compressé par ce procèdé, n'est pas encore standardisé à ce moment. Elle a été proposée pour la première fois par Michael Barnsly 1988. [7]
- Les méthodes standard de compression d'images peuvent être évaluées par leur taux de compression: le quotient de la mémoire occupée par l'image comme collection de pixels par la mémoire nécessaire pour stocker l'image sous forme compressée.
- Le taux de compression d'images par fractales est élevé, ce qui justifie notre choix de cette méthode de compression. [11]

➢ Principe :

- La compression d'images fixes par une méthode fractale utilise les propriétés bien connues des fractales : La récurrence des motifs. Ce genre de compression tend à éliminer la redondance d'informations dans l'image, en recherchant tous

les motifs, toutes les zones de l'image qui se répètent dans l'image. La récurrence des motifs s'effectue parfois de manière directe (seule l'échelle est différente), et parfois de manière indirecte (transformation, rotation, etc..). [12]

7.2. L'image Fractale :

Une image fractale est une figure géométrique ou est un ensemble fini de transformations géométriques (rotation, translation, agrandissements, réductions) appliquées aux sous ensemble et motifs identiques et de tailles variables qui la composent. [13]

7.3. Objet fractal :

C'est un objet naturel dont la forme est extrêmement irrégulière et de plus (éventuellement) interrompue et fragmentée. [13]

7.4. Auto- similarité :

Un objet est dit auto - similaire si sa forme est la même à n'importe quel grossissement : un détail de l'objet une fois agrandie, reproduit exactement une partie plus grande de l'objet. Cependant il existe des objets fractals qui ne sont pas exactement auto- similaires, l'objet ne contient pas une transformation de lui-même, mais des parties de lui-même, c'est ce qu'on appelle l'auto- similarité locale. La Figure 6 présent le Triangle de Sierprinski qui montre le principe d'auto- similarité. [13]

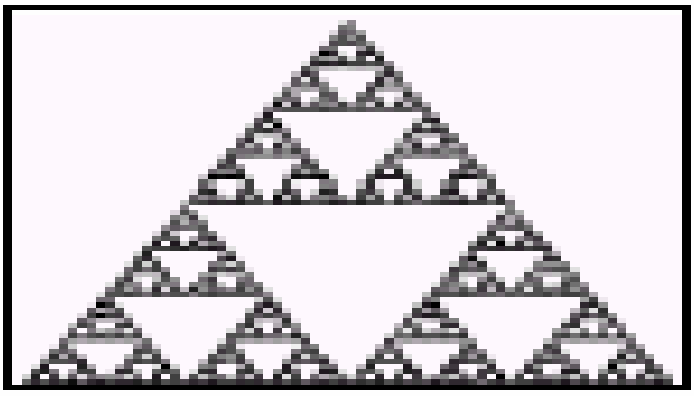


Figure 6 : Triangle de Sierprinski [7]

7.5. Les méthodes de compression fractale :

Il existe plusieurs méthodes de compression fractale, et nous avons choisis de travailler avec l'une d'entre elles :

- La compression Jacquin.
- La compression par subdivisions successives de triangles.
- La compression de Delaunay.

Ces trois algorithmes sont fondés sur une même idée :

- Définir une zone dans l'image (un carré ou un triangle suivant la méthode utilisée),
- Appliquer des transformations fondamentales (réduction, normalisation de la moyenne, rotation etc...)
- Réimplanter le résultat dans l'image dans une zone plus réduite (Destination). La zone de destination doit être la plus 'proche' possible du résultat.
- La répétition de ce modèle dans toute l'image, et ceci de manière itératif, ce qui implique une certaine convergence de l'image reconstruite par l'assemblage des transformations/réductions locales, vers l'image d'origine. [12]

7.5.1. Méthode Jacquin :

➢ Principe :

Le principe de l'algorithme de compression par la méthode Jacquin c'est :

- Correspond parfaitement à l'algorithme générique présenté dans le deuxième chapitre.
- Utilise des régions carrées pour segmenter l'image.
- Cet algorithme de compression/décompression est le plus simple que les trois algorithmes présentés précédemment.
- Consiste a manipulé des régions carrées prédéfinies (tableaux 2D), et que cette dernière est beaucoup plus simple à implémenter que les régions triangulaires dynamiques.
- La méthode Jacquin consiste à utiliser quatre étapes importantes nous verrons par la suite dans le chapitre 2. [12]
- La Figure 7 présente le principe fonctionnement de la compression fractale par la méthode de Jacquin.

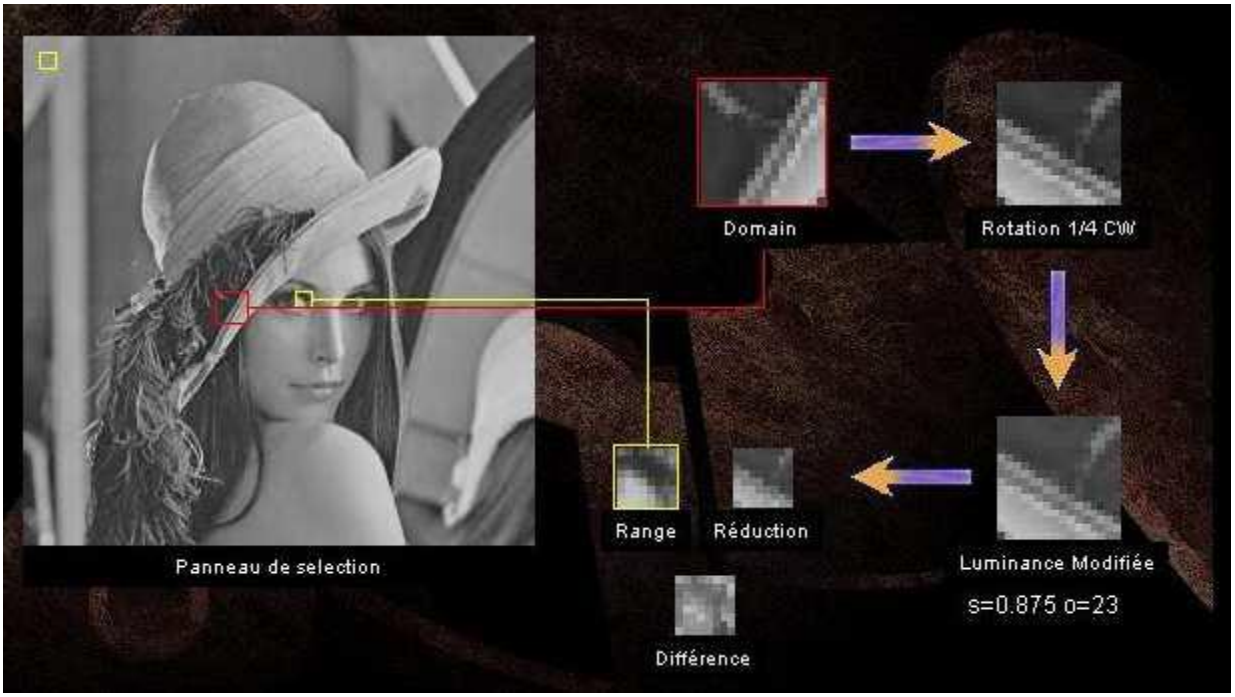


Figure 7: Principe de méthode Jacquin [12]

7.5.2. Méthode par subdivisions de triangles :

➢ Principe :

- La méthode de compression par subdivisions de triangles est quasiment identique à la méthode Jacquin. Elles possèdent cependant deux différences majeures. D'une part, nous travaillons maintenant sur des triangles, et d'autre part, un module de subdivision des triangles a été ajouté. Mais avant de présenter la subdivision de triangles en elle-même, il est n´nécessaire d'évoquer la méthode utilisée pour gérer les triangles.
- Travailler sur un carré est relativement simple, il suffit de définir la hauteur du carré, et un point de d´épart (en haut à gauche en général). Un triangle est plus complexe, il faut soit 3 points, soit 3 droites.
- Ainsi, pour savoir si un point (ou un pixel) appartient ou non à un triangle, nous avons choisi de modéliser un triangle à l'aide de 3 droites d1, d2 et d3, comme le montre la Figure 8. [12]

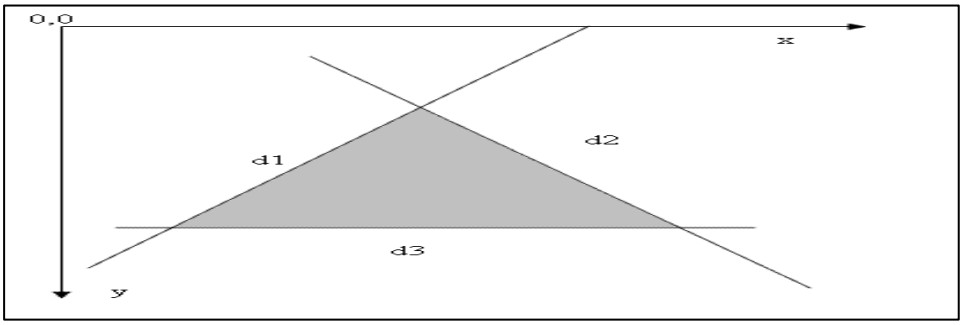


Figure 8 : Appartenance d'un point à un Triangle [12]

- Pour connaître l'appartenance d'un point à un triangle, il suffit de vérifier les équations définies ci-dessous :

$$P.x \geq d1 \quad P.y \geq d1$$
$$P.x \geq d2 \quad P.y \geq d2$$
$$P.x \geq d3 \quad p.y \geq d3$$

- Si un point P (de coordonnées (P.x,P.y)) appartient au triangle formé par les segments d1, d2 et d3, alors il vérifie ces inéquations.

7.5.3. La subdivision :

Deux points essentiels pour lancé l'opération de subdivision :

- Découpe l'image à traiter en un pavage de triangles. Pour améliorer l'efficacité de l'algorithme de compression, une division des triangles ayant un nombre important de détails est ajoutée.
- Pour connaître le niveau de détails d'une zone de l'image (couverte par un triangle considéré), nous calculons simplement l´écart type des pixels de celui-ci. Plus cette valeur sera importante, et plus le triangle est loin d'être uniforme. [12]

7.5.3.1. Les étapes de subdivision de Triangles :

Le triangle sera divisé en 6 plus petits triangles comme le montre la Figure suivante :

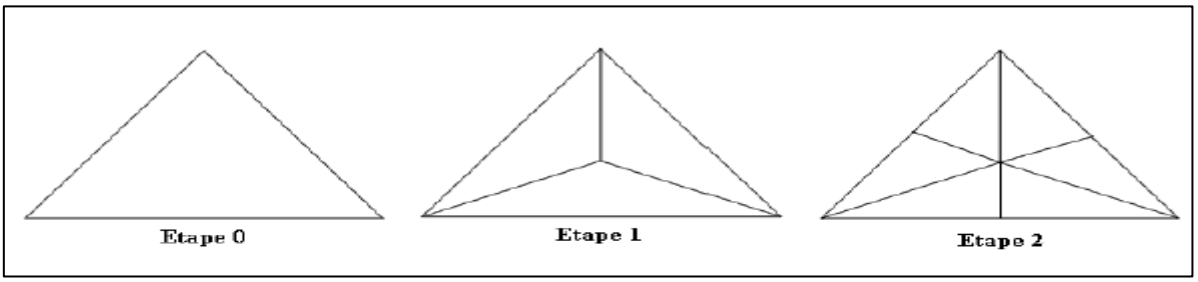


Figure 9 : Subdivisons de Triangles [12]

- **L'étape 1 :** nous divisions une première fois le triangle en 3 triangles, qui seront formés d'une part, avec les 3 points du triangle parent, et d'autre part avec son barycentre.
- **L'étape 2 :** va ensuite scinder les 3 nouveaux fils en 2, en utilisant la médiane (partant de l'ancien barycentre) de chacun.
- **L'étape 3 :** Cette subdivision amélioré considérablement le rendu de l'image compressée. La perte de qualité est diminuée, mais le temps de compression impose une profondeur de division restreinte. De plus, la taille du fichier de sortie augmente car il est nécessaire de stocker les différentes subdivisions. Un simple booléen suffit (1 : triangle divisé, 0 : triangle non subdivisé). [12]

➢ Transformation d'un triangle :

- Le processus discrétisé d'une réduction, qu'elle soit avec un carré (méthode de Jacquin) ou avec un triangle, reste identique. Il s'agit simplement de caractériser une matrice de transformation des coordonnées d'une figure de d'épart, vers les coordonnées d'une figure d'arrivée. Prenons l'exemple de deux triangles montré dans la Figure 10 : [12]

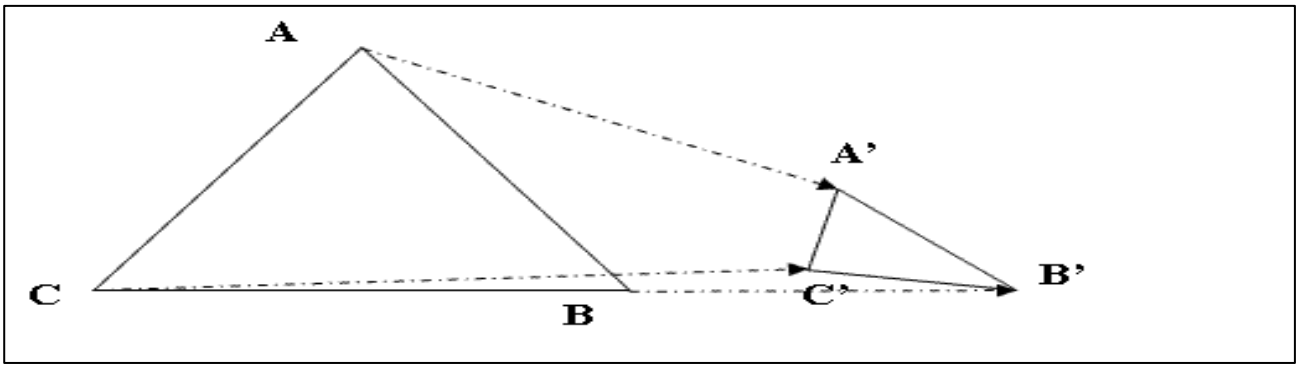


Figure 10 : Transformation d'un Triangle [12]

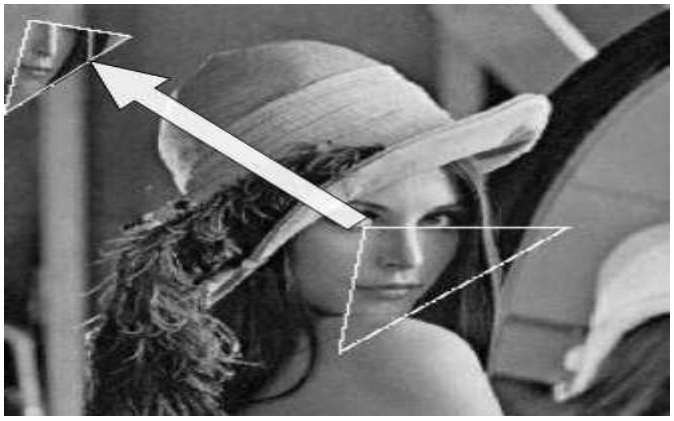


Figure 11 : Réduction d'un Triangle [12]

➢ Partitionnements :

- Le partitionnement des figures Sources et Destinations par la méthode de subdivision de triangles sont identiques à ceux de la méthode de Jacquin. Il s'agit ici, de réaliser un pavage de triangles (et non plus de carrés), avec un pas déterminée. Le pavage Destination est bien entendu plus compact que le pavage Source, pour conserver le caractère fractale de la compression. [12]
- La Figure suivante montre les deux pavages Source et Destination sur l'image lena.jpg.

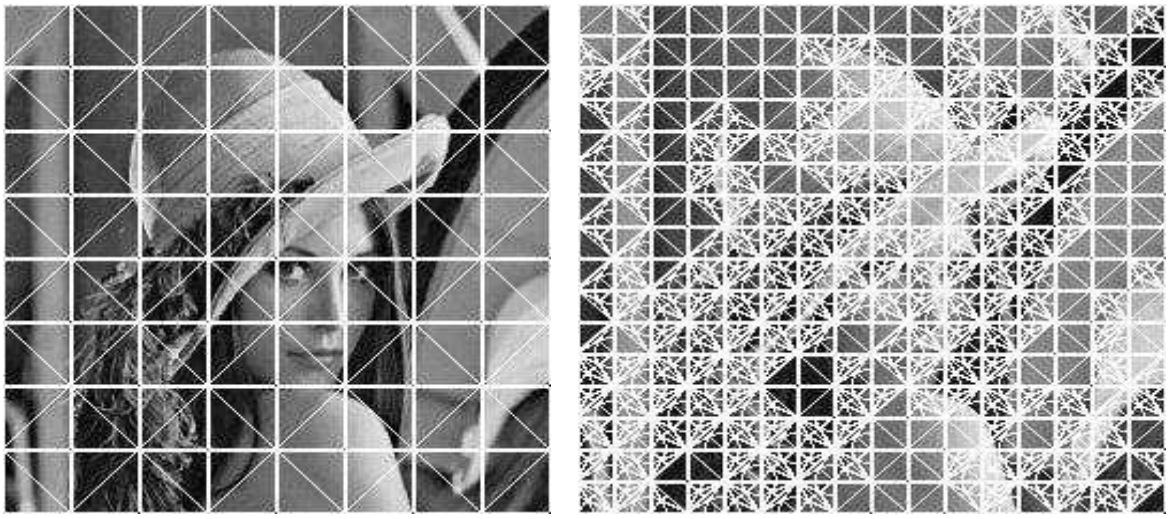


Figure 12 : Pavage Source et Destination, Méthode par subdivisions se Triangles [12]

7.5.4. Méthode de Delaunay :

➢ Principe :

La dernière méthode de compression fractale, est la compression de Delaunay. Cette méthode manipule tout comme la Subdivision de triangles, des figures triangulaires pour segmenter l'image. Mais au lieu de partitionner suivant un pavage régulier de triangles, la méthode de Delaunay utilise comme son nom l'indique, un pavage formé par les triangles de Delaunay. Seul, le pavage Destination est formé ainsi, le partitionnement Source étant un simple pavage régulier.

Pour augmenter encore la qualité de restitution des détails lors de la reconstruction de l'image, l'algorithme de compression de Delaunay segmente l'image en triangles de Delaunay. [12]

7.5.4.1. Triangulation de Delaunay :

La construction des triangles de Delaunay à partir d'un diagramme de Voronoi est assez simple. En effet, l'algorithme de Delaunay recherche premièrement les germes voisins (donc séparés par un arrête de Voronoi), pour chaque germe du Diagramme, et les relie entre eux. Comme le montre la Figure suivante : [12]

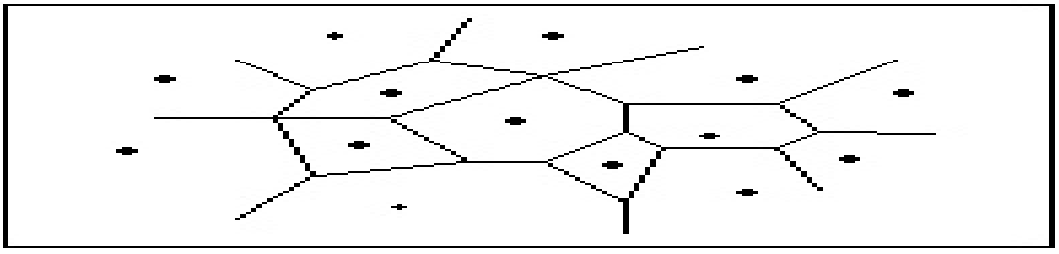


Figure 13 : Digramme de Voronoi [12]

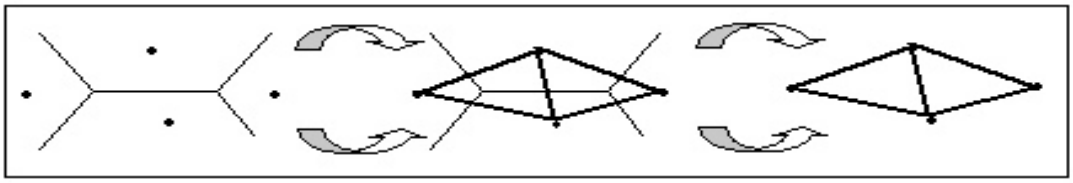


Figure 14 : Du diagramme de Voronoi aux Triangles de Delaunay [12]

La conception de ce genre de segmentation étant assez complexe à réaliser.

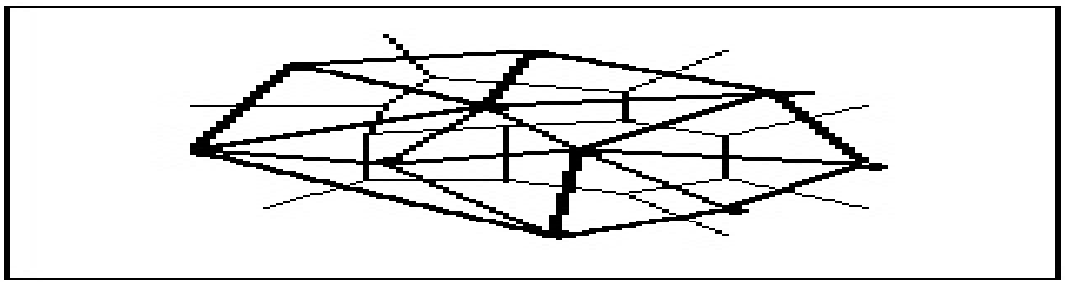


Figure 15 : Diagramme de Voronoi, Partitions de Delaunay [12]

La Figure 16 présente la Pavage source et destination selon la méthode de Delaunay, ce que nous pouvons dire sur la figure de droite est que les zones homogènes sont composées de triangles assez grands (notamment au-dessus du chapeau de Lena, et à droite de l'image). Les détails sont quant à eux localisés par de plus petits triangles (les cheveux de Lena). [12]

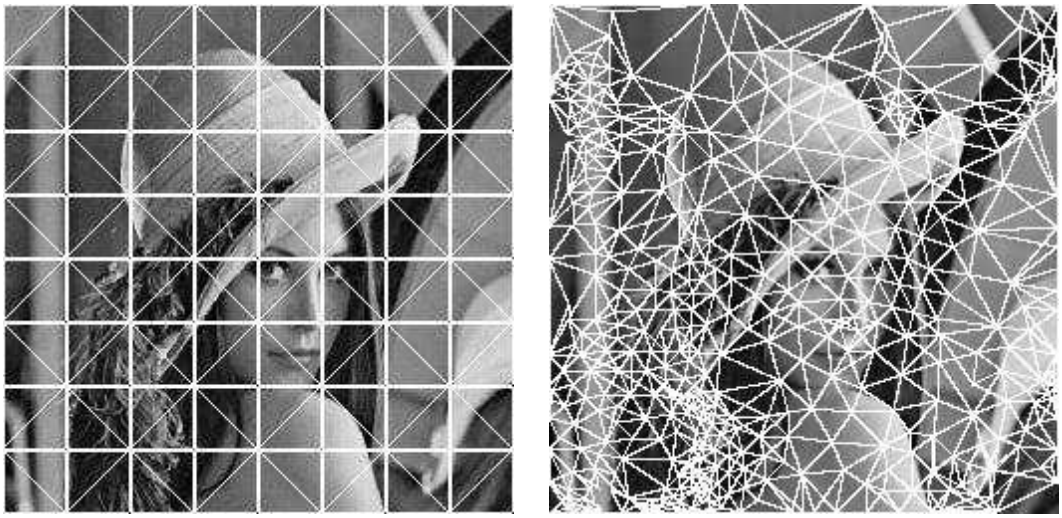


Figure 16 : Pavage Source et Destination, Méthode de Delaunay [12]

II. La Cryptographie :

1. Définition :

La cryptographie est la science qui utilise les mathématiques pour le cryptage et le décryptage de données. Elle nous permet ainsi de stocker des informations confidentielles ou de les transmettre sur des réseaux non sécurisés (tels que l'Internet), afin qu'aucune personne autre que le destinataire ne puisse les lire.

Alors que la cryptographie consiste à sécuriser les données, la cryptanalyse est l'étude des informations cryptées, afin d'en découvrir le secret. La cryptanalyse classique implique une combinaison intéressante de raisonnement analytique, d'application d'outils mathématiques, de recherche de modèle, de patience, de détermination et de chance. Ces cryptanalyses sont également appelés des **pirates**[9]**.** [14]

Cryptologie = Cryptographie + Cryptanalyse.

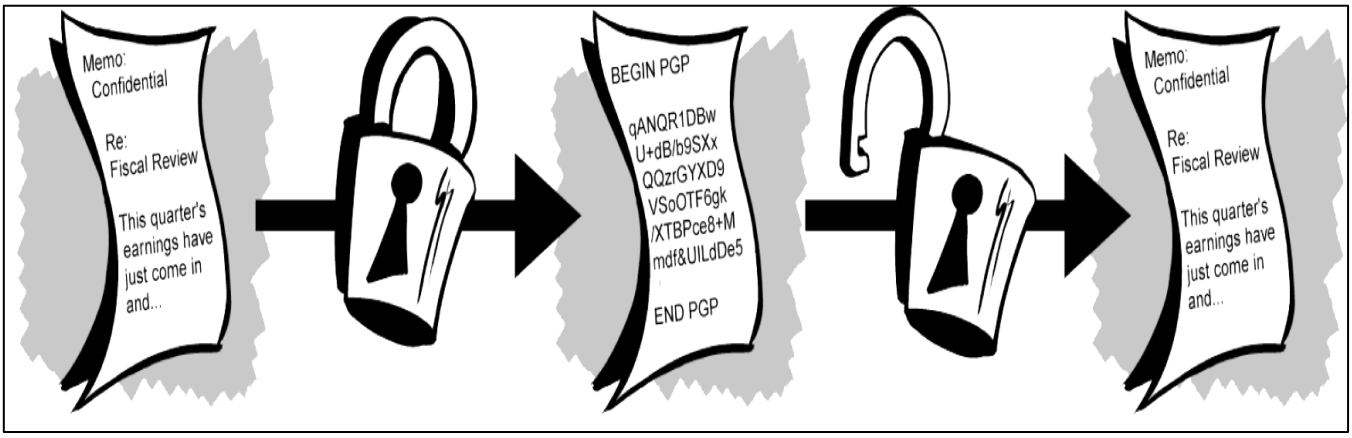


Figure 17 : Schéma générale de la Cryptographie [14]

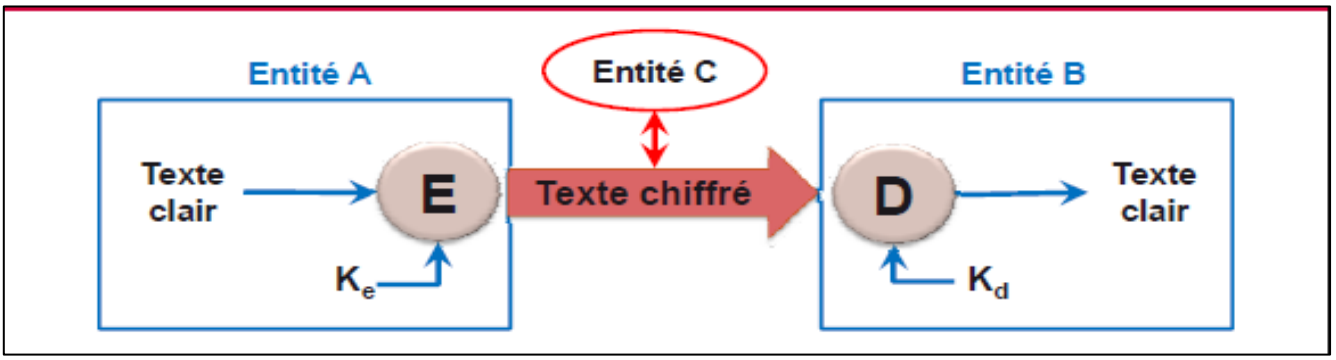


Figure 18 : Schéma de processus de cryptage et décryptage [15]

[9] **Pirate :** un pirate ou hacker en sécurité informatique, un hacker, francisé hackeur ou hackeuse, est un spécialiste d'informatique, qui recherche les moyens de contourner les protections logicielles et matérielles.

2. Les objectifs de la cryptographie

Les principaux services à garantir par l'application de la cryptographie sont:

- **Confidentialité de l'information** : le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé « Organisation internationale de normalisation (**ISO**[10]) »
- **Intégrité de l'information** : L'information ne doit subir aucune altération ou destruction volontaire ou accidentelle, lors de son traitement, conservation ou transmission.
- **Détection de l'altération de l'information par une entité non-autorisée.**
- **Non-répudiation** : Empêcher qu'une entité réfute des actions ou engagements antérieures.
    - Vérifier que l'expéditeur et le destinataire sont les entités qui ont envoyé ou reçu l'information. [15]

3. Cryptographie classique :

La cryptographie classique a été conçue avant la création des ordinateurs et qui ont donné les concepts et les bases pour l'évolution de plusieurs algorithmes symétriques encore utilisés de nos jours.

Les **crypto-systèmes** [11]classiques sont groupés en chiffrement mono alphabétique et poly alphabétique. Le chiffrement mono alphabétique est très primaire, il s'agit d'une substitution simple. Chaque lettre est remplacée par une autre lettre ou symbole conformément à un certain algorithme. [17] [15]

3.1. Classification :

3.1.1. Chiffrement par décalage:

Un des systèmes les plus anciens et simples est le chiffrement par décalage appel aussi le code de César. Ce code est un des plus anciens. Le principe est de remplacer dans un message une ou plusieurs entités (ex : lettres) par une ou plusieurs autres entités. [17] [15]

[10] **ISO** : International Organisation for Standardisation). Organisation internationale de standardisation regroupant les organismes similaires de 89 nations. L'ISO se charge des standards qui régissent l'Internet actuellement.

[11] **Un crypto-système** est un terme utilisé en cryptographie pour désigner un ensemble composé d'algorithmes cryptographiques et de tous les textes en clair, textes chiffrés et clés possibles.

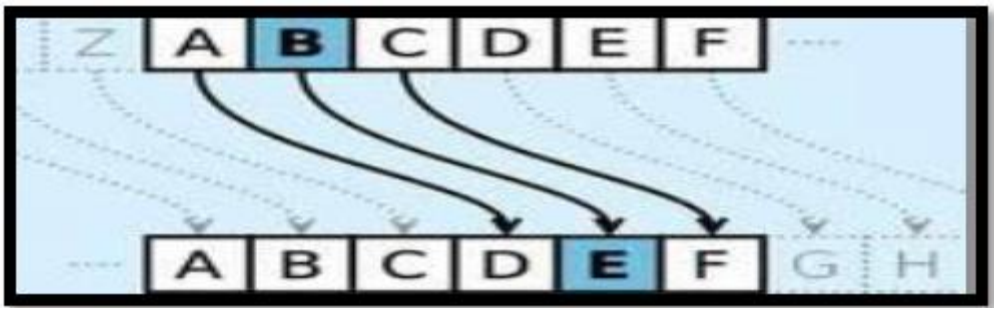


Figure 19 : Principe de code de César [17]

3.1.2. Chiffrement par substitution :

Il s'agit d'une méthode plus générale qui englobe le chiffrement par décalage. En effet, à chaque lettre de l'alphabet on fait correspondre une autre, c'est-à-dire que l'on effectue une permutation de l'ensemble des lettres.

Pour la première lettre a, on a 26 possibilité de substitutions. Pour la lettre b, on n'en a plus que 25 et ainsi de suite. [17] [15]

3.1.3. Le code de Vigenère :

Le chiffrement de Vigenère est une amélioration décisive du chiffre de César. Il a été élaboré par Blaise de Vigenère (1523-1596), diplomate français du XVIe siècle. [14] [15]

- Substitution poly-alphabétique.
- Basé sur la table de Vigenère.
- La colonne correspondante à la lettre en clair.
- La ligne correspondante à une lettre de la clé.
- la lettre chiffrée est le croisement de la ligne et de la colonne.
- La clé est répétée boucle autant que nécessaire.

La Figure 20 explique le principe du fonctionnement du code Vigenére.

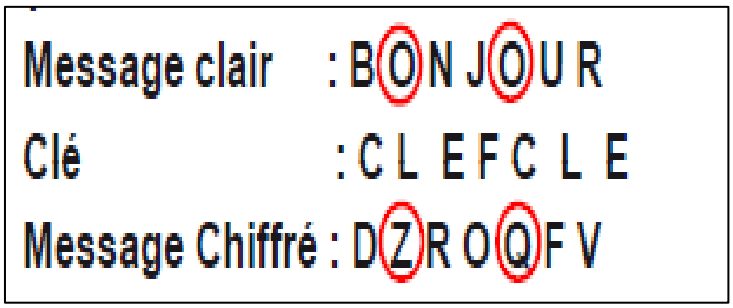


Figure 20 : Exemple sur le code Vigenére [15]

3.1.4. Chiffrement de Vernam :

Une réalisation célèbre de la confidentialité parfaite est le chiffrement de Vernam (voir Figure 21), également connu sous les noms masque jetable ou one-time pad. Il fut inventé par Gilbert Vernam en 1917 pour chiffrer et déchiffrer des messages télégraphiques. [18]

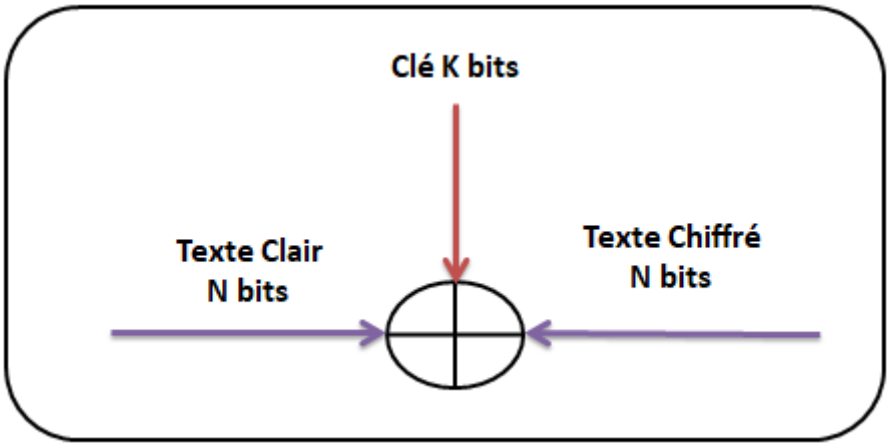


Figure 21 : Chiffrement de Vernam [18]

4. Cryptographie moderne :

La cryptologie moderne a pour objet l'étude des méthodes qui permettent d'assurer les services d'intégrité, d'authenticité et de confidentialité dans les systèmes d'information et de communication. La cryptologie se partage en deux sous-disciplines :

- La cryptographie qui propose des méthodes pour assurer ces services.
- La cryptanalyse qui recherche des failles dans les mécanismes proposés.

Pour assurer les objectifs de la cryptographie moderne nous pouvons utiliser des algorithmes basés sur des clés. Ces algorithmes sont définis par plusieurs types de cryptographie moderne, on distingue deux approches :

- La cryptographie symétrique.
- La cryptographie asymétrique.

4.1. La cryptographie Asymétrique :

Le chiffrement asymétrique (ou chiffrement à clés publiques) consiste à utiliser une clé publique pour le chiffrement et une clé privée pour le déchiffrement. [15] [17]

4.1.1. Principe :

Pour mieux comprendre le principe de la cryptographie asymétrique (voire la Figure 22).

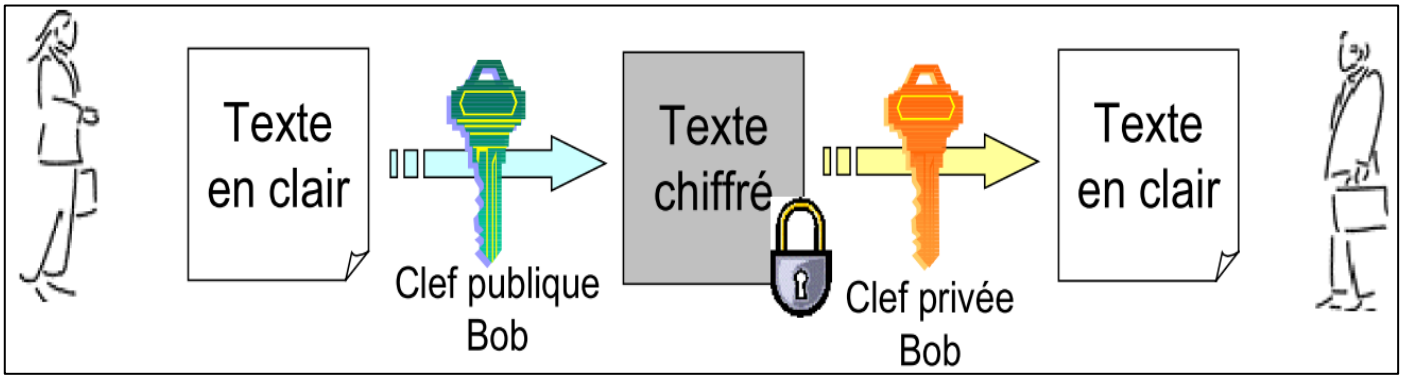


Figure 22 : Principe de chiffrement Asymétrique [17]

- Pas nécessaire d'échangé la clé secrète entre les deux interlocuteurs, seule la clé publique est partagé au travers d'un canal **non sécurisé**[12].
- Une paire de clés : clé publique connue de tous et clé privée connue que de son propriétaire: lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire.
- Chiffrement requiert beaucoup d'opérations et n'est pas recommandé pour de grande quantité d'informations.
- Algorithme de chiffrement **RSA**[13].

4.1.2. Les avantages du cryptage asymétrique :

- L'élimination de la problématique de la distribution de la clé privée.
- La possibilité d'utiliser la signature électronique.
- L'impossibilité de décrypter le message dans le cas de son interception par une personne non autorisé.
- Une paire de clés (publique/secrète) peut être utilisée plus longtemps qu'une clé symétrie.

4.1.3. Les inconvénients du cryptage asymétrique :

- Le temps d'exécution : plus lent que le cryptage symétrique.
- Le danger des attaques par substitution des clés (d'où la nécessité de valider les émetteurs des clés).
- Taille des clés, plus grand que celle des systèmes symétriques (>512 bits).

4.2. La cryptographie symétrique :

Le chiffrement symétrique (aussi appelé chiffrement à clé privée ou à clé secrète), ce type se base sur l'utilisation d'une clé pour crypter et décrypter les messages. La sécurité de cette solution repose sur le fait que la clé est connue uniquement par l'émetteur et le récepteur du message.

[12] **Un canal non sécurisé** est non crypté et peut être sujet à l'écoute clandestine. Des communications sécurisées sont possibles sur un canal non sécurisé si à communiquer le contenu est crypté avant la transmission.

[13] **RSA** : Le chiffrement RSA est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman.

4.2.1. Principe :

Pour mieux comprendre le principe de la cryptographie asymétrique (voire la Figure 23).

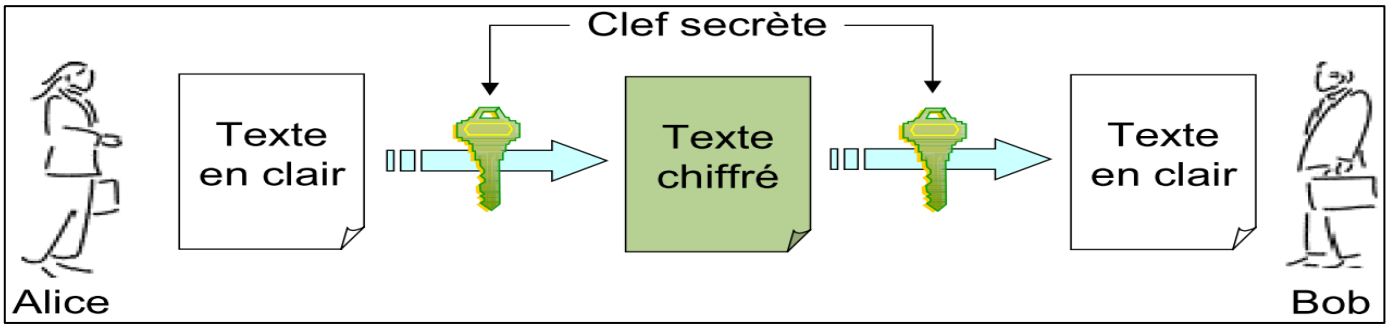


Figure 23 : Principe de chiffrement symétrique [17]

- L'échangé la clé secrète entre les deux interlocuteurs doit s'effectuer à travers un canal sécurisé ou sécuritaire
- Le chiffrement symétrique consiste à appliquer une opération (algorithme) sur les données à chiffrer à l'aide de la clé privée, afin de les rendre inintelligibles.
- On donne le message à quelqu'un et on lui fournit la clé privée, pour qu'il puisse déchiffrer le message.
- Le cryptage symétrique fonctionne selon deux procédés catégories : par Bloc/ par Flots.

4.2.2. Le cryptage par flot (Stream Cipher) :

Les algorithmes de chiffrement de flux peuvent être définis comme étant des algorithmes de chiffrement par blocs, où le bloc a une dimension unitaire (1 bit, 1 octet, etc.) ou relativement petite. Leurs avantages principaux viennent du fait que la transformation (méthode de chiffrement) peut être changée à chaque symbole du texte clair et du fait qu'ils soient extrêmement rapides.

Quelques algorithmes de cryptographie symétrique par flot:

- A5: utilisé dans les téléphones mobiles de type GSM pour chiffrer la communication par radio entre le mobile et l'antenne-relais la plus proche.
- RC4, le plus répandu, conçu en 1987 par Ronald Rivest l'un des inventeurs du RSA, pour les Laboratoires RSA, utilisé notamment par le protocole WEP, un algorithme récent de Eli Biham – E0 utilisé par le protocole Bluetooth. [15]

4.2.3. Le cryptage par bloc (Block Cipher) :

C'est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique. Le principe de cryptage par Blocs est-il consiste de deviser le message en bloc de bits de longueur fixe (souvent une puissance de deux comprise entre 32 et 512 bits) (voir la Figure 24). Les blocs sont ensuite chiffré les uns après les autres. [15]

Deux algorithmes très connus DES et AES.

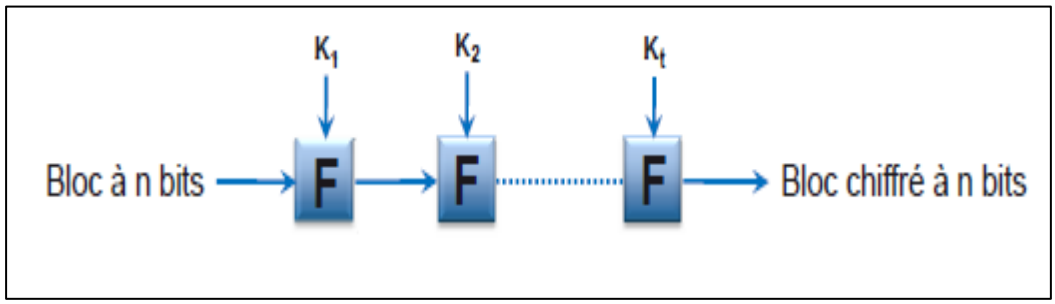


Figure 24 : Principe de chiffrement par Bloc [15]

- Le chiffrement par bloc utilise quatre modes opératoires :
  - Electronic Code Book(ECB)
  - Cipher Block Chaining (CBC).
  - Output Feedback (OFB).
  - Cipher Feedback(CFB).

Electronic Code Book(ECB) : Dictionnaire de codes Un message réel en générale composé de nombreux blocs. La façon la plus immédiate pour chiffrer un tel message est de chiffrer successivement chaque bloc, avec la même clé. Comme montrer dans la Figure 25.

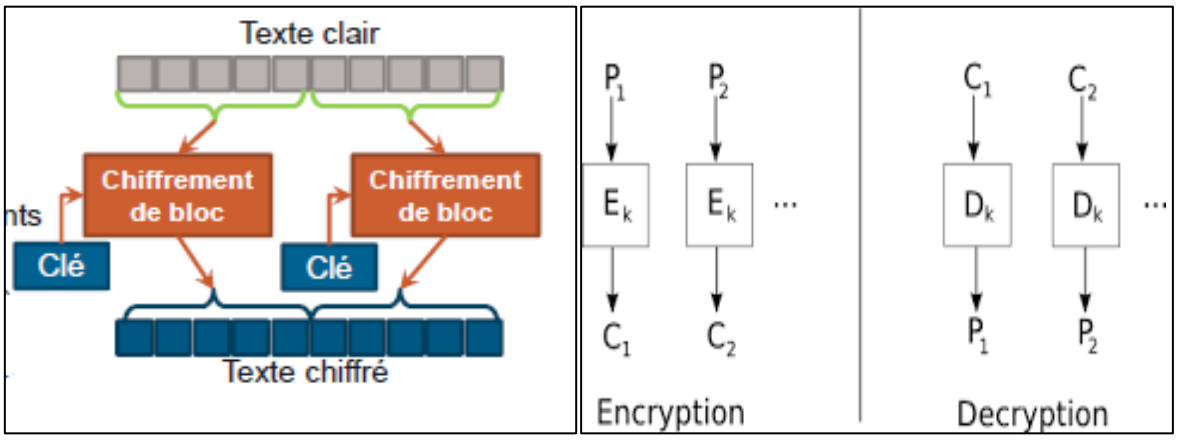


Figure 25 : Le chiffrement par Bloc mode ECB [15]

Cipher Block Chaining (CBC) : L'enchaînement des blocs consiste, avant le chiffrement d'un bloc, à le masquer par le résultat du chiffrement du bloc précédent au moyen de l'opération XOR. Le premier bloc clair est lui aussi masqué, par une valeur habituellement notée IV (Initial Value) et de préférence variable (la date et l'heure peuvent faire une bonne (IV) pour que les chiffrements successifs du même message soient différents. La valeur initiale IV n'a pas besoin d'être secrète, et elle est en générale transmise en clair avant le message chiffré. Noter que si le destinataire reçoit un bloc chiffré avec des bits erronés, cela affecte le déchiffrement de ce bloc et du suivant mais pas des autres (Voir la Figure 26). [15]

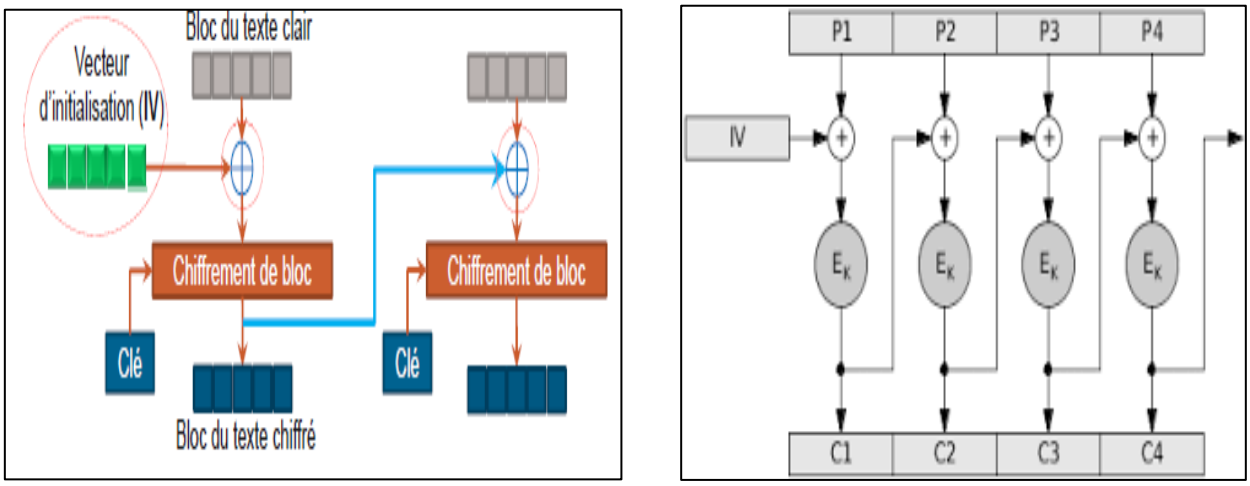


Figure 26 : Le Chiffrement par Bloc mode CBC [15]

Figure 27 : Le chiffrement par ECB et CBC [15]

4.2.4. Les avantages du cryptage symétrique :

- ✓ La rapidité d'exécution (jusqu'à 100 fois plus rapide que les solutions asymétriques)
- ✓ .La simplicité d'implémentation (gestion d'une seule clé).
- ✓ Clés relativement courtes (entre 64-128 bits).

4.2.5. Les inconvénients du cryptage symétrique :

- Gestion des clés difficiles (nombreuses clés).
- Point faible = l'échange de la clé secrète.
- Dans un réseau de N entités susceptibles de communiquer secrètement il faut distribuer N*(N-1)/2 clés.

III. Conclusion :

Dans ce chapitre on a vu que la compression s'impose comme une étape incontournable pour optimiser l'utilisation des grands volumes d'informations dans les réseaux informatiques. L'objectif principal de la compression d'image est de réduire la quantité d'information nécessaire à une représentation visuelle fidèle à l'image originale. En générale on différencie les méthodes de compression selon la perte d'informations. Les méthodes réversibles, utilisent uniquement le principe de la réduction de la redondance et n'engendrent pas de perte. Celles irréversibles, définissent une représentation approximative de l'information. La compression fractale d'image est un outil assez récent pour le codage des images. Elle est basée sur l'auto similarité locale et la génération des copies de blocs sur la base des transformations mathématiques. La technique paraît intéressante dans la théorie et l'application mais l'inconvénient majeur est dans l'usage du temps dû à la haute exigence de ressources quand on chiffre de grandes masses de données.

D'autre part, on a survolé la cryptographie, ses différents types et les propriétés de chaque type, tout en mettant l'accent sur la cryptographie symétrique précisément le Block Cipher.

La combinaison de ces deux techniques, à savoir la compression et le cryptage, et le choix de l'algorithme de compression et de cryptage pour en faire un système hybride de crypto-compression, tout ça sera détaillé dans le deuxième chapitre.

DEUXIEME CHAPITRE

CRYPTO-COMPRESSION

1. Introduction :

La réduction de la taille des bits présentés sur l'image est devenue de plus en plus important dans le stockage, la cryptographie aussi est devenue un critère très important dans la transmission sécurisé des images. Pour assure ces deux critères on a besoin des techniques qui combinent les deux technologies la compression et le cryptage des images qui ont signifié aux techniques de Crypto-compression qui ont pris de l'essor ces dernières années. Le concept vise à combiner à la fois les techniques de cryptage et de compression de manière jointe. L'objectif est de procurer un volume de données de taille réduite avec une confidentialité robuste. Le travail présenté dans ce chapitre porte sur une approche de crypto-compression à base de Block Cipher et d'un algorithme de compression fractale. Nous allons commencer tout d'abord par présenter l'algorithme générique de compression et décompression fractale. Ensuite nous présentons une approche de compression par la méthode de Jacquin et nous allons présenter l'approche de cryptage à base de Block Cipher en utilisant l'algorithme AES.

2. Compression fractale par blocs :

Sur la base des **IFS**[14], Jacquin a proposé en 1989 une approche fractale ne nécessitant pas d'intervention humaine et permettant de coder une image naturelle. La méthode était basé sur une série de transformation affines contractantes et ''locales'' définissant un opérateur contractant noté W. Elle a l'avantage d'être automatique mais ne résout pas directement le problème inverse des IFS puisque l'image n'est pas vue comme une union de transformation d'elle-même mais comme une union de sous-partie transformées d'elle-même. [19]

2.1. Transformation contractive :

Une transformation est dite contractive si pour tous deux points P1 et P2 la distance :

$$\boldsymbol{d(w(P1), w(P2)) < S * d(P1, P2)\ tel\ que\ 0 < S < 1}$$ **Équation 7**

Cette transformation contractive appliquée à deux points va les rapprocher. Cette définition est absolument général, elle s'applique à tout espace métrique (espace sur lequel on peut définir une distance d (P1, P2)), si les points ont pour coordonnés P1=(x1, y1) et P2=(x2, y2) alors :

$$\boldsymbol{d(P1, P2) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}}$$ **[13] Équation 8**

[14] **Les IFS (**Iterated Function System) ont été introduits en 1985 par M.F Barnsley et S.Denko.

2.2. La définition d'un IFS :

Un IFS consiste en une collection de transformations contractantes{ wi: $R^2 \rightarrow R^2 / i = 1, \dots, n$}, Cette collection de transformations définit une fonction :

$$\boldsymbol{W}(.) = \cup_{i=1}^{N} \boldsymbol{W}(.)$$ **[13]**Équation 9

La fonction W n'est pas appliquée au plan, mais à des ensembles de points du plan, Hutchinson a démontré un fait important pour cette fonction [5] est que, quand les wi sont contractantes dans le plan, alors W est contractant dans l'espace de l'ensemble de points (du plan). [13]

2.3. Théorème du point fixe :

Ce théorème formalise une constatation intuitive, si une fonction est contractante, alors lorsqu'on l'applique de façon récurrente à partir d'un point initial, on converge vers un unique point fixe. [13]

2.4. Les IFS comme outils de compression :

M. Barnsley suggéra que de stocker les images comme code IFS (collection de transformations affines contractantes) puisse amener la compression de l'image. Par exemple la feuille de fougère est générée par seulement quatre transformations affines chaque transformation affine est définie par 6 nombres (a, b, c, d, e, f) qui ne nécessitent pas beaucoup de mémoire pour être stockés, elles peuvent être stockées dans 4(transformations)×6(nombres)×32 bit (pour chaque nombre)=768 bits alors que l'image sous forme de pixels requiert beaucoup plus de mémoire au moins 65 536 bits.

On voit que le stockage des images sous formes de collection de transformations contractantes amène directement la compression puisque le stockage d'une image sous formes de pixels nécessite beaucoup de place mémoire que de stocker l'image sous forme de collection de transformations contractantes. [13]

2.5. Codage :

La compression d'une image par fractales repose sur une transformation fractale qui consiste à transformer l'image à l'aide d'un opérateur finalement contractant W, de manière à ce que son aspect visuel reste quasiment inchangé. Pour cela, l'opérateur W est constitué de N sous-transformation élémentaires w_n, chacune opérant sur un bloc de l'image (Figure 28). [19]

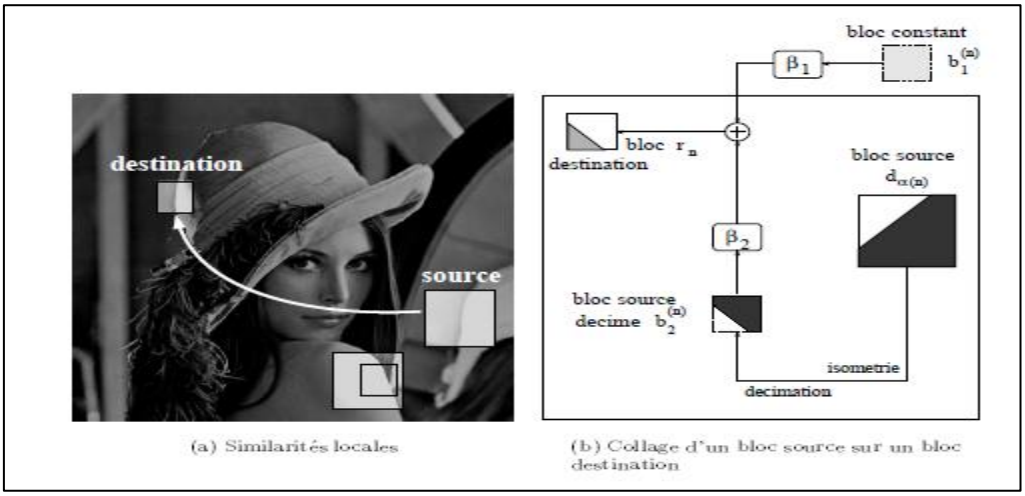


Figure 28 : Principe de codage par fractales [19]

L'image A est partitionné en N blocs r_n appelés blocs destination. Chaque bloc destination est ensuite mis en correspondance avec un autre bloc $d_{\alpha(n)}$ au moyen d'une transformation locale w_n. Le bloc $w(d_{\alpha(n)})$ ressemble au bloc r_n au sens d'une mesure d'erreur (quadratique) sur les niveaux de gris. Le bloc $d_{\alpha(n)}$, appelé bloc source est recherché au travers d'une librairie composée de Q blocs ne forment pas nécessairement une partition de l'image mais sont représentatifs de toute l'image. La transformation W de l'image A est ainsi formulée de façon suivante :

$$\boldsymbol{W(A) = \cup_{n=1}^{N} W_n(d\alpha_n) = \cup_{n=1}^{N} \hat{r}_n \simeq \cup_{n=1}^{N} r_n} = A \quad \text{Équation 10}$$

Où $\hat{r}_n$ est l'approximation du bloc destination r_n obtenue en transformation le bloc source $d_{\alpha(n)}$ par w_n . La transformation w_n a pour effet de coller le bloc source $d_{\alpha(n)}$ sur le bloc destination r_n, en déformant son support (isométrie, translation) es en modifiant sa fonction de luminance. L'approximation du bloc r_n, noté $\hat{r}_n$, est donnée par une combinaison linéaire de deux blocs, parmi lesquels le bloc $d_{\alpha(n)}$ est extrait de l'image elle-même.

On comprend aisément que la difficulté réside dans la contraction, recherche des similarités et codage des transformations w_n, sur la base de partitions de l'image. [19]

3. Algorithme générique de compression et décompression :

Ces deux figures représentent le principe de l'algorithme générique de la compression/décompression fractale. Ces algorithmes sont génériques. Les trois méthodes que nous avons vu dans le chapitre précédente dans ce document, ajoutent leurs propres fonctionnalités afin d'améliorer le taux de compression. [12]

Algorithm 1 Algorithme générique de la compression fractale.

```
1: Fonction COMPRESSION
2:     Créer un pavage de figures Sources
3:     Créer un pavage de figures Destinations
4:     Pour toutes les figures Destinations Faire
5:         Pour toutes les figures Sources  Faire
6:             Pour toutes les transformations définies Faire
7:                 Appliquer la transformation à la figure Source
8:                 Ajuster la moyenne des couleurs des pixels
9:                 Appliquer la réduction de la figure Source vers la figure Destination
10:                Calculer l'erreur entre le résultat et la figure de destination
11:                Si l'erreur est minimale pour la figure de destination Alors
12:                    Sauver les modifications effectuées
13:                FinSi
14:            Fin Pour
15:            Ecrire dans le fichier de sortie les valeurs sauvées
16:        Fin Pour
17:    Fin Pour
18: Fin Fonction
```

Figure 29 : Algorithme générique de compression [12]

Algorithm 2 Algorithme générique de la décompression fractale.

```
1: Fonction DÉCOMPRESSION
2:     Créer une image
3:     Créer un pavage de figures Sources
4:     Créer un pavage de figures Destinations
5:     Pour toutes les itérations désirées Faire
6:         Pour toutes les figures Destinations Faire
7:             Lire dans le fichier les transformations à appliquer
8:             Lire le numéro de la figure Source à manipuler
9:             Appliquer les transformations à la figure précédemment évoquée
10:            Appliquer la réduction de la figure Source vers la figure destination
11:            Inserer le résultat dans l'image
12:        Fin Pour
13:    Fin Pour
14: Fin Fonction
```

Figure 30 : Algorithme générique de décompression [12]

4. Méthode Jacquin :

La compression fractale c'est une méthode destructive puisque l'ensemble des données de départ ne se retrouve pas dans l'image finale. Il existe plusieurs méthodes (subdivision de triangles, Delaunay etc.) mais la compression par la méthode Jacquin est la plus connue et la plus simple.

4.1. Fonctionnement :

- ✓ [etape0] : On commence par générer sur l'image un pavage de carrés appelés Sources et un pavage de carrés Destinations. Puis, on cherche quelle est la meilleure combinaison Source/Destination pour chaque figure Source. Pour cela l'étape 01.
- ✓ [Étape 1] : On applique différentes transformations (flip et/ou rotation, réduction de l'intensité) sur le carré Source courant (comme le montre la figure 07 Chapitre 1).
- ✓ [Étape 2] : on réduit sa taille pour le rendre de la même hauteur que la figure destination.
- ✓ [étape 3] : on calcule l'erreur entre les deux entités.
    - L'erreur est évaluée grâce à l'´écart type des pixels des 2 régions. Si cette erreur est minimale, on sauvegarde le couple Source/Destination ainsi que les modifications apportées.
    - Pour recréer l'image à partir du fichier final, il suffit de recréer les 2 partitions source et destination, et d'appliquer plusieurs fois les transformations des couples sauvés dans le fichier.
    - Le gène fractal de l'algorithme de Jacquin garantit alors la convergence de l'image vers l'image de d´épart. [12]

4.2. Partitionnements :

Le partitionnement est l'opération qui consiste à segmenter une image en régions. Dans compression par la méthode Jacquin, nous avons besoin de deux partitionnements : Source et destination. Comme nous l'avons vu précédemment, la méthode Jacquin utilise des figures carrées. Voici les pavages réalisés lors de la compression et décompression Jacquin montré dans la Figure 31 : [12]

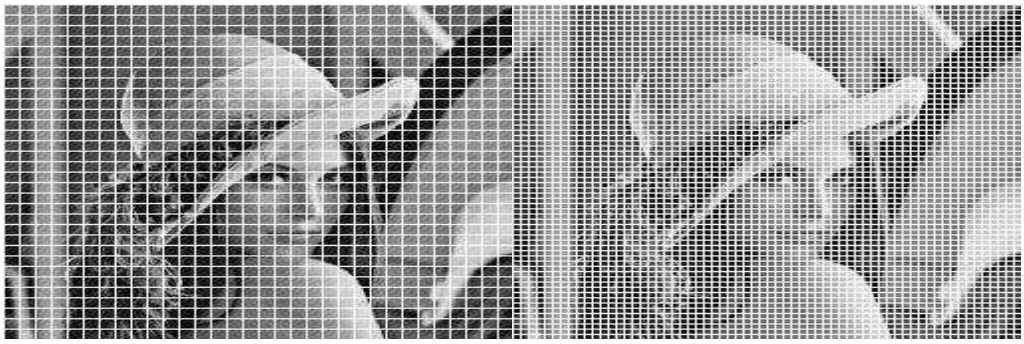


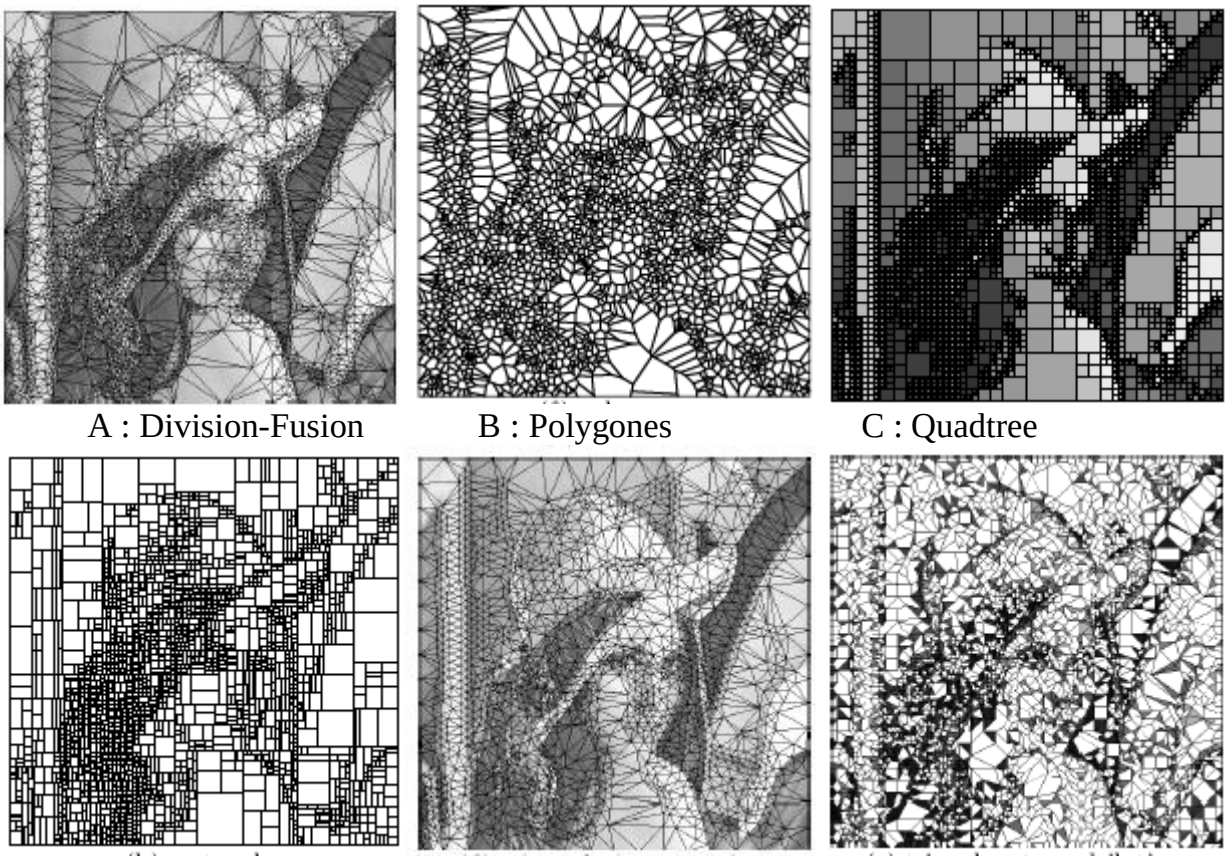
Figure 31 : Pavage Source et Destination, Méthode Jacquin [12]

Un point essentiel dans les partitionnements Source et Destination est que le pavage destination doit être plus petit que le pavage source. En effet, dans le cas contraire, nous serions amenés à faire un agrandissement (et non une réduction) lors de la transposition des figures sources vers les figures destinations. [12]

4.2.1. Partitionnement adaptatifs :

Différents partitionnements de l'image ont été étudiés de façon à minimiser le nombre de transformation locales, et à coder au mieux les similarités inter-blocs. Nous les énumérons ci-après selon une graduation du plus en plus souple :

- Partitionnement Carrés.
- Partitionnement Quadtree.
- Partitionnement Rectangulaire.
- Partitionnement Triangulaire.
- Partitionnement Triangulaire adaptif (Delaunay).
- Partitionnement à base de Quadrilatères. [19]



A : Division-Fusion B : Polygones C : Quadtree

D : Rectangle E : Triangulation Contrainte F : Triangulation quadrilatères

Figure 32 : Différentes Partitionnement allant d'une géométrie rigide (carrés) à une géométrie souple (triangle, Polygones) [19]

5. La Décomposition Quadtree :

Le principal problème est que le codage fractal prend trop de temps. De nombreuses approches pour réduire le temps d'encodage ont une mauvaise affection sur l'image qualité après itération, donc la méthode de codage hybride de combiner le codage fractal et d'autres méthodes de codage deviennent une direction importante des méthodes fractales.

L'approche Quadtree divise une image carrée en quatre blocs carrés de taille égale, puis teste chaque bloc pour voir si répond à un critère d'homogénéité. Si un bloc répond au critère, il n'est plus divisé, et le critère de test est appliqué à ces blocs. Ce processus est répété de manière itérative jusqu'à ce que chaque bloc réponde au critère.

Le résultat peut avoir des blocs de plusieurs tailles différentes. [20]

6. Cryptage a clé privé :

Le chiffrement symétrique ou chiffrement a clés privé, c'est l'un des deux catégories on cryptographie moderne, et qui consiste à utiliser un seul clé pour le chiffrement et le déchiffrement des données, les algorithmes de chiffrement a clé privé les plus reconnues sont l'AES et le DES.

6.1. L'AES (Advanced Encryption Standard) :

- Est un algorithme cryptographique par bloc à clé secrète.
- C'est le successeur du DES (Data Encryption Standard) qui a été implémenté dans un grand nombre de modules cryptographiques à une échelle mondiale depuis son apparition en 1977.
- L'AES conserve toujours le haut niveau de sécurité proposé par le DES
- L'innovation apportée par l'AES est notée au niveau de la clé secrète ainsi que la taille des données traitées en entrée.
- Nous passons d'une clé de chiffrement de 64 bits (8 octets) pour le DES vers une clé de taille double 128 bits (16 octets) pour l'AES.
- La taille des données à crypter est notablement plus grande puisque nous passons de 56 bits vers 128 bits. Ce qui permet d'exploiter l'algorithme dans des applications supportant des fichiers de données de taille grande.
- L'AES utilise un bloc d'une longueur fixée à 128 bits et une clé d'une longueur de 128, 192, et 256 bits.
- Comme tout système cryptographique symétrique, l'AES dispose de deux entrées, à savoir le texte clair à chiffrer (plaintext) ainsi que la clé de cryptage (key) (Figure 33). [21]

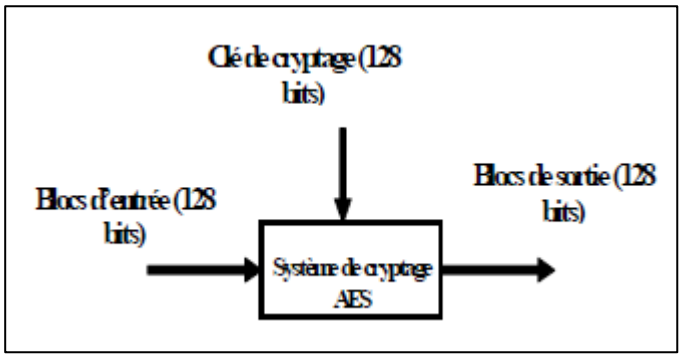


Figure 33 : Système de cryptage AES [21]

6.2. AES : Algorithme :

AES est un algorithme de chiffrement par blocs, les données sont traitées par blocs de 128 bits pour le texte clair et le chiffré. La clef secrète a une longueur de 128 bits, d'où le nom de version: AES 128 (il existe deux autres variantes dont la clef fait respectivement 192 et 256 bits). Nous allons présenter les étapes de l'algorithme dans ce qui suit :

Bloc divisé en octets répartis dans des matrices 4x4 (1 octet = 8 bits) (Figure 34).

Séquence de 4 transformations répétées :

- ✓ 1ère étape : substitution (confusion)
- ✓ 2ème étape : décalage des lignes (diffusion)
- ✓ 3ème étape : brouillage des colonnes (diffusion)
- ✓ 4ème étape : addition des sous-clés.

Les tours :

- ✓ Tour initial: addition XOR des sous-clés aux blocs
- ✓ Tours similaires itérés: les 4 étapes sont répétées.
- ✓ Dernier tour: transformation sans la 3ème étape.

Étapes de chiffrement sont inversées et réordonnées pour produire un algorithme de déchiffrement. [22]

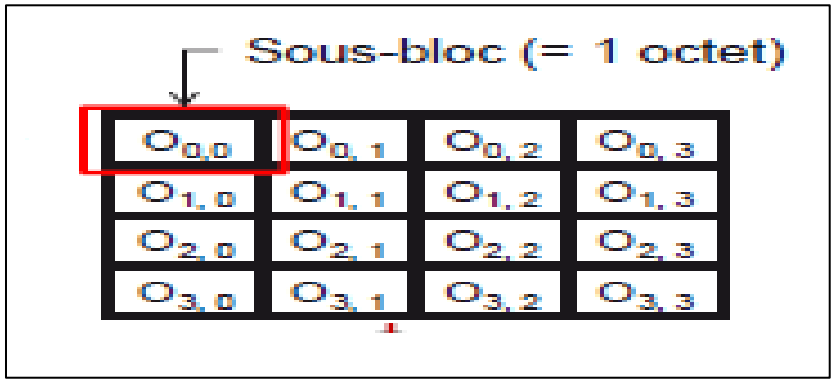


Figure 34 : Le bloc présenté par l'algorithme l'AES [22]

7. L'algorithme utilisé pour la compression :

Nous avons utilisé une approche hybride de compression fractale et qui de base sur la décomposition Quadtree et le codage de Huffman, les étapes de l'algorithme sont les suivantes :

a. Diviser l'image originale en utilisant la décomposition Quadtree du seuil est de 0.2, minimum, la dimension minimale et la dimension maximale sont respectivement 2 et 64.
b. Enregistrer les valeurs des coordonnées x et y, la valeur moyenne et la taille du bloc de Quadtree Décomposition.
c. Enregistrer les informations du codage fractal pour terminer le codage de l'image à l'aide du codage Huffman et calculer le taux de compression.
d. Pour l'image codée, on applique le décodage Huffman pour reconstruire l'image et calculer le PSNR. La Figure 35 montre la technique de compression fractale utilisée.

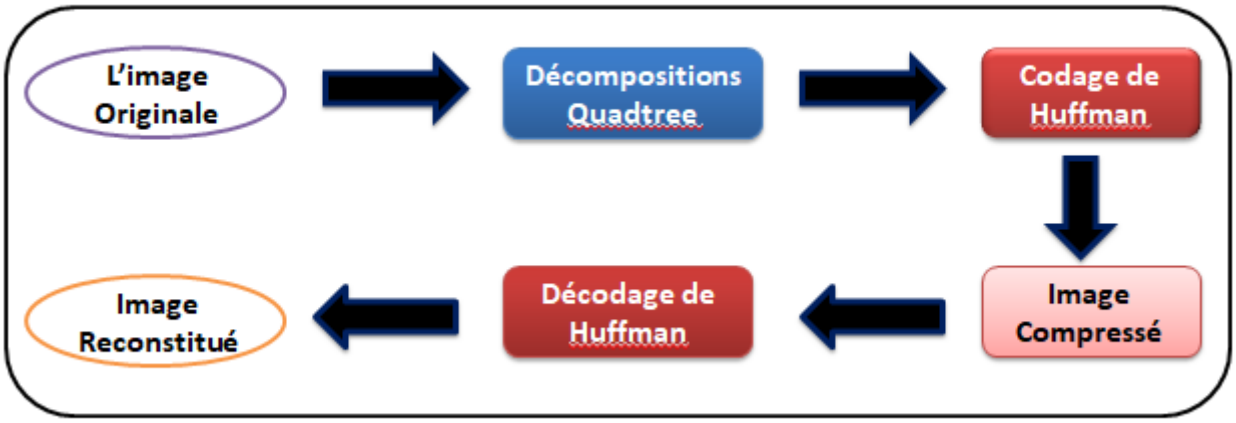


Figure 35: Technique de compression Fractale basée sur la décomposition en Quadtree [20]

7.1. Le codage de Huffman :

L'algorithme de codage Huffman commence par construire une liste de tous les symboles de l'alphabet dans ordre décroissant de leurs probabilités.

Il construit ensuite, de bas en haut, un arbre binaire avec un symbole à chaque feuille. Cela se fait par étapes, où à chaque étape deux symboles avec le plus petit les probabilités sont sélectionnées, ajoutées en haut de l'arborescence partielle, supprimées de la liste et remplacées avec un symbole auxiliaire représentant les deux symboles originaux. Lorsque la liste est réduite à un seul symbole auxiliaire (représentant tout l'alphabet), l'arbre est complet. L'arbre est alors traversé pour déterminer les mots de code des symboles. [20]

7.2. Décodage Huffman :

Avant de commencer la compression d'un fichier de données, l'encodeur doit déterminer les codes. Il fait ça sur la base des probabilités de fréquences d'occurrence des symboles. Les probabilités ou les fréquences doivent être écrites, comme information latérale, sur la sortie, de sorte que tout décodeur Huffman pourra décompresser les données. C'est facile, car les fréquences sont des nombres entiers et les probabilités peuvent être écrites sous forme d'entiers mis à l'échelle.

Il ajoute normalement quelques centaines d'octets à la production. Il est également possible d'écrire les codes de longueur variable eux-mêmes sur la sortie, mais cela peut être maladroit, car les codes ont des tailles différentes. Il est également possible d'écrire l'arbre Huffman sur la sortie, mais cela peut nécessiter plus d'espace que les seules fréquences. Dans tous les cas, le décodeur doit savoir ce qui se trouve au début du fichier compressé, le lire et construire le Huffman arbre pour l'alphabet. Ce n'est qu'alors qu'il peut lire et décoder le reste de son entrée.

L'algorithme pour le décodage est simple. Commencez à la racine et lisez le premier bit de l'entrée (le fichier compressé). Si c'est zéro, suivez le bord inférieur de l'arbre; s'il en est un, suivez le bord supérieur. Lisez le bit suivant et déplacez un autre bord vers les feuilles de l'arbre. Lorsque le décodeur arrive sur une feuille, il y trouve le symbole original, non compressé, et ce code est émis par le décodeur. Le processus commence à nouveau à la racine avec le bit suivant. [20]

8. L'AES algorithme de cryptage d'image :

L'algorithme de cryptage AES consiste à utiliser quatre étapes importantes :

1ère phase ("Byte Sub") :

- ✓ Substitution de chaque sous-bloc avec des fonctions non linéaires ("S-boxes") : transformation non linéaire.

2ème étape ("Shift Row") :

- ✓ Décalage des lignes de chaque sous-bloc: les 3 dernières lignes sont décalées cycliquement vers la gauche avec des décalages différents : la deuxième ligne est décalée de trois octets, la troisième ligne de deux octets et la première ligne d'un octet.

3ème étape (“Mix Column”) :

- ✓ Brouillage des colonnes : multiplication d'une matrice aux sous-blocs : Chaque colonne du bloc est transformée linéairement par la multiplication d'une matrice dont les coefficients sont constants.
- ✓ Transformation linéaire.

4ème étape (“Add Round Key”):

- ✓ Addition des sous-clés aux sous-blocs avec XOR : Une clé, différente à chaque tour et de même longueur que le bloc, est ajoutée bit à bit au bloc par un XOR. Cette clé de tour est dérivée de la clé de chiffrement par un sous-algorithme, nommé algorithme de cadencement de clé (Figure 36). [21]

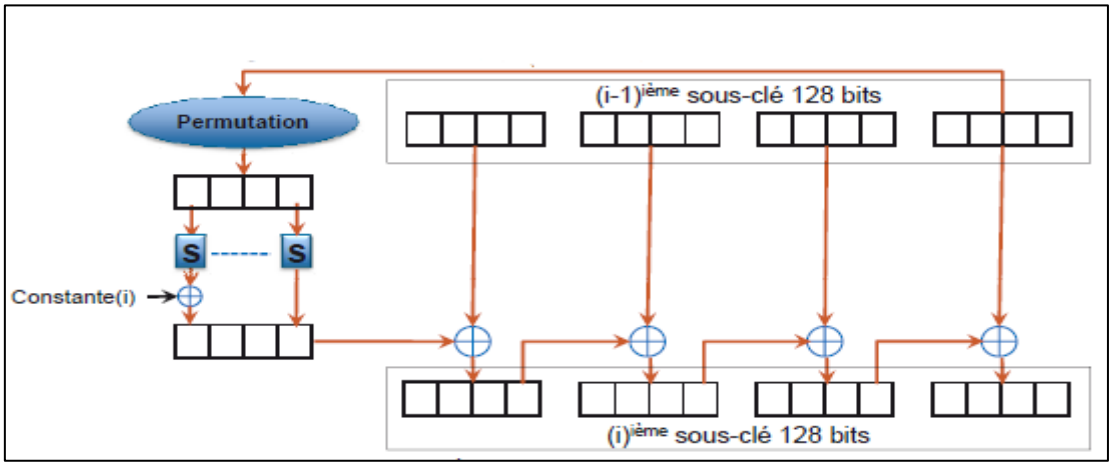


Figure 36 : Algorithme de génération de clés. [23]

8.1. L’algorithme de cryptage :

L'implémentation de l'algorithme de chiffrement et de déchiffrement AES-128 à l'aide du logiciel MATLAB est effectuée. Dans laquelle l'entrée est une image et la clé au format hexadécimal et la sortie est la même que celle de l'image d'entrée. Pour le processus de cryptage, divisez d'abord l'image et rendez-la en 4 * 4 octets, c'est-à-dire au format matriciel. Calculez le nombre de tours en fonction de la taille de la clé et développez la clé à l'aide de notre clé. Et il y a (n-1) tours effectués qui sont des octets de substitution, décalent les lignes, mélangent les colonnes et ajoutent une clé. Le dernier tour «n» ne comprend pas de colonne de mélange dans l'itération. La Figure 38 montre le flux de l'algorithme. [24]

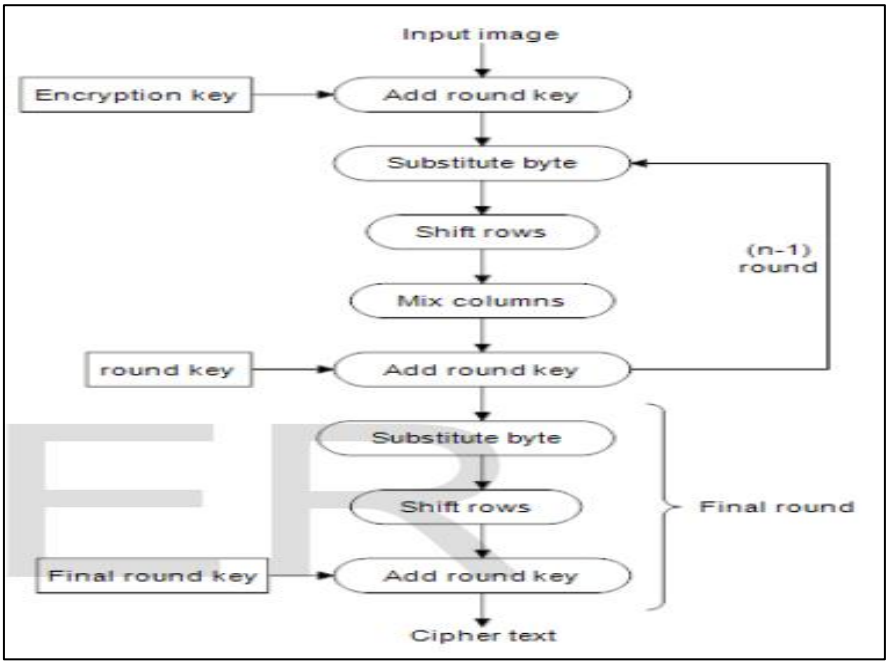


Figure 37 :L'algorithme proposé pour le Cryptage AES [24]

8.2. L'algorithme de décryptage :

Le processus de décryptage AES est le processus inverse de celui du processus de cryptage. La figure 37 ci-dessus montre le flux de l'algorithme de décryptage AES. Composé de texte chiffré comme entrée, la clé est la même pour le processus de déchiffrement que pour le chiffrement. En cas de déchiffrement, l'octet de substitution inverse, les lignes de décalage inverses et les colonnes de mélange inverses doivent être implémentés. Alors que la touche d'ajout de rond reste la même. [24]

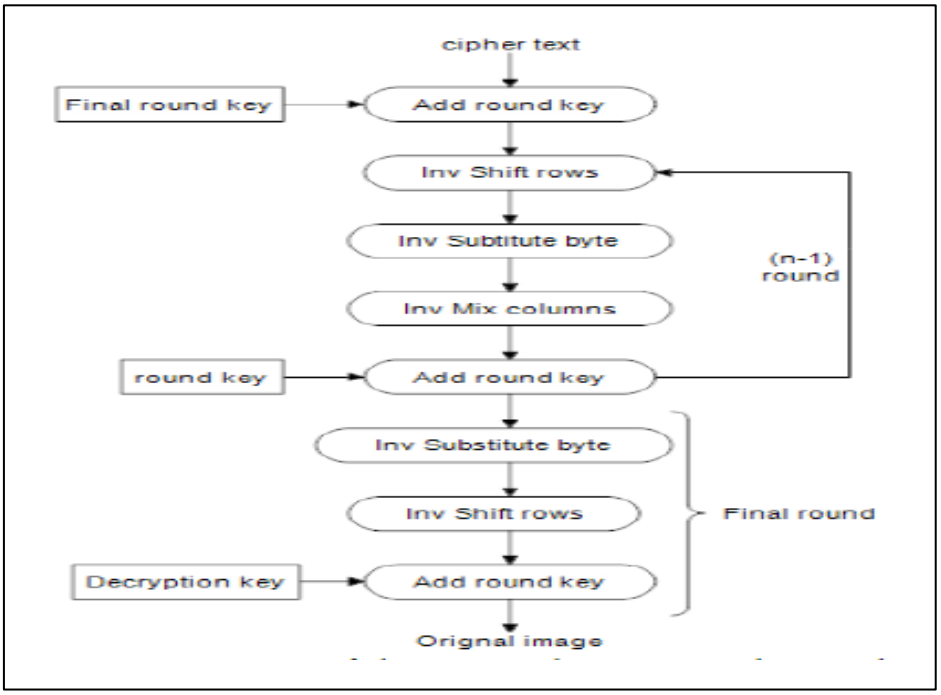


Figure 38 : L'algorithme proposé pour le Décryptage AES [24]

9. Système de Crypto-Compression proposé :

L'approche introduite est basée sur deux algorithmes: un pour compresser et crypter l'image et l'autre pour reconstruire l'image. Ces deux algorithmes sont présentés comme montre la Figure 39.

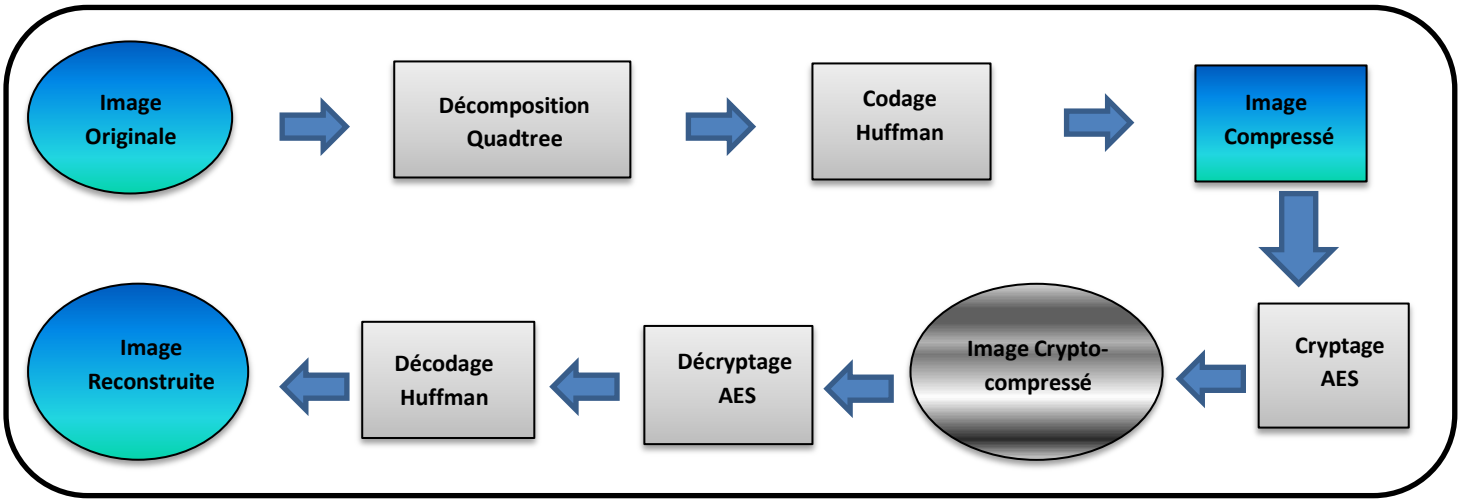


Figure 39: Système de Crypto-Compression proposé

10. Conclusion :

Dans ce chapitre, nous avons commencé par une présentation détaillée sur la compression fractale ainsi qu'une étude de compression fractale par bloc, le théorème du point fixe, les IFS comme outils de compression fractale, comment ça marche le codage fractale et enfin, la décomposition Quadtree. Ensuite, nous avons présenté l'approche de cryptage AES.

Enfin on a terminé avec la présentation du système de crypto-compression proposé. L'implémentation de l'algorithme du système proposé sera présentée dans le chapitre suivant avec comparaison des différents résultats.

TROISIEME CHAPITRE

REALISATION ET IMPLEMENTATION

1. Introduction :

Notre but consiste à la réduction de la taille des bits présentés sur l'image (la taille de l'image), au même temps assurer leur transmission sécurisé sur le réseau, pour cela le travail de ce mémoire, il consiste à la réalisation d'un système hybride de crypto-compression, utilisant une méthode de compression fractal et un chiffrement par bloc, ces deux techniques sont appliqués l'un après l'autre dans un seul système, sur des images réelles.

Plusieurs résultats sont obtenues grâce à l'exécution de notre système sur différentes images, ensuite on a changé la façon de l'application de ces deux approches sur les mêmes images pour procéder une étude comparative avec notre technique utilisé dans notre système et d'autre technique ; se basant sur des paramètres calculés : le MSE, PSNR, CR, Taux de compression, temps de compression et cryptage, temps de décompression et décryptage.

Pour la réalisation de l'interface de travaille, 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.

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.

2. Environnement de travail :

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 10 dont les performances sont les suivantes :

- Processeur : Intel core (TM) i3-5010u CPU 2.10Ghz.
- Fréquence de 2.10 GHz.
- Mémoire installé (RAM) : RAM de 4 Go DDR3.
- Type de système : système d'exploitation 64 bits, processeur x64.
- Disque 500 Go HGST HTS545050A7E680.

2.2. Langage de programmation :

2.2.1. 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**[15] 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 coeur 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.

2.2.2. 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 guidé en cliquant sur les boutons selon notre choix.

2.2.3. 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 40 illustre l'organigramme du logiciel élaboré.

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 décomposition Quadtree et le codage Huffman, et pour le chiffrement nous appliquerons l'algorithme d'AES.

[15] **HMI :** Les interfaces homme-machine ou IHM sont les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine.

En deuxième partie nous faisant le déchiffrement et la décompression, pour bien valider la qualité de l'image reconstruite on calcul les paramètres de la distorsion à savoir le PSNR et MSE.

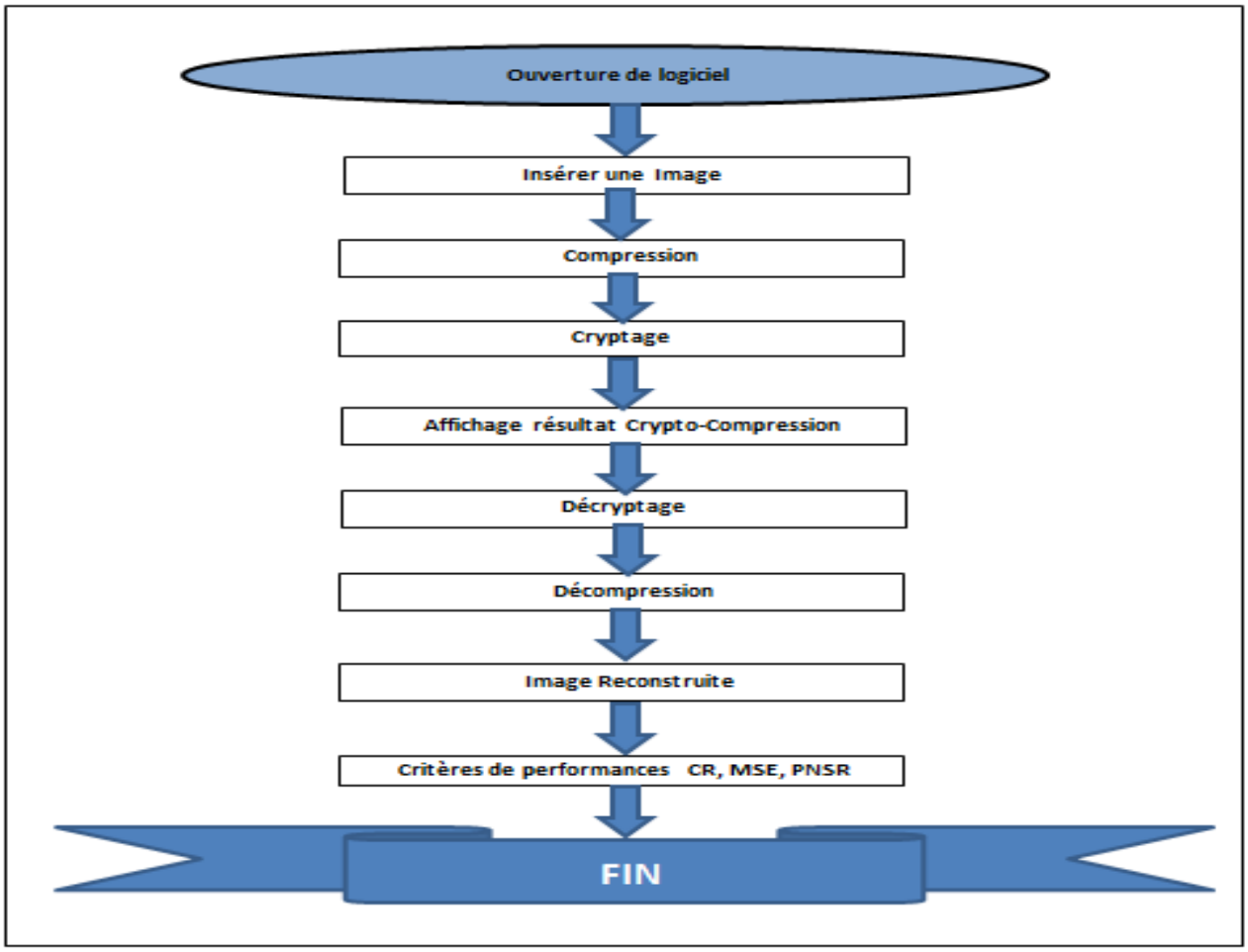


Figure 40 : Organigramme du logiciel élaboré

3. Principe de fonctionnement de l'application :

La figure ci-dessous présente l'interface de l'application qui s'intitule « Image crypto-compression System ».

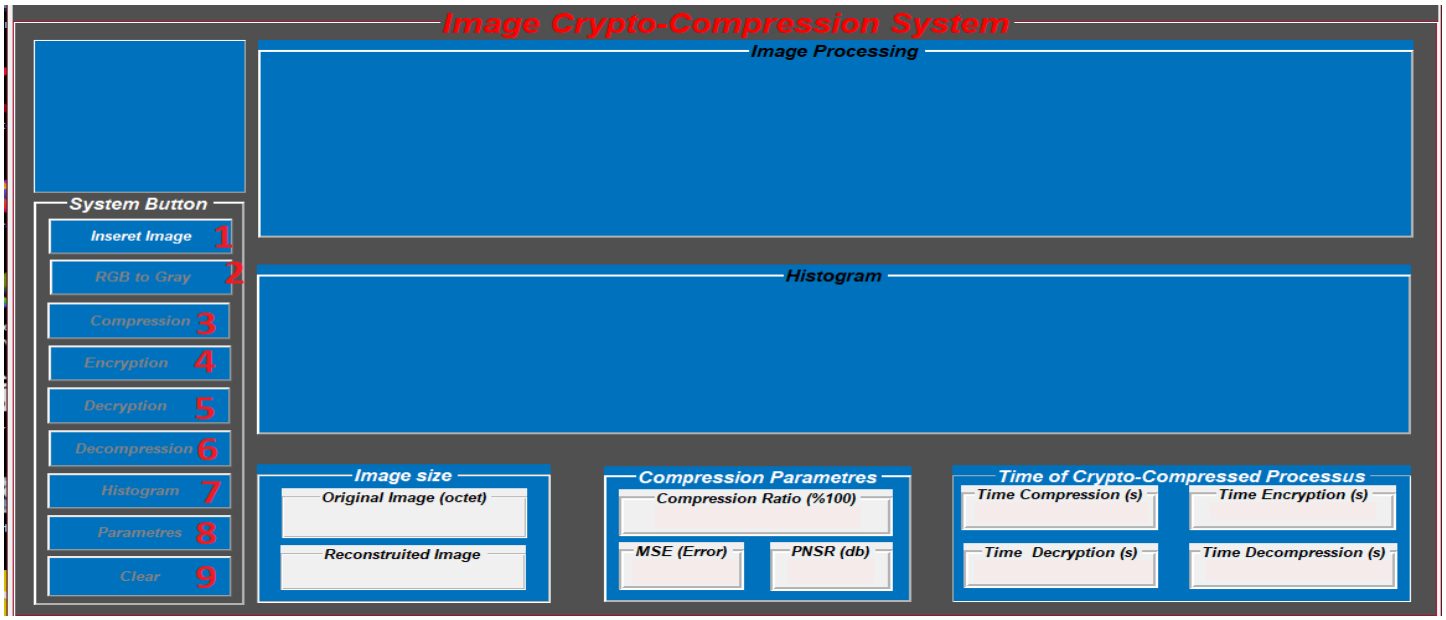


Figure 41 : Interface du système

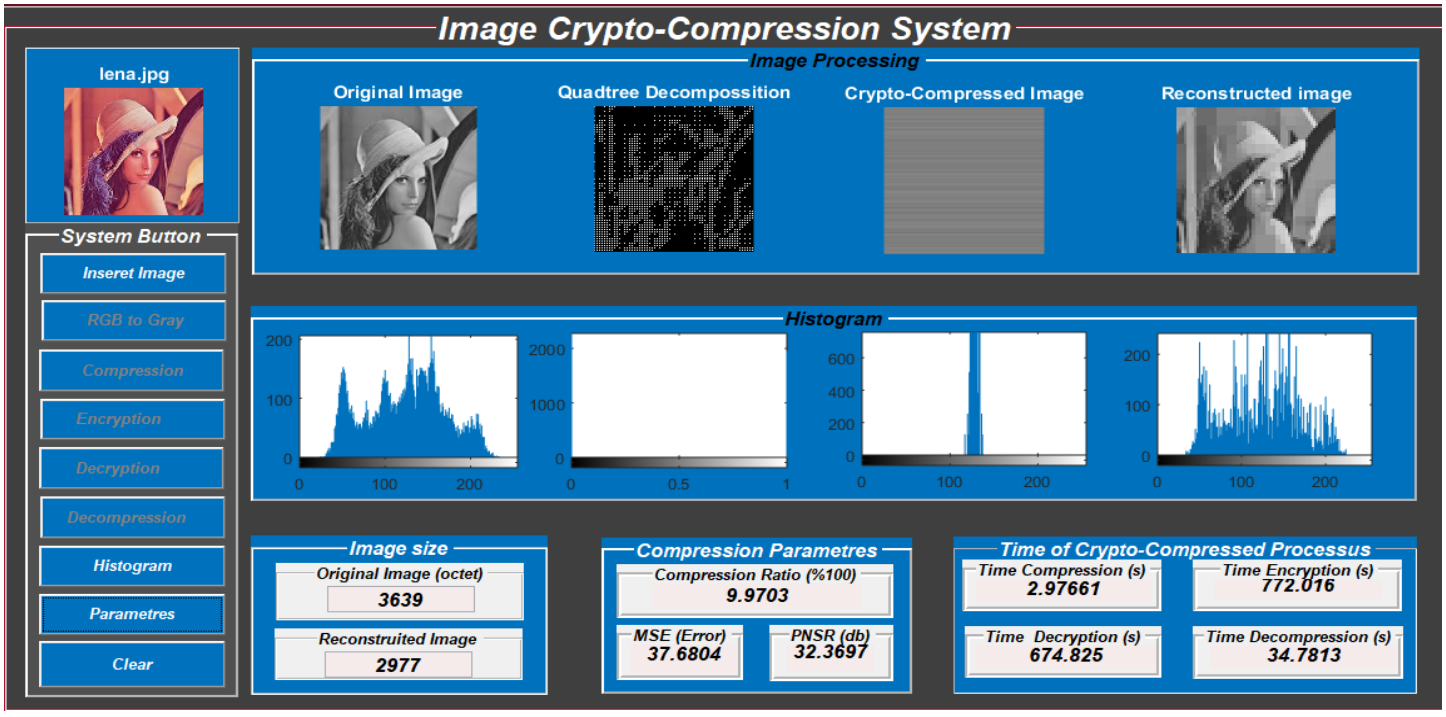


Figure 42 : Interface de l'application après exécution

3.1. Description des modules de système :

Le fonctionnement de notre système de crypto-compression d'image basé sur des modules qui sont exécutés respectivement l'un après l'autre, nous allons présenter la description de ces modules :

1. Module 'Lecture' : Permet de charger une image à partir de n'importe qu'elle endroit du PC.
2. Module de 'Transformation' : Permet de changer la couleur de l'image de RGB au gris, d'autre façon l'image **RGB**[16] est une matrice compose de trois matrice uint8.
3. Module 'compression' : Ce module est le plus important dans notre système ; il contient l'algorithme hybride «décomposition Quadtree, et le codage de Huffman», il permet de faire la compression fractale de l'image originale, l'algorithme est réalisé on Matlab pour coder et décoder les images « La valeur de Threshold est 0.2, maximum dimension est 64 et minimum dimension est 2 pour la décomposition Quadtree.
4. Module 'chiffrement' : Ce module permet de crypter une image par le système de chiffrement AES, l'entré de ce module est le résultat obtenue par le module de compression ''matrice uint8 après la transformation d'un vecteur double qui contient des informations compressé''. La clé utilisé est un hexadécimale clé à 16 bits, c'est la même clé pour le module de décryptage.
5. Module 'déchiffrement' : Ce module permet de décrypter l'image crypté. L'entrée de ce module est la sortie du module précèdent est une image crypter « matrice uint8 », la sortie de ce module est une image « matrice uint8 » décrypter avec la même clé utilisé dans le module de décryptage, on le transforme cette dernier on vecteur double pour la préparer au module de décompression.
6. Module 'décompression' : Il permet de décompresser l'image et afficher l'image reconstruite. Ce module est la dernière étape dans notre système de Crypto-Compression, d'autre façon c'est la sortie de notre système qui donne l'image reconstruite.
7. Module de ''calcule l'histogramme'' : Ce module permet de calculer l'histogramme de chaque étape du processus de crypto-compression.
8. 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 signal a bruit (Peak Signal to Noise Ratio, PSNR).

[16] Rouge, vert, bleu, abrégé en RVB ou en RGB est un système de codage informatique des couleurs, le plus proche du matériel

9. Module de ‘’Clear’’ ou ‘’Reset’’ : Ce module permet de supprimer toutes les champs de l’interface utilisateur les variables utilisé pendant l’exécution du processus de notre système.

4. Bibliothèque d’images :

Voici là c’est collection des images que nous avons travaillé avec pendant l’exécution à l’aide de notre système :

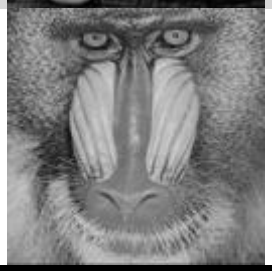
Bibliothèque des images				
Nom : **Lena** Taille Physique : **3.63 ko** Dimension : **128 X 128** Type de fichier : **BMP** Type de fichier après transformation : **BMP**		**Nom :** Avion **Taille Physique :** 3.55 ko **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	
Nom : **Barbara** Taille Physique : **3.63 ko** Dimension : **128 X 128** Type de fichier réel : **BMP** Type de fichier après transformation : **JPG**		**Nom :** Clown **Taille Physique :** 3.58 **ko** **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	
Nom : **Fruit** Taille Physique : **3.58** ko Dimension : **128 X 128** Type de fichier réel : **BMP** Type de fichier après transformation : **JPG**		**Nom :** Pimen **Taille Physique :** 3.76 **ko** **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	
Nom : **Mandr** Taille Physique : **4.11 ko** Dimension : **128 X 128** Type de fichier réel : **BMP** Type de fichier après transformation : **JPG**		**Nom :** House **Taille Physique :** 3.78 **ko** **Dimension :** 128 X 128 **Type de fichier réel :** BMP **Type de fichier après transformation :** JPG	

Tableau 1: Bibliothèques des images utilisées

5. Tests expérimentaux :

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

5.1. 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, ensuit nous appliquons le cryptage et décryptage de ces images et enfin la décompression (comme le montre la Figure 43), en discutant sur les

paramètres de performances de cette dernières, parmi ses paramètres : MSE, PSNR, le taux de compression, le temps de cryptage et décryptage et le temps de compression/ décompression. Le Tableau 02 présente les résultats obtenus.

L'entrer est une image.

La clé utilisé dans les tests qui suit est en hexadécimale sur 16 bits :

```
Key='000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f';
```

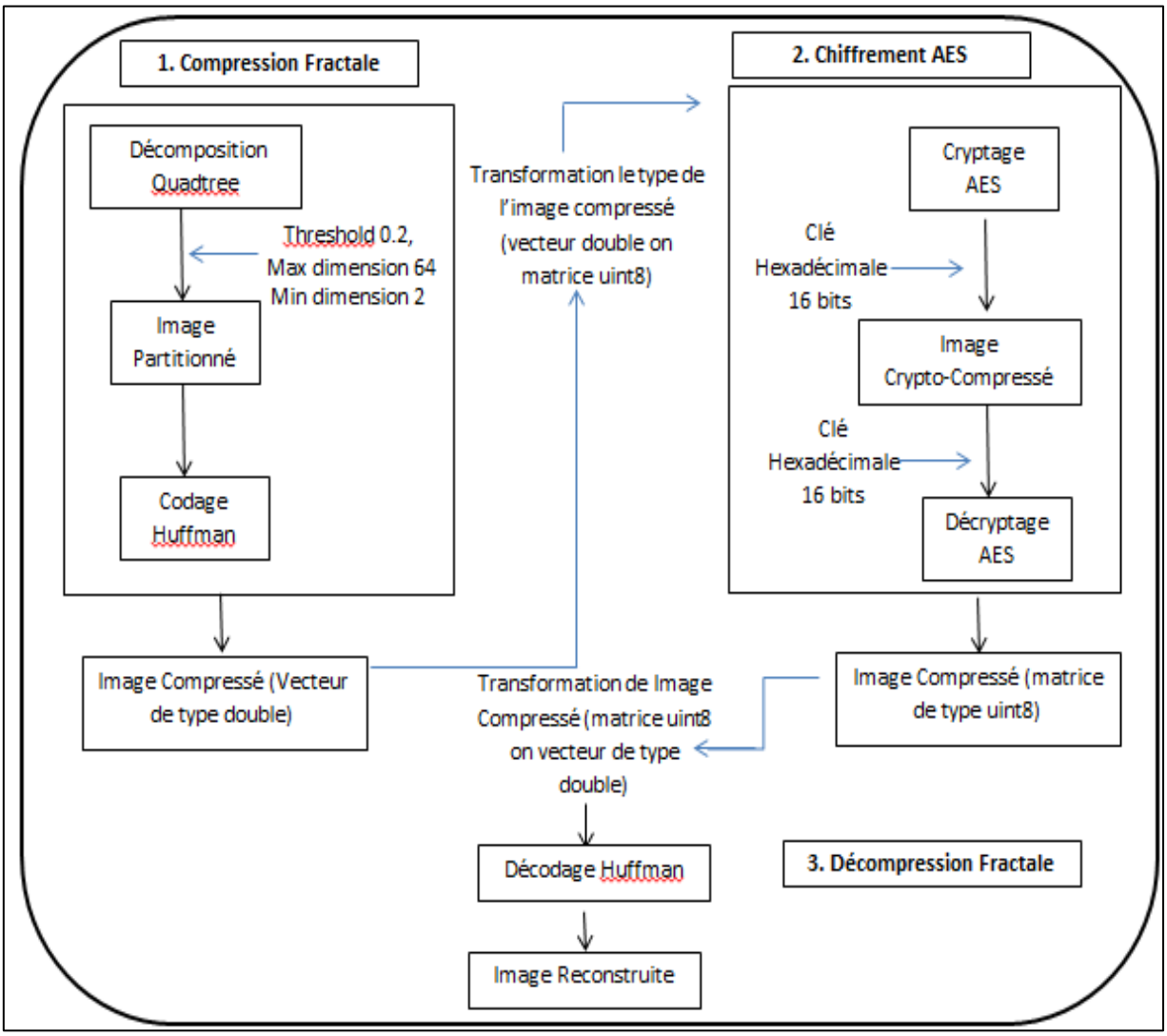


Figure 43 : Schéma synoptique de notre système

5.1.1. Testes et résultats :

Les images de la collection étudiée, sont compressées avec la compression Fractale suivant l'algorithme hybride Décomposition Quadtree -Codage Huffman, et chiffré suivant l'algorithme AES -256.

5.1.1.1. Premier Cas :

Comme nous avons montré dans la figure 40 et 43 notre système applique la compression comme une première étape ensuite le cryptage et le décryptage et enfin la décompression, alors c'est notre premier test, et voilà le Tableau 02 qui présente les résultats obtenus.

L'image			Test	Paramètres						
Nom Taille O Taille R	**Dimension**	**Type**		**CR%**	**MSE (Err)**	**PNSR (db)**	**Temps compression (S)**	**Temps Cryptage (S)**	**Temps Décryptage (S)**	**Temps Décompression (S)**
Lena **3.63 ko** **2.90 ko**	128 X 128	BMP	01	9.98	38.68	33.369	2.97	772.015	674.825	25.60
Barbara **3.71 ko** **3.01 ko**	128 X 128	BMP	02	9.55	43.18	31.77	2.84	808.556	698.263	24.40
Clown **3.66 Ko** **2.99 Ko**	128 X 128	BMP	03	9.86	40.97	32.005	3.61	790.324	660	21.30
Avion **3.63 ko** **2.68 ko**	128 X 128	BMP	04	11.25	36.45	32.513	2.78	770.299	704.179	30.04
Mandr **4.21 ko** **2.92 ko**	128 X 128	BMP	05	8.67	51.11	31.043	3.17	921.731	758.229	25.59
Fruit **4.27 ko** **3.30 ko**	128 X 128	BMP	06	8.08	48.02	31.310	3.71	969.586	790.309	29.22
House **3.87 ko** **2.94 ko**	128 X 128	BMP	07	9.10	44.67	31.630	3.33	861.691	709.709	27.16
Boat **3.30 ko** **2.36 ko**	128 X 128	BMP	08	12.09	35.14	32.599	4.31	690.886	525.823	15.294
Isabe **3.60 ko** **2.42 ko**	128 X 128	BMP	09	12.90	42.18	31.87	3.31	761.956	608.786	15.199
Pimen **3.86 ko** **3.23 ko**	128 X 128	BMP	10	9.62	40.73	32.03	3.15	1019.04	745.595	24.11

Tableau 2 : Résultat d'application de notre système sur différentes images

5.1.1.2. Processus de traitement et Histogramme :

Voici c'est les résultats obtenues par la première variante où on a appliqué le protocole d'exécution de notre système :

Test 01 : Lena.jpg :

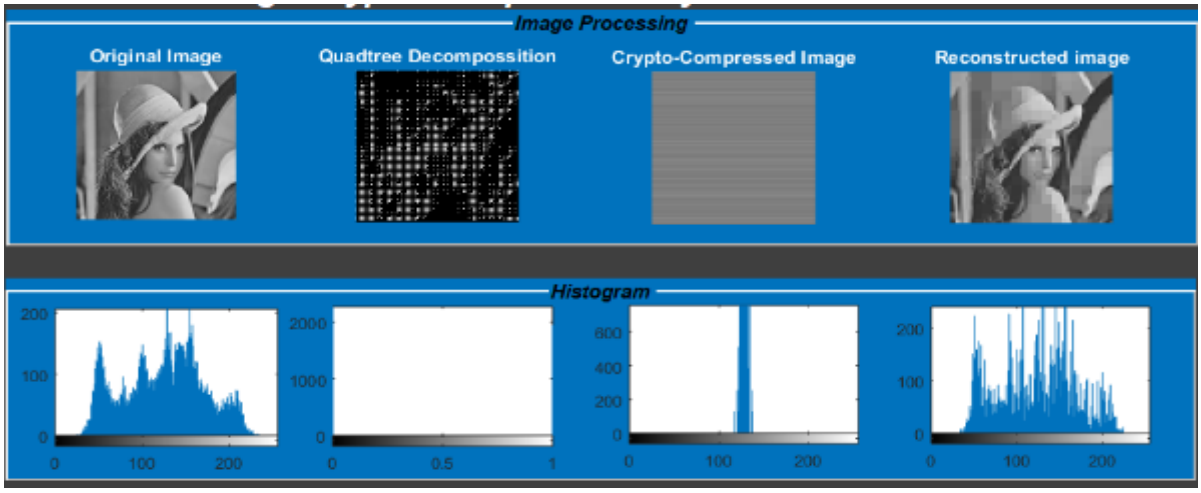


Figure 44 : Processus de traitement et l'histogramme Lena.jpg

Test 02 : Brabara.bmp :

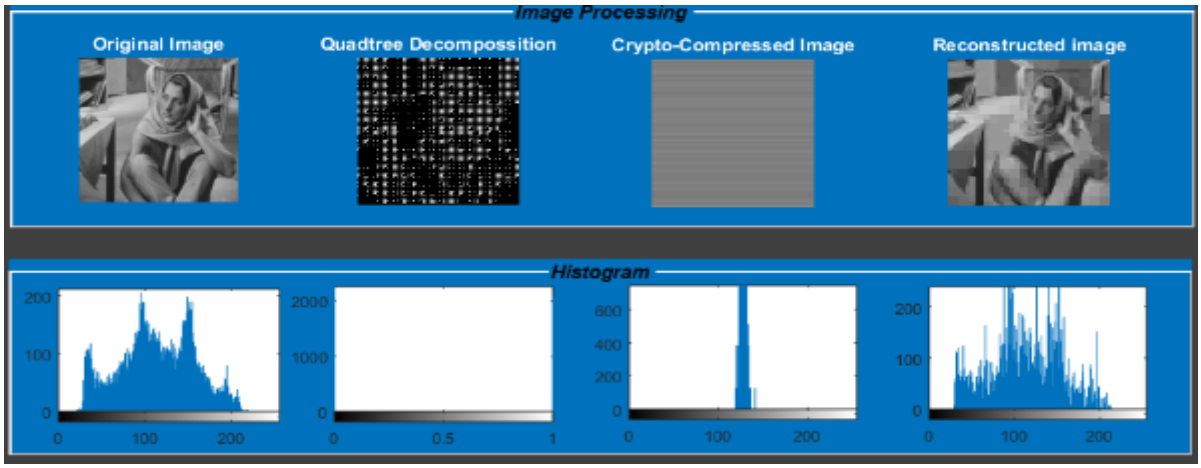


Figure 45: Processus de traitement et l'histogramme Brabara.bmp

Test 03 : Clown.bmp :

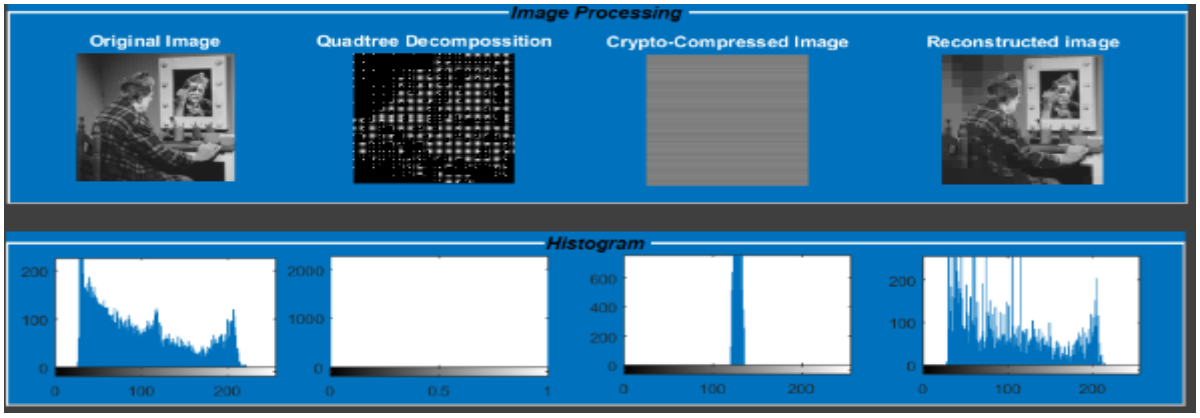


Figure 46: Processus de traitement et l'histogramme Clown.bmp

Test 04 : Avion.bmp :

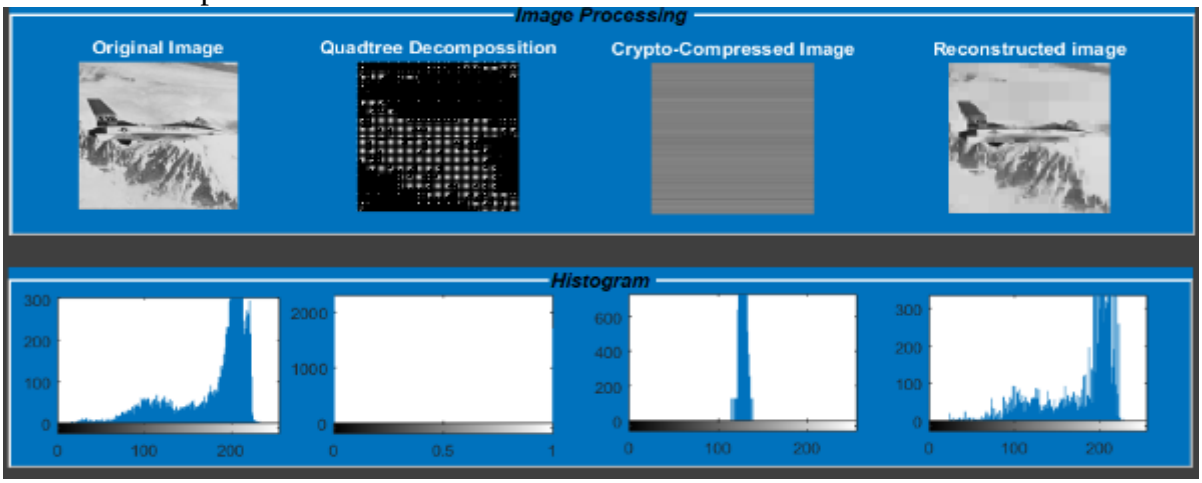


Figure 47: Processus de traitement et l'histogramme Avion.bmp

Test 05: Mandr.bmp:

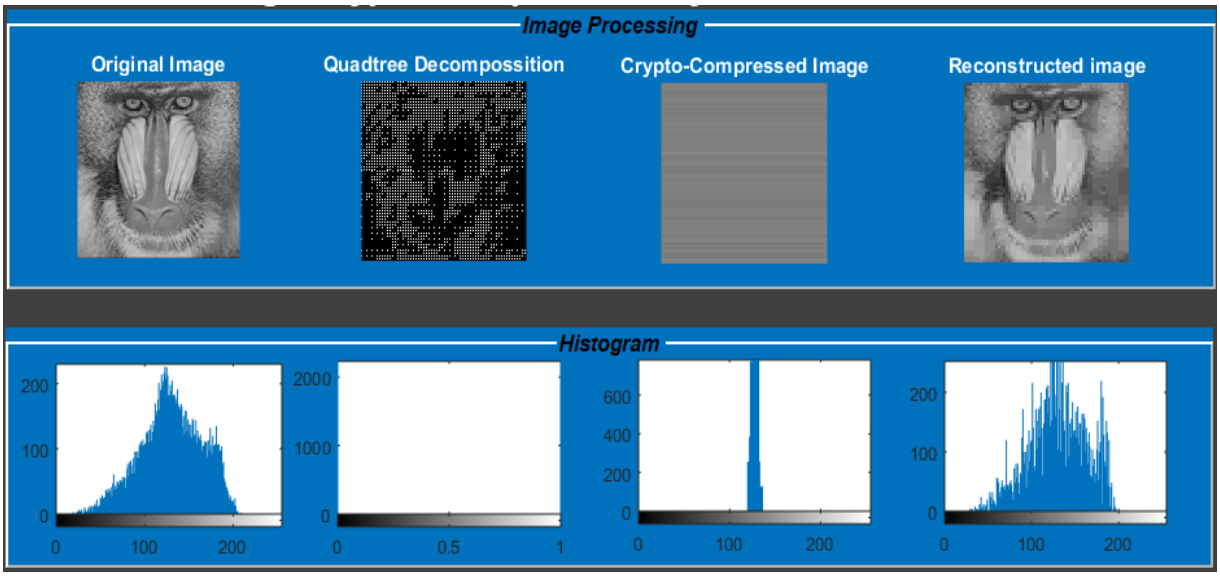


Figure 48 : Processus de traitement et l'histogramme Mandr.bmp

Test 06: Fruit.bmp:

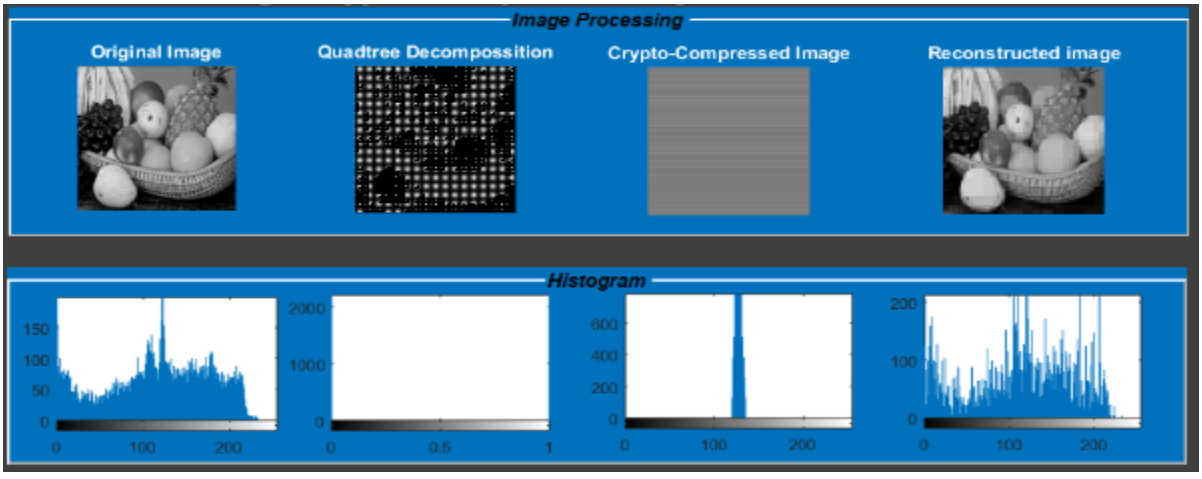


Figure 49: Processus de traitement et l'histogramme Fruit.bmp

Test 07: House.bmp:

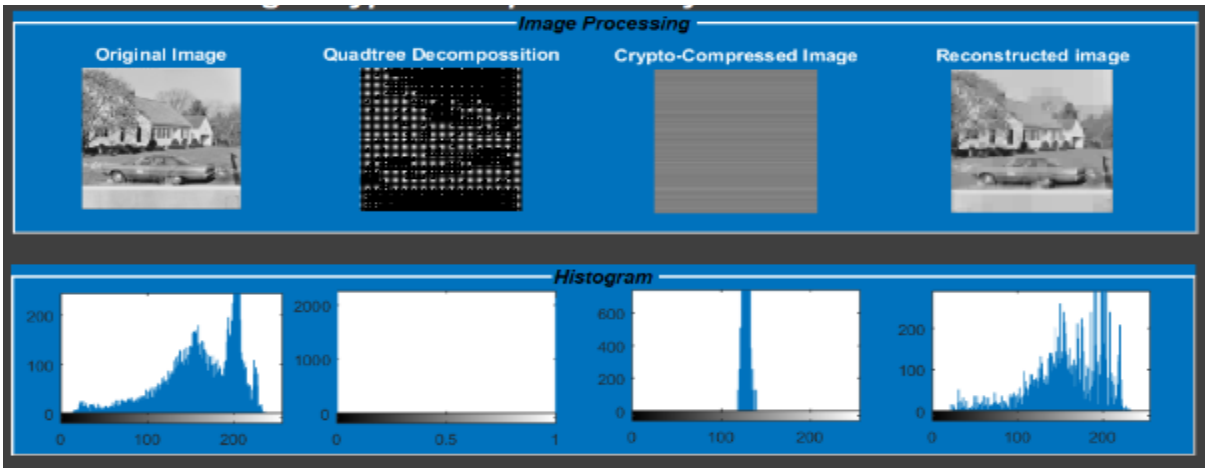


Figure 50: Processus de traitement et l'histogramme House.bmp

Test 08: Boat.bmp:

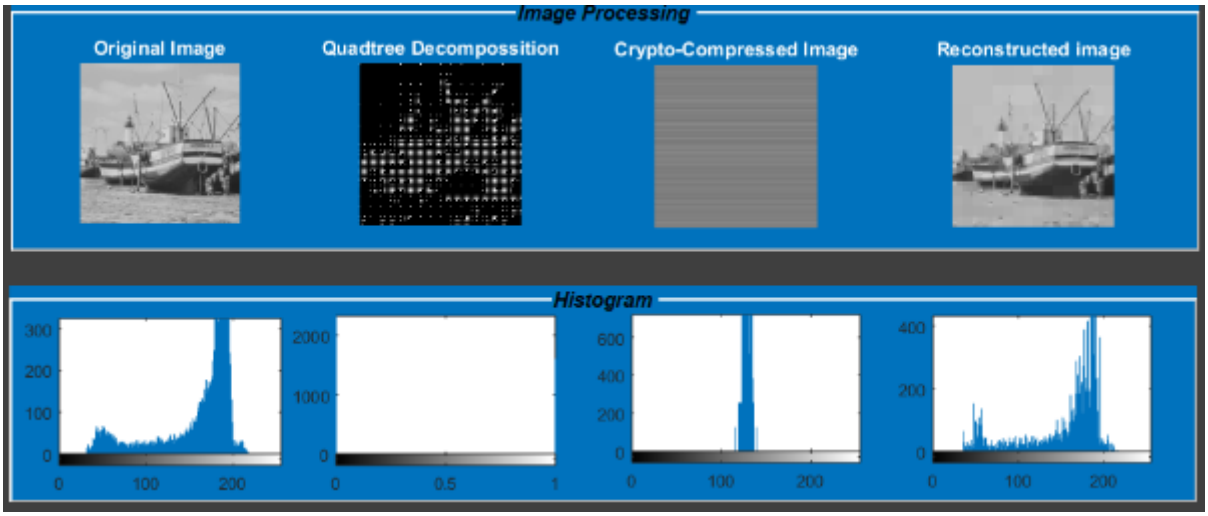


Figure 51: Processus de traitement et l'histogramme Boat.bmp

Test 09: Isabe.bmp:

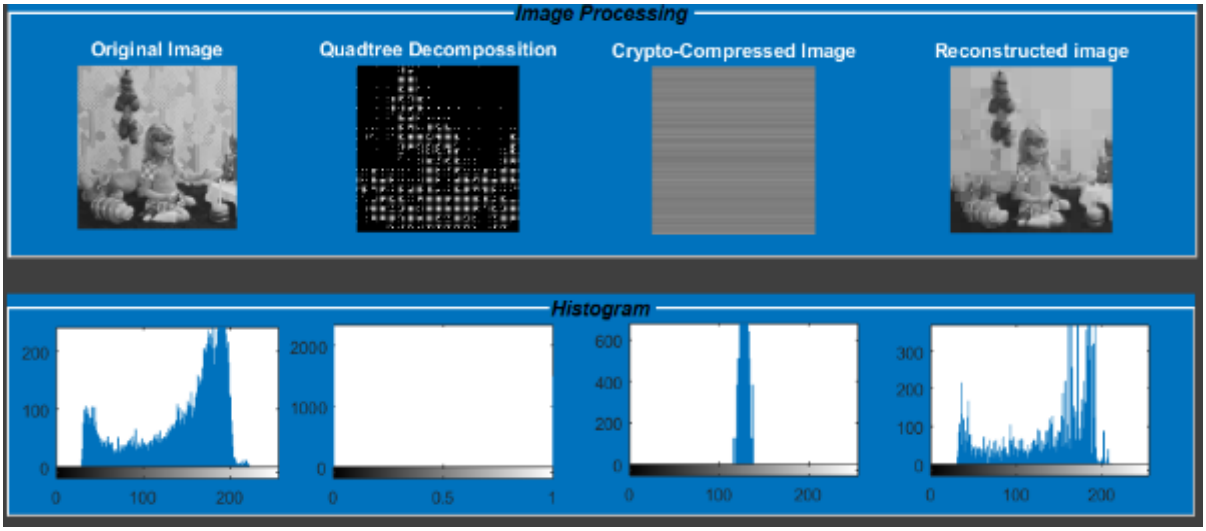


Figure 52: Processus de traitement et l'histogramme Isabe.bmp

Test 10: Pimen.bmp:

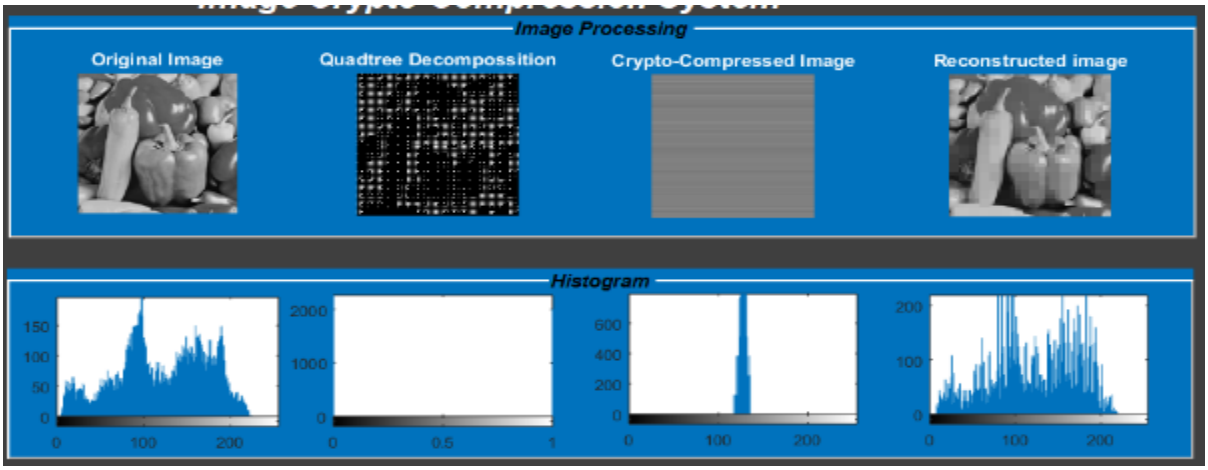


Figure 53: Processus de traitement et l'histogramme Pimen.bmp

5.1.1.3. Deuxième Cas :

Nous essayons maintenant de changer l'ordre d'appliquer les étapes de notre système c'est-à-dire nous allons appliquer le cryptage comme une première étape ensuite la compression et décompression et enfin le décryptage sur les mêmes images utilisées dans le premier cas, et on va calculer les paramètres de performances tel que le CR, Temps de Cryptage, Temps de Compression, Temps de Décompression et Temps de Décryptage.

L'entrer est une image.

La clé utilisé dans les tests qui suit est en hexadécimale sur 16 bits :

```
Key='000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f'
```

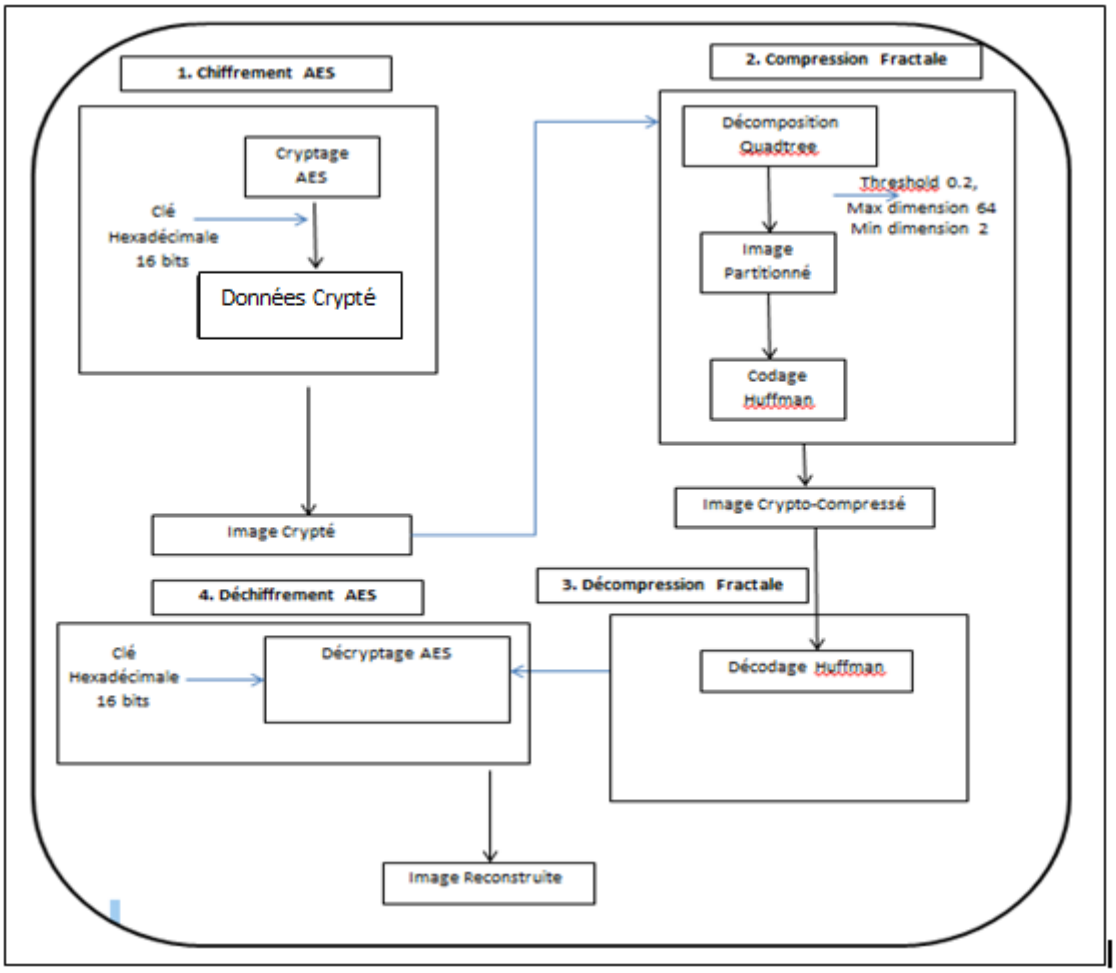


Figure 54 : Schéma synoptique du deuxième cas

L'exécution d'enchainement des étapes dans ce cas se fait parfaitement sans aucune erreur d'exécution mais les résultats obtenus ne le sont pas, comme le montrent les Figures suivantes :

Workspace

Name	Value	Size	Class	Min	Max
blkcount	4096	1x1	double	4096	4096
CompRatio	5.0822	1x1	double	5.0822	5.0822
Data	*16387x1 single*	16387x1	single	0	246
Dict	*217x2 cell*	217x2	cell		
Image_crypt_comp	*99689x1 double*	99689x1	double	0	1
Image_crypte	*128x128 uint8*	128x128	uint8	0	255
Image_Recons	*128x128 uint8*	128x128	uint8	0	255
Img	*128x128 uint8*	128x128	uint8	21	239
Img_crypte_dcmp	*128x128 uint8*	128x128	uint8	22	246
key	'000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f'	1x64	char		
lena	*220x220x3 uint8*	220x220x3	uint8	0	255
Quadtre	*128x128 double*	128x128	double	0	2
Tcmp	6.1721	1x1	double	6.1721	6.1721
Tcrp	128.7782	1x1	double	128.7782	128.7782
TDcomp	82.6942	1x1	double	82.6942	82.6942
TDcrypt	93.3261	1x1	double	93.3261	93.3261

Figure 55: Résultat du deuxième Cas Lena.jpg

Workspace

Name	Value	Size	Class	Min	Max
barba	*512x512x3 uint8*	512x512x3	uint8	<Too man...	<Too man...
blkcount	4096	1x1	double	4096	4096
CompRatio	5.0796	1x1	double	5.0796	5.0796
Data	*16387x1 single*	16387x1	single	0	237
Dict	*219x2 cell*	219x2	cell		
Image_crypt_comp	*99711x1 double*	99711x1	double	0	1
Image_crypte	*128x128 uint8*	128x128	uint8	0	255
Image_Recons	*128x128 uint8*	128x128	uint8	0	255
Img	*128x128 uint8*	128x128	uint8	21	232
Img_crypte_dcmp	*128x128 uint8*	128x128	uint8	20	237
key	'000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f'	1x64	char		
Quadtre	*128x128 double*	128x128	double	0	2
Tcmp	5.9425	1x1	double	5.9425	5.9425
Tcrp	117.4616	1x1	double	117.4616	117.4616
TDcomp	83.1003	1x1	double	83.1003	83.1003
TDcrypt	186.2532	1x1	double	186.2532	186.2532

Figure 56:Résultat du deuxième Cas Barbara.jpg

Les Figures 55 et 56 montrent qu'à cause du type de l'algorithme de compression Fractale (avec perte d'information), l'algorithme de Cryptage AES qui se base sur le block Cipher, trouve que les informations Cryptées dans la première étape ne sont pas les mêmes, alors Il n'a pas pu décrypter ces informations.

Dans chaque résultat obtenu nous remarquons que l'image originale « Img» est de type ''uint8'' avec min valeur 21 et max 232, et l'image reconstruite «Image_Recons » est aussi de type ''uint8 '', mais les valeurs min et max ne sont pas les mêmes ce qui implique que l'algorithme de décryptage n'a pas pu de décrypter les informations cryptées auparavant.

6. Interprétation des résultats :

Un algorithme performant de compression possède un gain de compression maximale et une erreur quadratique moyenne minimale, pour cela nous calculons le Taux de compression et le gain de compression en utilisant les résultats précédents obtenus dans le Tableau 02 (le Tableau suivant présente ces deux paramètres).

- Taux de compression : $\boldsymbol{T = \frac{1}{CR}}$
- Gain de compression : $\boldsymbol{G = 1 - T}$

L'image	Test	CR %	T	G
Lena	01	9.97	**0,1003**	**0,8997**
Barbara	02	9.55	**0,1047**	**0,8953**
Clown	03	9.86	**0,1014**	**0,8986**
Avion	04	11.25	**0,0888**	**0,9112**
Mandr	05	8.67	**0,1114**	**0,8886**
Fruit	06	8.08	**0,1237**	**0,8763**
House	07	9.10	**0,1098**	**0,8902**
Boat	08	12.09	**0,0827**	**0,9173**
Isabe	09	12.90	**0,0775**	**0,9225**
Pimen	10	9.62	**0,1039**	**0,8961**

Tableau 3: Taux et Gain de Compression

Dans le Tableau précèdent on remarque que les valeurs des Gain de compression sont dans l'intervalle [0.89, 0.92], c'est-à-dire sont proches du « un », ce qui implique une compression performante.

Comme on a vu dans le premier chapitre, pour mesurer la distorsion entre l'image reconstruite et l'image originale (Mesure de la qualité visuelle de l'image reconstruite) on va utiliser l'Erreur Quadratique Moyenne MSE (Mean Square Error) ou du rapport signal à bruit PSNR (Peak Signal to Noise Ratio). Dans le Tableau 02, les valeurs des MSE et PNSR dans l'intervalle [35,51], [30, 32], sont un peu élevés.

L'histogramme d'une image mesure la distribution des niveaux de gris dans l'image. Pour un niveau de gris x, l'histogramme permet de connaitre la probabilité de tomber sur un pixel de valeur x en tirant un pixel au hasard dans l'image.

- Concrètement, l'histogramme d'une image à valeurs entières est construit de la manière suivante: pour chaque niveau de gris x, on compte le nombre de pixels ayant la valeur x.
- L'histogramme permet d'obtenir rapidement une information générale sur l'apparence de l'image. Une image visuellement plaisante aura généralement un histogramme équilibré (proche d'une fonction plate).
- On prend notre résultat obtenues sur les différentes images par notre système nous avons remarqué que l'histogramme est tassé sur le centre dans la plupart des tests (01, 02, 05, 06, 10), mais dans les tests (04, 07, 08, 09) l'histogramme est tassé sur la droite, et le test 03 l'histogramme est tassé sur la gauche.
- On remarque que la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse, dans l'image originale n'est pas la

même dans l'image reconstruite ce qui implique la perde d'information grâce à l'algorithme de compression.

6.1. Discussion :

6.1.1. Premier cas :

Dans notre système de crypto-compression on a appliqué le chiffrement AES après la compression Fractale, D'après les résultats obtenus 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 a vu, on perd de la qualité de l'image et la reconstitution n'est pas fidèle.

6.1.2. Deuxième Cas :

Nous avons essayé d'appliquer le système d'une autre façon où on a appliqué le chiffrement avant la compression : D'après les résultats obtenus on a remarqué, qu'on a des problèmes dans la reconstitution de l'image puisque les données ou les informations cryptées au début du processus (étape de chiffrement avant la compression) au niveau d'image, ne sont pas les mêmes (Dernière étape du processus décryptage après compression et décompression). Cela est causé par l'algorithme de compression fractale (avec perte d'information).

7. Conclusion :

Dans la première partie de ce chapitre, nous avons présenté l'environnement de travail et le langage de programmation que nous avons utilisé, ainsi que notre système de crypto-compression et les modules de fonctionnement de ce système.

Dans la deuxième partie de ce chapitre nous avons testé plusieurs types d'image à l'entrée de notre système. Ensuite nous avons appliqué le chiffrement sur la compression et nous avons enregistré les résultats.

Dans la dernière partie nous avons discuté sur les résultats obtenus, et d'après les résultats présentés, on remarque bien que le chiffrement AES agit sur la robustesse des techniques compressions ainsi sur la reconstruction de l'image comme.

On a vu aussi, que si on applique le chiffrement avant compression. l'image obtenue est complètement bruitée, cela est dû à la perte d'information causée par l'algorithme de compression utilisé qui est avec perte.

CONCLUSION GENERALE

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 compression fractale en utilisant le partitionnement Quadtree et le codage de Huffman ; et pour le chiffrement on a choisi le système de chiffrement par bloc, et on a utilisé l'algorithme de cryptage AES-256.

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 la compression sur le chiffrement et l'inverse.

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

Liste des abréviations :

A:

- AES : Advanced Encryption Standard

B:

- BMP : BitMaP

C:

- CR : En anglais « Compression Ratio » : Rapport de compression.
- Cipher Block Chaining (CBC).
- Cipher Feedback(CFB).

D:

- DES : Data Encryption Standard
- DB : Le décibel ou (dB)

E:

- Electronic Code Book(ECB)

G:

- GSM : Global System for Mobile Communications
- GUI (pour Graphical User Interface

I:

- IFS (Iterated Function System)
- ISO : International Organisation for Standardisation).
- IRM : L'imagerie par résonance magnétique (IRM)

J:

- JPG : Joint Photographic Group

L :

- LZW : LZW (pour Lempel-Ziv-Welch)

M :

- MSE : En anglais « Mean Square Error » L'Erreur Quadratique Moyenne.

O :

- Output Feedback (OFB).

P :

- PSNR: En anglais « Peak Signal to Noise Ratio » Rapport signal à bruit.

R :

- RLC : Le run-length encoding, appelé en français le codage par plages.
- RSA : **R**onald Rivest, Adi **S**hamir et Leonard **A**dleman.
- RGB : Rouge, vert, bleu, abrégé en RVB ou en RGB
- RC4 (Rivest Cipher 4)

BIBLIOGRAPHIE

Bibliographie :

[1] M. ALDOSSARI, «Nouvelle méthode optique de compression et de cryptage simultanés des images (fixes/vidéo) pour les systèmes télécommunication,» *" HAL" Thèse de Doctorat UNIVERSITÉ DE BRETAGNE OCCIDENTALE* , p. 28, 02 Février 2006.

[2] S. Pigeon, *Contribution à la compression de données,Thèse présentée à la Faculté des arts et sciences en vue de l'obtenation du grade Philosophie Doctor en Informatique,* Montréal,Canada: Département d'informatique et de recherche opérationnelle, 2001.

[3] A. Mourad, «Crypto compression d'image par cryptage partiel En vue de l'obtention d'un Master II en Réseau et télécommunication,» Université Mouloud Mammeri de Tizi-Ouzou, 2015.

[4] Z. Athmane, «Ondelettes et techniques de compression d'images numérique,» *THESE pour l'obtention du Diplôme de Doctorat en Sciences en Electronique,* pp. 5-16, 2012/2013.

[5] S. Renard, «La compression des données,» chez *Club Photoshop de Nantes,* 14 Octobre 1999.

[6] L. Diane, Rapport de recherche "Cours de traitements d'images" , Centre National de la Recherche Scientifique,, ISRN I3S/RR–2004-05–FR, 22 Janvier 2004.

[7] C. TAOUCHE, «Implémentation d'un Environnement Parallèle pour la Compression d'Images à l'aide des Fractales , Memoire Pour l'obtention du diplôme de Magister en Informatique Option Information & Computation,» Université Mentouri Faculté des Sciences de l'Ingénieur Département d'Informatique, Constantine, 2005.

[8] P. Plumé, «Techniques de compression de données,» *Edition EYROLLES et la revue « PC EXPERT »*, pp. 1-6, Janvier 1995..

[9] C. Wagner, «De l'image vers la compression. [Rapport de recherche] RR-2035, INRIA.1993,» INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (INRIA), Rennes, 24 May 2006.

[10] M. B. Mme Dougherty, «Les formats de compression d'image,» Département Génie Électrique et Informatique Industrielle, Institut Universitaire de Technologie de Tours, Promotion 2002-2004.

[11] Y. Fisher, «FRACTAL IMAGE COMPRESSION,» Super Computer Center, University of California, San Diego.

[12] J. M. Florian AGEN, «La Compression Fractale, Méthodes de Jacquin, Subdivisions de triangles et Delaunay,» UNIVERSITE François Rabelais TOURS Polytech'Tours-Département Informatique, Juin 2005.

[13] N. H.-S. Adda ALI-PACHA, «Compression des Images Fixes par Fractale : Partitionnement Quadtree,» 2007.

[14] «Guide de Network Associates International BV sur la cryptographie».

[15] S. Tag, Écrivain, *Support de Cours de 1er Année Master Sécurité Informatique.* [Performance]. 2018-Octobre.

[16] C. E. SHANNON, «A Mathematical Theory of Communication,» *The Bell System Technical Journal,*, p. 55, July, October, 1948..

[17] A. L. DAHMANE Zouhir, «Implémentation d'un algorithme de cryptage sur un circuit FPGA,» chez *MEMOIRE DE MASTER*, Mai 2017, pp. 8-12.

[18] B. Rabab, «THESE DE DOCTORAT 3ème Cycle Sécurité des images Numériques compressées JPEG,» Université Djilal Liebes– Sidi Bel Abbes.

[19] J.-M. C. Franck Davoine, «COMPRESSION D'IMAGES PAR FRACTALES,» Laboratoire TIMC-IMAG Institut Albert Bonniot, LA TRONCHE Cedex France.

[20] a. A. Veenadevi, «FRACTAL IMAGE COMPRESSION USING QUADTREE DECOMPOSITION AND HUFFMAN CODING,» *An International Journal (SIPIJ) Vol.3, No.2,* n° %1209, pp. 209-210, April 2012.

[21] Z. B. K. A. S. A. Z. Hamid Meraoubi, «Un système de crypto-compression des images médicales basé sur la DCT 2x2-IDS et l'AES,» Centre de Développement des Technologies Avancées, Baba Hassen, Alger.

[22] S. Guillem-Lessard, «Tutoriel de Cryptographie,,» 2002.

[23] A. C. e. F. Lévy-dit-V'hel, «La cryptographie moderne, Revue Armement,» 2001.

[24] P. Deshmukh, «An image encryption and decryption using AES algorithm,» *International Journal of Scientific & Engineering Research*, Vols. %1 sur %2Volume 7, Issue 2, n° %1212, pp. 210-213, February-2016.