**Université Mouloud MAMMERI de Tizi-Ouzou**

**République Algérienne Démocratique et populaire**

**Ministère de l'enseignement supérieur et de la recherche scientifique**

**Université Mouloud Mammeri de Tizi-Ouzou**

**Faculté de génie électrique**

**Département Electronique**

**Spécialité Réseau et télécommunication**

***Mémoire***

***En vue de l'obtention d'un Master II en*** **Réseau et télécommunication**

# Thème du mémoire :

# Crypto compression d'image par cryptage partiel

**Réalisé par :**

Mr AMRANE Mourad

**Encadré par :**

Mr LAHDIR Mourad

**Septembre 2015**

# REMERCIMENTS

*Ce travail a été effectué dans le cadre de la préparation du diplôme de master académique en réseaux et télécommunications Mouloud Mammeri de Tizi-Ouzou.*

*Je tiens à exprimer mes profondes gratitudes et mon immense respect a mon promoteur* ***Mr. LAHDIR MOURAD*** *pour la qualité de son encadrement, sa disponibilité, ses hautes qualités morales et scientifiques et pour m'avoir découvrir un domaine de recherche si passionnant et aussi pour ses conseils précieux et son soutien affectif durant mon étude et réalisation de ce projet.*

*Mes remerciements les plus vifs s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.*

# Dédicaces

Je dédie ce mémoire à :

- A mon père, qui a fait tant de sacrifice pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.
- A Ma mère, qui a tout fait pour ma réussite, son soutien, tous les sacrifices qu'elle a consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie.
- A Mes frères qui n'ont cessé d'être là pour moi des exemples de persévérance, de courage et de générosité.
- A Mes professeurs de l'Université de ***Mouloud MAMMERI*** de Tizi-Ouzou et en particulier, ceux du département de génie électrique et surtout à Monsieur ***LAHDIR Mourad.***
- A tous mes amis et camarades et tous ceux qui ont contribué à la réalisation de ce travail.

***AMRANE Mourad***

# SOMMAIRE

**Chapitre 3 : La compression JPEG**



**Chapitre IV : Application**

# Table des figures

# Résumé

Dans ce travail, nous proposons le test de plusieurs techniques de compression sur des images fixes et aussi la combinaison entre compression et cryptage des images.

Pour la compression des images fixes, généralement, l'utilisateur ne s'intéresse qu'à certaines zones d'une même image. Ceci suggère que ces différentes zones peuvent être traitées par des approches différentes réversibles ou irréversibles. Nous proposons une approche de compression adaptative à l'aide de méthodes irréversibles. Après sélection des zones qualifiées d'intérêt, l'approche consiste à appliquer une compression par DCT sur ces zones et une compression par la méthode classique JPEG avec pertes sur le contexte de l'image. Le test de cette approche sur des images avec une ou plusieurs zones d'intérêts, a révélé la supériorité de cette approche par rapport aux méthodes de compression classique en termes de taux de compression. En plus, notre approche s'avère toujours plus efficace en termes de taux de compression qui est réglable pour une qualité visuelle comparable.

Pour le cryptage qui est historiquement développer pour garantir le secret dans la messagerie, le cryptage des informations est maintenant utilise pour interdire l'accès ou la modification des informations sensibles et garantir la confidentialité dans les applications informatiques. Nous proposons une approche de cryptage à l'aide de ses techniques symétriques et asymétriques, cette approche consiste à étudier le développement et le fonctionnement des algorithmes standard des techniques de cryptage tel que le DES, RSA et le AES.

Enfin, pour la crypto-compression, Le développement des applications liées à plusieurs domaines de traitement d'images nécessite l'utilisation des technologies de l'information et des télécommunications qui ont évolué très rapidement ces dernières années. La compression et le cryptage de données sont deux techniques dont l'importance croit d'une manière exponentielle dans une myriade d'applications. L'usage des réseaux informatiques, pour la transmission et le transfert des données, doit satisfaire à deux objectifs qui sont la réduction du volume des informations pour désencombrer le maximum possible, les réseaux publics de communications et la protection en vue de garantir un niveau de sécurité optimum. Pour cela nous avons proposé une approche hybride de crypto-compression, qui repose sur une association d'algorithmes de cryptage tel que l'algorithme DES et l'algorithme AES avec des algorithmes de compressions. Cette méthode de crypto-compression a bien montré sa bonne performance concernant sont usage dans différents domaines tel que les réseaux informatique, la transmission et le transfert de données.

**Mots clefs :** Compression d'image, algorithmes standard de cryptages, DCT, quantification, cryptage partiel, crypto-compression.

**Abstract**

The trafic of digital imges has been increased rapidly in the internet. Security of image becomes important for many sectors mainly for medical application. Nowadays, the transmission of images is a daily routine, especially over wirless (battlefields, traffic accidents, etc). Partial encryption is an approach to reduce the computational resources for huge volumes of multimedia data in this kind of network.

This paper presents a method of partial or selective encryption for JPEG images. It is based on encryption of the quantifed DCT coefficients. The purposed method results in a significant reduction in encryption and decryption processing time. It is fast and does not reduce the compression performance of the JPEG algorithm.

# Introduction

## Introduction

Le transfert d'images est encore actuellement peut sécuriser. Les algorithmes standards de chiffrement ne conviennent pas au cas particulier des images. L'idéal serait de pouvoir appliquer des systèmes de chiffrement asymétriques a fin de ne pas avoir de clés à transférer. Du fait de la connaissance de la clef publique, les systèmes asymétriques sont très couteux en temps de calcule, et donc pas envisageable pour un transfert sécuriser d'images. Les algorithmes symétriques imposent de transférer la clef secrète. Les méthodes classiques de chiffrement d'images nécessite le transfère de la clef secrète par un autre canal ou un autre moyen de communication [2, 1,5].

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 limage n'est pas maximale. Le second problème et 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.

Pour le transfère d'images les algorithmes de chiffrement d'images doivent pouvoir être combinée avec les algorithmes de compression d'images tels que JPEG [4]. Le problème et que les algorithmes de cryptage ont pour objectif de supprimer toute les redondances afin d'éviter des attaques statistiques alors que les algorithmes de compression cherchent les redondances contenues dans les images afin de réduire la quantité d'information.

Dans ce résumé nous proposons une méthode cryptant partiellement le contenue d'une image afin de garder une certaine quantité de redondance permettant de comprimer l'image partiellement cryptée. Pour cela nous intégrant notre algorithme de cryptage partiel dans la chaine de l'algorithme de compression JPEG. A ce jour, peu de travaux proposent des solutions de cryptage partiel. En combinant compression et cryptage [3] conclurent que pour obtenir un haut niveau de confidentialité, au minimum 12,5% de données doivent être chiffrées.

Apre avoir rappelé les base de l'algorithme JPEG, nous présentons notre stratégie de cryptage partiel et nous présentant les résultats de notre méthode appliquer a une image.

# Chapitre I

# Chapitre1 : Généralités sur le cryptage et la compression

## 1.1 Préambule :

La puissance des processeurs augmente plus rapidement que les capacités de stockage, et énormément plus vite que la bande passante d'Internet qui demande beaucoup de changements d'infrastructures telles que les installations téléphoniques. Ainsi il semble plus simple de réduire la taille des données plutôt que d'augmenter les espaces de stockage et/ou les infrastructures téléphoniques.

La réduction de la taille des données se fait donc par la compression du codage du fichier source. Ainsi pour réduire la taille d'une image, il faut arriver à diminuer son code source, en trouvant un algorithme qui le convertira au mieux en un code moins redondant, et en restituant l'image originale avec le moins de pertes possibles.

Nos études portent donc sur les principales manières de compresser les images numériques fixes. Nous avons d'un côté les compressions dites conservatrices telles que le codage RLE et le codage LZW car l'image une fois compressée est identique à l'originale, il n'y a donc aucune perte de qualité. De l'autre côté, nous avons les compressions non conservatrices : elles restituent l'image originale avec des pertes plus ou moins minimes de qualité, on s'intéressera plus particulièrement à la norme JPEG, qui permette de choisir ce niveau de pertes. Nous dégagerons finalement les spécificités de chaque type de compression.

Le cryptage est historiquement développé pour garantir le secret dans la messagerie, le cryptage des informations est maintenant utilisé plus largement pour interdire l'accès ou la modification des informations sensibles et garantir la confidentialité dans les applications informatiques. Cependant, le cryptage n'est qu'un élément dans l'ensemble des dispositifs d'un système complexe. La protection qu'il assure n'est valable que si elle s'insère dans un ensemble cohérent.

Après une synthèse des techniques de cryptage, cette présentation décrit sommairement les dispositifs de sécurité et de confidentialité, face aux risques spécifiques du domaine médical. Les principes généraux d'une approche de la sécurité et de l'intégration du cryptage des données sont évoqués dans la conclusion.

## 1.2 Généralités sur la compression en imagerie

### 1.2.1 Rappel sur limage numérique

L'acquisition d'une image se fait par moyen d'un appareil photo ou une prise d'un film sur une camera (suite d'images).la nature brute de l'image a la sortie des capteurs est un signal analogique qui doit être discrétisé (numérisé) pour pouvoir l'exploité par un processeur de données numériques.

#### 1.2.1.1 Définition de limage numérique :

L'image numérique correspond à une matrice (ensemble ordonné à deux ou trois dimensions) de données numériques. Nous nous intéresserons uniquement aux images en deux dimensions. On peut concevoir ces images en deux dimensions comme un tableau de valeurs auxquelles on fait correspondre une position sur un plan (x, y) et une couleur pour visualiser l'image sur l'écran d'un ordinateur [1] :

– Exemple d'un détail d'une image binaire dont la couleur est codée en 0 ou en 1 :

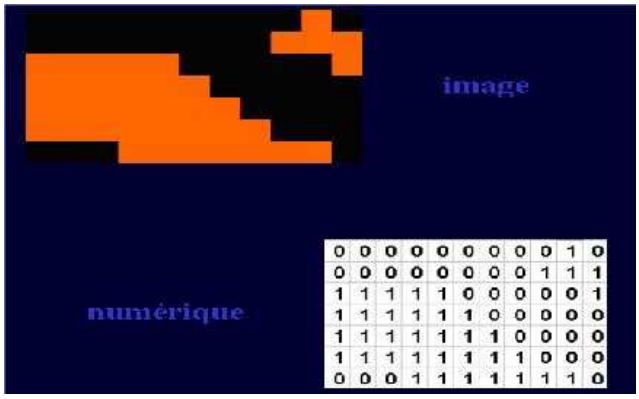



Fig. 1.1 – Détail d'une image binaire.

– Exemple d'un détail d'une image en niveaux de gris dont la valeur de gris est codée entre 0 et 255 :

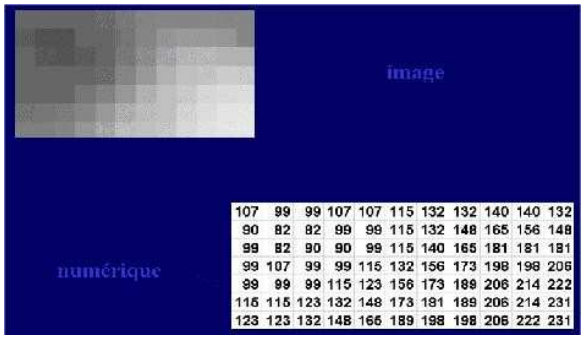



**Fig. 1.2 – Détail d'une image en niveaux de gris.**

Une image numérique 2D est donc composée d'unités élémentaires appelées pixels (ou " picture elements ") qui représente chacune, une portion de l'image, codée par des valeurs numériques. Une image est définie par le nombre de pixels qui la composent en largeur et en hauteur (qui peut varier théoriquement presque à l'infini) et l'étendue des teintes de gris ou des couleurs que peut prendre chaque pixel (on parle de dynamique de l'image). Toutes les données correspondant aux informations chiffrées contenues dans l'image sont structurées, afin de permettre leur stockage [2].

## 2.2.2 Notion de pixel et espace de couleur :

L'acronyme pixel c'est la contraction de « Picture element », il représenté un point de limage.la clarté ou la définition de limage est proportionnelle aux nombre de pixel, c'est-à-dire la résolution dune image.

Pour une image de bonne qualité la résolution doit être au minimum de 300px/2.54cm (2.54=1pouce).par ex le nombre de pixel dans une image de résolution 640×480=307200 pixels

Chaque pixel est représenté dans un espace de couleur finis. Pour une image de niveau gris, par exemple, chaque pixel est représenté sur une échelle de 256 niveaux de gris ou chaque niveau représente l'intensité de la luminosité du pixel. Cet espace de couleur nécessite 8 bits pour être coder (2^8=256), c'est des images 8 bits.

Typiquement pour un espace de couleur RVB (Rouge, Vert, Bleu), une image chromatique nécessite pour chaque pixel trois couches de niveau de luminosité des trois couches de couleurs RVB. Cela nécessite au total 3*8=24bits pour coder un pixel, ou chaque

couleur de pixel est représentée sur une échelle de prés de 16 mallions de niveaux de couleurs (2^24),

**Problème :** le poids des donnes de limage numérique sont proportionnelles à la résolution de limage.

**Solution :** compresser les données de l'image numérique.

### 1.2.2 Rappel sur la compression des images :

La compression des données ou le codage source, permet on appliquant des algorithmes de compression spécifique de réduire la taille d'une image sur une mémoire ou de manière équivalente de réduire son temps de transmission.

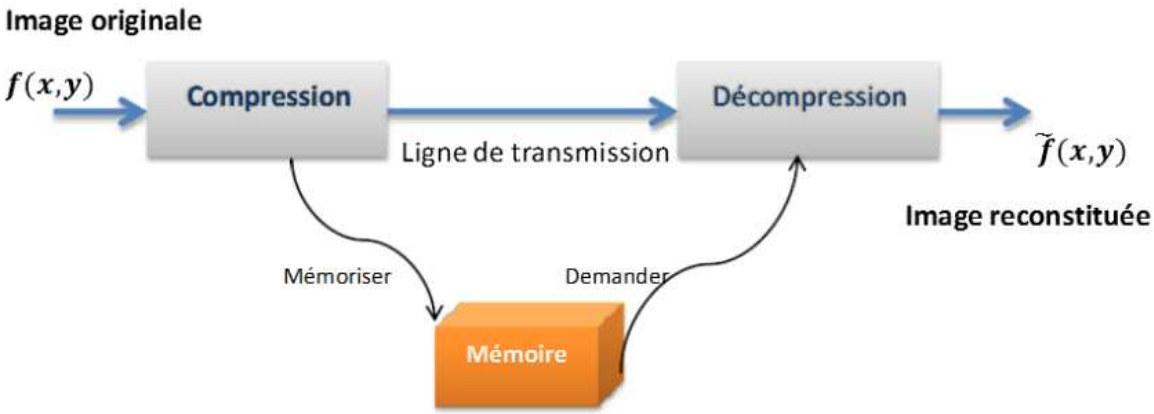



**figure1.3: principe de compression d'image**

La compression peut être sans perte, l'image restera fidele a l'image original, soit elle sera avec perte de qualité (fidélité) pour réduire plus la taille de l'image, dans ce cas-là, la compression sera au prix de la dégradation autorisée, ces type de compression sont faites grâce aux redondances des données présente sur l'image, ces redondance sont :

- **Redondance psycho visuel** : des détails non perceptible a l'œil humain qu'on peut 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 bis, en rencontre généralement a la fin de la compression, pendant l'étape de codage.

### 1.2.3 Les types de compression :

- **Compression sans perte :** 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**.**

Ce type de compression on le trouve beaucoup dans le domaine ou la précision est majeure comme l'image médicale(IRM) ou la télédétection (imagerie satellite). Les algorithmes de compression employée sont nombreux, les plus importants sont :

- Codage à répétition : par ex. RLC (Run Length Coding)

**- Cordage entropique:**

Base sur le codage a longueur variable ou VLC (variable Vength Coding), par ex : le codage de Hufman, le codage arithmétique, …etc.

**-codage dictionnaire au codage Lempel-ziv-Welch (LZW) :**

Ce codage ne nécessite plus de connaitre les probabilités des symboles comme dans le cas du codage entropique.

- **Compression avec perte :**

C'est une compression destructrice, elle permet de sacrifier certaines détails de l'image non récupérable en décompression au profile de réduction de poids. Cette dégradation peut être contrôlée selon la qualité qu'on veut obtenir en fonction du taux de compression choisie.

Ce type de compression on le trouve généralement dans le domaine là ou la réduction du poids de limages est très important, comme le domaine multimédia par exemple (Web, photographie) ou la fidélité envers l'image original n'est pas très importante et le taux de

compression sera plus grand que celui dune compression sans perte du fait qu'on est juste limites par la qualité qu'on souhaite obtenir.

Les algorithmes de compression employés sont nombreux, on cite l'une des méthodes les plus utilisées : le codage par transformation.

Ce type de codage fait appelle aux transformées mathématiques pour avoir une « cartographie » des fréquences spatiale présente dans l'image.

La transforme en cosinus discrète DCT et la transforme en ondelette sont les transformées les plus utilisées.

### 1.2.4 Intérêt 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, car cela demande d'énormes changements dans les infrastructures de télécommunications. Ainsi, pour pallier ce manque, il est courant de réduire la taille des données en exploitant la puissance des processeurs plutôt qu'en augmentant les capacités de stockage et de transmission des données [3].

### 1.2.5 La compression de données

La compression consiste à réduire la taille physique de blocs d'informations. Un compresseur utilise un algorithme qui sert à optimiser les données en utilisant des considérations propres au type de données à compresser ; un décompresseur est donc nécessaire pour reconstruire les données originelles grâce à l'algorithme inverse de celui utilisé pour la compression [4]. La méthode de compression dépend intrinsèquement du type de données à compresser : on ne compressera pas de la même façon une image qu'un fichier audio... [5].

### 1.2.6 Caractérisation de la compression

La compression peut se définir par le quotient de compression, c'est-à-dire le quotient du nombre de bits dans l'image originale par le nombre de bits dans l'image compressée. Le taux de compression, souvent utilisé, est l'inverse du quotient de compression, il est

habituellement exprimé en pourcentage. Enfin le gain de compression, également exprimé en pourcentage, est le complément à 1 du taux de compression [5] : ρ=1-(Taille compressé/Taille normale)

### 1.2.7 La norme de compression JPEG

La norme JPEG (Joint Photographic Experts Group) est conçue par le groupe ISO (International Standards Organisation) et le groupe CEI (Commission Electronic International). Elle est destinée à la compression des images fixes en couleurs et à niveaux de gris en vue de leurs stockages sur les supports numériques [6].

Elle a été réalisée dans la perspective de couvrir les applications les plus diversifiées en tenant compte des contraintes réalistes par rapport aux applications les plus visibles : publication, transmission, banques d'images [7].

Les techniques définies par la norme JPEG se divisent en deux classes : les méthodes de compression avec pertes qui sont basées sur la DCT suivie d'une quantification et d'un codage entropique. La seconde classe, concerne les processus de codage sans pertes, cette classe de codeurs n'est pas basée sur la DCT mais sur le codage MICD suivi d'un codage entropique.

Pour les méthodes avec pertes, quatre codeurs ont été spécifiés :

– Un codage de base où l'image compressée puis décompressée n'est plus identique à l'image originale, ce processus utilise la DCT et un codage de Huffman [8].

– Les trois autres types de codage sont une extension de codage de base. Ils diffèrent de codage de base principalement par le codage entropique en utilisant un codage arithmétique ou par restitution progressive de l'image [10].

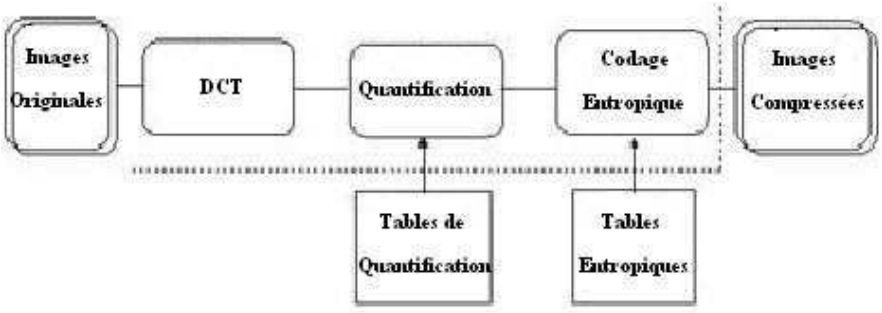



**Fig. 1.4 – Principe de l'algorithme JPEG avec pertes.**

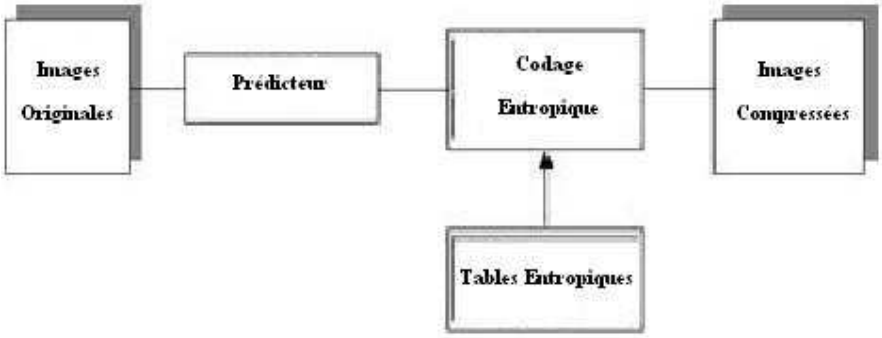



**Fig. 1.5 – Principe de l'algorithme JPEG sans pertes.**

## 1.3. Généralités sur la cryptographie :

### 1.3.1 La cryptographie

Depuis longtemps, la transmission de données sensibles a nécessité l'utilisation d'un système de sécurisation performant.

Les services secrets des grandes puissances économiques et

Politiques, de tout temps très Impliqués, ont développé, tout d'abord, des codages alphabétiques et numériques simples, puis des techniques cryptographiques plus poussées, grâce à l'outil mathématique pour rendre inviolables et inexploitables directement leurs données sensibles.

La cryptologie, véritable science régissant le codage de l'information , a connu une réelle explosion avec le développement des systèmes informatiques, passant d'une ère artisanale et confidentielle à des systèmes de très hautes technologies nécessitant une

importante puissance de calcul. Elle a connu un plus large essor encore avec l'arrivée des systèmes de communications modernes (internet, etc...) où il y a une nécessité absolue de protéger les données échangées pour respecter les individus

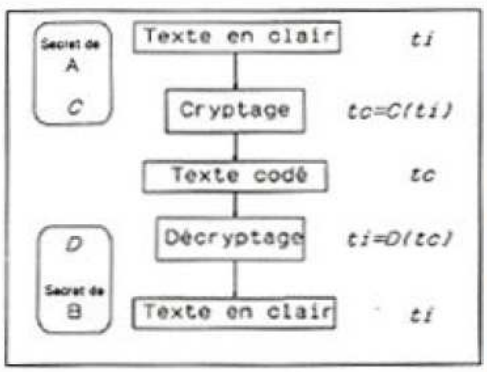
### 1.3.2Principe de la cryptographie

Le cryptage ou chiffrement des données est généralement décrit par une communication secrète d'information entre deux interlocuteurs. Dans un système informatique, cette confidentialité intervient en fait sous plusieurs formes, notamment dans la protection du stockage de l'information (copie de fichier), des accès (interrogation dune base de données) et de sa transmission ("écoute").

Quelque soit son support physique, l'information est toujours représentée par un codage binaire. Ce codage est conforme a un standard qui permet de communiquer avec les dispositifs de la machine (clavier, écrans, imprimantes, traceurs….). Lorsque les ensemble de donnes a stocker ou a transmettre sont très volumineux (en particulier les images), on fait appel a des technique de compression pour optimiser le volume et la vitesse de transmission.

Le cryptage repose sur une transformation du code vers une forme non standard, de façon en limiter l'utilisation.

La technique est très ancienne, mais elle a était largement développée et transformée avec l'utilisation intensive de l'informatique et des codages numériques. Ce domaine reste, pour des raisons évidentes, très secret, et les progrès les plus récents ne sont pas divulgués. Cette présentation se limite évidement au technique et outils du domine public, qui sont ceux effectivement utilisée dans les grandes applications informatiques




Le principe du cryptage est illustré par la figure, L'émetteur de l'information A veut mettre un texte ti sous une forme qui ne soit lisible que par le récepteur B.
Il effectue pour cela un cryptage pour produire le texte t.. Le récepteur B effectuera l'opération inverse (décryptage) pour récupérer le texte sous sa forme initiale. Le système ne peut fonctionner que si A et B ont convenu à l'avance du procédé à utiliser (les fonctions C et D). Il est nécessaire de garantir le secret, c'est à dire de vérifier que les fonctions C (secret de A) et D (secret de B) ne sont pas divulguées, et qu'elles ne peuvent être reconstituées.

**Figure 1.6 principe du cryptage**

Pour limiter les risque de divulgation, on doit pouvoir charger periodiquement de technique de cryptage.

### 1.3.3 Type de cryptographie :

Il existe deux type de cryptographie dans le monde : la cryptographie qui protège vos document de la curiosité de vous amis et celle qui empêche les gouvernements les plus puissant d'accéder a vos fichier.

Donc la cryptographie peut être invulnérable ou vulnérable, comme décrit précédemment. Cette vulnérabilité se mesure en termes de temps et de ressource nécessaire pour récupérer le texte en claire. Une cryptographie invulnérable pourrait être définie comme un texte crypte particulièrement difficile a déchiffrer sans laide d'un outils de décodage approprie. Mais alors, comment déterminer cette difficulté ? Etant donne la puissance informatique et le temps machine actuellement disponible, il devrait être impossible de déchiffrer le résultat dune telle cryptographie avant la fin du monde (même avec un milliard d'ordinateur effectuant un milliard de vérification par second).

### 1.3.4 Les techniques de cryptographie

La cryptologie, science fondamentale qui régit la cryptographie, est essentiellement basée sur l'arithmétique [10].

Ainsi dans le cas d'un texte, il s'agit de transformer les lettres qui composent le message en une succession de chiffres (sous forme de bits dans le cas de l'informatique pour permettre le fonctionnement binaire des ordinateurs), puis ensuite de faire des calculs sur ces chiffres pour :

– d'une part les modifier et les rendre incompréhensibles. Le résultat de cette modification

(Le message chiffré) est appelé cryptogramme,

– d'autre part, faire en sorte que le destinataire sache les déchiffrer en utilisant les outils préétablis ou joints aux données.

Le fait de coder un message de façon à le rendre secret s'appelle chiffrement. La méthode inverse consistant à retrouver le message original, est appelée déchiffrement.

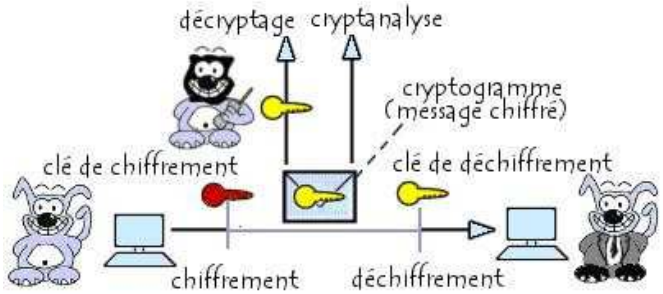



**Fig. 1.7 – Schéma de chiffrement et déchiffrement**.

Le chiffrement se fait généralement à l'aide d'une clef de chiffrement, le déchiffrement avec une clef de déchiffrement. On distingue généralement deux types de clefs :

– Les clés symétriques : on utilise des clés identiques à la fois pour le chiffrement et pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète. Il s'agit de la cryptographie à clé privée.

– Les clés asymétriques : on utilise de clés différentes pour le chiffrement et le déchiffrement. On parle alors de chiffrement asymétrique. Il s'agit de la cryptographie à clé publique.

Au cours des années soixante dix, un système de sécurisation basé sur la polarisation des photons est apparu : la cryptographie quantique. Cette technique est différente des autres crypto systèmes à clé puisqu'elle fonctionne sur des propriétés physiques intrinsèques au système [11].

### 1.3.4.1 La cryptographie à clé privée

Le chiffrement à clé privée, aussi appelé chiffrement symétrique ou chiffrement à clé secrète, consiste à utiliser la même clé pour le chiffrement et le déchiffrement [12].

Si A veut envoyer un message à B, tous deux doivent au préalable s'être transmis la clé. Celle-ci est identique chez l'émetteur et le destinataire du message. Les deux parties doivent se communiquer la clé à un moment ou à un autre, ce qui constitue un risque non négligeable d'interception. Elle peut servir pour plusieurs messages ou être modifiée à chaque échange. Dans le premier cas, elle repose sur la confiance en l'utilisateur [13].

Les systèmes à clé privée posent un second problème. Si une clé différente est mise en œuvre pour chaque paire d'utilisateurs du réseau, le nombre total des clés augmente beaucoup plus rapidement que celui de protagonistes. t


**Fig. 1.8 – Schéma de chiffrement à clé symétrique.**

Dans les années 20, Gilbert Vernam et Joseph Marlogne mettent au point la méthode du one time pad (méthode du masque jetable) , basée sur une clé privée générée aléatoirement, utilisée une et une seule fois puis détruite. Plus tard, le Kremlin et la Maison Blanche sont reliés par le fameux téléphone rouge, dont les communications étaient cryptées par une clé privée selon la méthode du masque jetable. La clé était alors échangée au moyen de la valise diplomatique (jouant le rôle de canal sécurisé)[12].

Dans les années 80, Claude Shannon démontra que pour être totalement sûr, les systèmes à clef privée doivent utiliser les clefs d'une longueur au moins égale à celle du message à chiffrer, ce qui pose problème [12].

De plus, le chiffrement symétrique impose d'avoir un canal sécurisé pour l'échange de la clé, ce qui dégrade sérieusement l'intérêt d'un tel système de chiffrement [12].

### 1.3.4.2 La cryptographie à clé publique

La cryptologie moderne est née en 1976 avec l'introduction par deux chercheurs de l'Université de Stanford, Whitfield Diffie et Martin Hellman, du concept de clé publique [14].

Le principe émet que seule l'opération de déchiffrement doit être protégée par une clé gardée secrète. Le chiffrement peut parfaitement être exécuté à l'aide d'une clé connue publiquement, à condition, bien sûr, qu'il soit virtuellement impossible d'en déduire la valeur de la clé secrète. On parle alors de " cryptographie asymétrique ". Les deux inventeurs butent cependant sur la difficulté de proposer un véritable cryptosystème à clé publique ; la solution vient du MIT en 1978, avec la publication d'un procédé de chiffrement mettant en œuvre les idées de Diffie et Hellman [14].

Ils constatent que la clé publique permet le transport des clés conventionnelles, qui ne repose pas sur l'existence d'une hiérarchie cloisonnée. C'est bien ainsi que fonctionne le système actuellement. Ils savent également qu'un système de chiffrement peut être utilisé comme mode d'authentification : c'est le principe de l'I.F.F. (Identification Friends and Foes), mis au point dans les années 1950 par l'armée de l'air américaine, qui identifie les appareils amis par leur capacité à déchiffrer un message choisi au hasard et inclus dans le signal radar. Dans le contexte de la clé publique, pouvoir déchiffrer un message produit la

preuve qu'on est en possession de la clé secrète. Contrairement au mode conventionnel, cette preuve est opposable aux tiers, puisque qui- conque peut vérifier par chiffrement public qu'on restitue le message initial. On réalise l'analogue d'une signature manuscrite liant un document à son auteur. C'est précisément ce mécanisme de signature numérique qui se met en place aujourd'hui pour les besoins du commerce électronique [15].

Au-delà de l'invention de la clé publique, l'un des apports de la cryptologie moderne est d'avoir su fournir un cadre conceptuel cohérent pour analyser qualitativement les menaces potentielles contre un système cryptographique. La sécurité est algorithmique : elle fait l'hypothèse que l'adversaire éventuel dispose d'une puissance de calcul importante mais bornée ; ceci est contraire à la théorie de Shannon qui attribue à l'ennemi une capacité infinie de calcul et conduit de ce fait à ce qu'on appelle la " sécurité inconditionnelle ". Cette dernière mène à des systèmes peu utilisés puisque la clé a nécessairement une longueur au moins égale au texte à chiffrer. Elle est toutefois parfaitement réalisable par combinaison du texte clair -supposé d'une suite de bits (c'est-à-dire 0 et 1) avec une autre suite constituant la clé, la combinaison étant réalisée par une addition de bits à bits, analogue à l'addition ordinaire, à ceci près que 1+1 vaut 0. Ce mécanisme, connu sous le nom de " chiffrement de Vernam ", est parfaitement sûr lorsque chaque clé n'est utilisée qu'une seule fois. On peut imaginer d'autres mécanismes de sécurité qui ne soient ni algorithmiques ni inconditionnels ; c'est ainsi qu'on envisage aujourd'hui la possibilité de procédés de cryptographie sur les lois de la physique quantique [14].

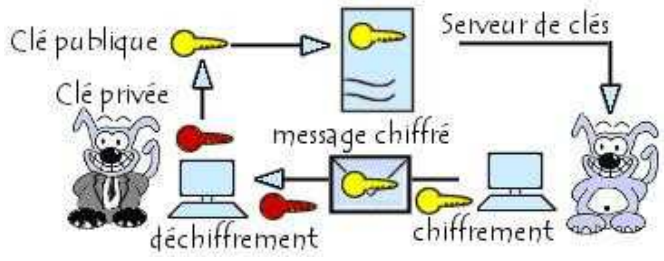



**Fig. 1.9 – Schéma de chiffrement à clé publique.**

### 1.3.5 Efficacité des algorithmes de cryptage

Un procède est décrit par des algorithmes de cryptage et de décryptage, réalisés Par des programmes. Pour être utilisable, un procède de cryptage ne doit pas être trop élaborer : les algorithmes de codage et de décodage doivent être suffisamment rapide don peu complexes, pour ne pas ralentir excessivement leur exploitation. Inversement, pour garantir le secret du cryptage, il faut que celui-ci soit difficile. Cela suppose d'une autre part que le secret des clés soit bien garde, d'autre part que les algorithmes de découverte des clés soit suffisamment complexe, donc suffisamment lents, pour décourager les recherches. La qualité d'un cryptage est donc essentiellement liée au temps d'utilisation (qui doit être court), et au temps de découverte, qui doit être long.

Le choix du cryptage doit tenir compte des intentions et des moyens de l'adversaire, et des risques encourus. Le déchiffrage dune donnée secrète donne accès a des informations confidentielles, la connaissance des clés de cryptage permet d'altérer des informations protégées ou d'en émettre de fausses (brouillage). La découverte d'un cryptage seras liée au temps disponible et a la puissance des ordinateur utilises par l'espion. La fréquence de chargement des clés doit évidement tenir compte de ces donnes.

Les agressions ne sont pas toujours malveillante, les donnes détruites ou divulguées sont le plus souvent le résultat d'erreur ou de négligence contre les quelles il faut aussi se protéger.

Les deux principaux procédés de cryptage connus sont les cryptages symétriques et asymétriques et les cryptages a clé publique (ou asymétrique) [16].

## 1.4 Discussion :

La compression des données est appelée à prendre un rôle encore plus important en raison du développement des réseaux et du multimédia. Son importance est surtout due au décalage qui existe entre les possibilités matérielles des dispositifs que nous utilisons (débits sur Internet, sur Numéris et sur les divers câbles, capacité des mémoires de masse,...) et les besoins qu'expriment les utilisateurs (visiophonie, vidéo plein écran, transfert de quantités d'information toujours plus importantes dans des délais toujours plus brefs). Quand ce décalage n'existe pas, ce qui est rare, la compression permet de toute façon des économies. Les méthodes de compression déjà utilisées couramment sont efficaces et sophistiquées (Huffman, LZW, JPEG) et utilisent des théories assez complexes, les méthodes émergentes sont prometteuses (DCT, ondelettes) mais nous sommes loin d'avoir épuisé toutes les pistes de recherche. Les applications de transmission de données font partie des domaines où la sécurité est critique, dans la mesure où les informations traitées concernent la vie humaine. Les risques sur la fiabilité des données interviennent aussi bien à la création (certification d'une donnée) que lors des stockages et de la transmission, où il convient de les protéger contre les modifications, qu'elles soient accidentelles ou dues à la malveillance. En permettant l'authentification et la signature, le cryptage est l'un des moyens d'assurer la fiabilité des informations.

Le deuxième aspect concerne la confidentialité des informations du dossier de communication, mais aussi de certaines informations globales, car les risques d'indiscrétion peuvent aussi bien porter sur des informations individuelles que globales (vol d'un fichier pour des motifs économiques).

Le troisième aspect concerne la construction des algorithmes de cryptage car les algorithmes de codage et de décodage doivent être suffisamment rapide don peu complexes, pour ne pas ralentir excessivement leur exploitation. Inversement, pour garantir le secret du cryptage, il faut que celui-ci soit difficile. Cela suppose d'une autre part que le secret des clés soit bien garde, d'autre part que les algorithmes de découverte des clés soit suffisamment complexe, donc suffisamment lents, pour décourager les recherches. La qualité d'un cryptage est donc essentiellement liée au temps d'utilisation (qui doit être court), et au temps de découverte, qui doit être long.

# Chapitre II

## 2.1 Préambule :

La puissance des processeurs augmente plus rapidement que les capacités de stockage, et énormément plus vite que la bande passante d'Internet qui demande beaucoup de changements d'infrastructures telles que les installations téléphoniques. Ainsi il semble plus simple de réduire la taille des données plutôt que d'augmenter les espaces de stockage et/ou les infrastructures téléphoniques.

La réduction de la taille des données se fait donc par la compression du codage du fichier source. Ainsi pour réduire la taille d'une image, il faut faire appelle au technique de compressions qui élabore un certain nombre de mécanismes base sur des algorithmes de compression tel que le JPEG a fin de réaliser une crypto-compression d'image dune image fixe.

Dans ce chapitre nous allons exploiter les techniques de cryptage et les méthodes de compression de données suivie de quelques applications.

## 2.2 Techniques de cryptage :

Les techniques de cryptage repose sur une transformation du code vers une forme non standard, de façon en limiter l'utilisation.

Ces techniques sont très anciennes, mais elle ont été largement développée et transformée avec l'utilisation intensive de l'informatique et des codages numériques qui vise a garder le secret dans la messagerie

### 2.2.1 La cryptographie à clé privé

Le chiffrement à clé privée, aussi appelé chiffrement symétrique ou chiffrement à clé secrète, consiste à utiliser la même clé pour le chiffrement et le déchiffrement [11].

Si A veut envoyer un message à B, tous deux doivent au préalable s'être transmis la clé. Celle-ci est identique chez l'émetteur et le destinataire du message. Les deux parties doivent se communiquer la clé à un moment ou à un autre, ce qui constitue un risque non négligeable d'interception. Elle peut servir pour plusieurs messages ou être modifiée à chaque échange. Dans le premier cas, elle repose sur la confiance en l'utilisateur [13].

Les systèmes à clé privée posent un second problème. Si une clé différente est mise en œuvre pour chaque paire d'utilisateurs du réseau, le nombre total des clés augmente beaucoup plus rapidement que celui de protagonistes. Dans les années 20, Gilbert Vernam et Joseph Marlogne mettent au point la méthode de l'one time pad (méthode du masque jetable), basée sur une clé privée générée aléatoirement, utilisée une et une seule fois puis détruite. Plus tard, le Kremlin et la Maison Blanche sont reliés par le fameux téléphone rouge, dont les communications étaient cryptées par une clé privée selon la méthode du masque jetable. La clé était alors échangée au moyen de la valise diplomatique (jouant le rôle de canal sécurisé) [12].

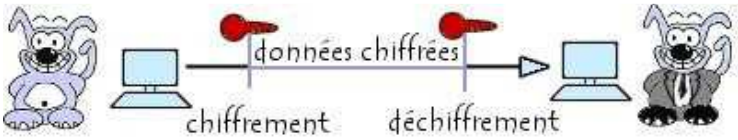



**Fig. 2.1 – Schéma de chiffrement à clé symétrique**.

Dans les années 80, Claude Shannon démontra que pour être totalement sûr, les systèmes à clef privée doivent utiliser les clefs d'une longueur au moins égale à celle du message à chiffrer, ce qui pose problème [12]

De plus, le chiffrement symétrique impose d'avoir un canal sécurisé pour l'échange de la clé, ce qui dégrade sérieusement l'intérêt d'un tel système de chiffrement [12].

### 2.2.2 La cryptographie à clé publique

La cryptologie moderne est née en 1976 avec l'introduction par deux chercheurs de l'Université de Stanford, Whitfield Diffie et Martin Hellman, du concept de clé publique [14].

Le principe émet que seule l'opération de déchiffrement doit être protégée par une clé gardée secrète. Le chiffrement peut parfaitement être exécuté à l'aide d'une clé connue publiquement, à condition, bien sûr, qu'il soit virtuellement impossible d'en déduire la valeur de la clé secrète. On parle alors de " cryptographie asymétrique ". Les deux inventeurs butent cependant sur la difficulté de proposer un véritable cryptosystème à clé publique ; la solution vient du MIT en 1978, avec la publication d'un procédé de chiffrement mettant en œuvre les idées de Diffie et Hellman [14]. Ils constatent que la clé publique permet le transport des clés conventionnelles, qui ne

repose pas sur l'existence d'une hiérarchie cloisonnée. C'est bien ainsi que fonctionne le système actuellement. Ils savent également qu'un système de chiffrement peut être utilisé comme mode d'authentification : c'est le principe de l'I.F.F. (Identification Friends and Foes), mis au point dans les années 1950 par l'armée de l'air américaine, qui identifie les appareils amis par leur capacité à déchiffrer un message choisi au hasard et inclus dans le signal radar. Dans le contexte de la clé publique, pouvoir déchiffrer un message produit la preuve qu'on est en possession de la clé secrète. Contrairement au mode conventionnel, cette preuve est opposable aux tiers, puisque qui- conque peut vérifier par chiffrement public qu'on restitue le message initial. On réalise l'analogue d'une signature manuscrite liant un document à son auteur. C'est précisément ce mécanisme de signature numérique qui se met en place aujourd'hui pour les besoins du commerce électronique [15].

Au-delà de l'invention de la clé publique, l'un des apports de la cryptologie moderne est d'avoir su fournir un cadre conceptuel cohérent pour analyser qualitativement les menaces potentielles contre un système cryptographique. La sécurité est algorithmique : elle fait l'hypothèse que l'adversaire éventuel dispose d'une puissance de calcul importante mais bornée ; ceci est contraire à la théorie de Shannon qui attribue à l'ennemi une capacité infinie de calcul et conduit de ce fait à ce qu'on appelle la " sécurité inconditionnelle ". Cette dernière mène à des systèmes peu utilisés puisque la clé a nécessairement une longueur au moins égale au texte à chiffrer. Elle est toutefois parfaitement réalisable par combinaison du texte clair -supposé d'une suite de bits (c'est-à-dire 0 et 1) avec une autre suite constituant la clé, la combinaison étant réalisée par une addition de bits à bits, analogue à l'addition ordinaire, à ceci près que 1+1 vaut 0. Ce mécanisme, connu sous le nom de " chiffrement de Vernam ", est parfaitement sûr lorsque chaque clé n'est utilisée qu'une seule fois. On peut imaginer d'autres mécanismes de sécurité qui ne soient ni algorithmiques ni inconditionnels ; c'est ainsi qu'on envisage aujourd'hui la possibilité de procédés de cryptographie sur les lois de la physique quantique [14].

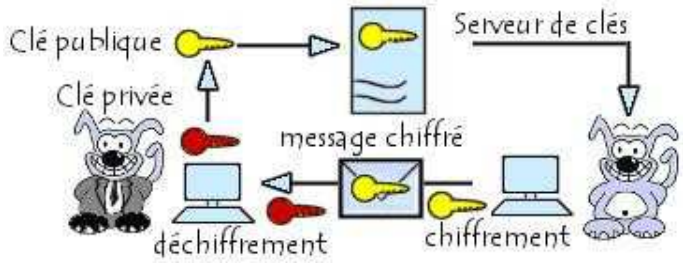



**Fig. 2.2 – Schéma de chiffrement asymétrique.**

### 2.2.3 Quelques applications de la cryptographie

Les banques, la médecine, le militaire, mais aussi de nombreuses entreprises, échangent couramment des informations confidentielles sous la forme de données télématiques par l'intermédiaire d'ordinateurs. Ces données sont en général transmises par le réseau téléphonique ou par d'autres réseaux publics, si bien qu'il convient de mettre au point des cryptages efficaces pour les protéger. En combinant les systèmes de cryptographie évoqués ci-dessus, on peut ainsi créer des chiffres de complexité variée, avec la contrainte que les clés sont elles aussi amenées à être transmises sur ces réseaux [17].

Avec suffisamment de temps et de matériel, on peut résoudre la plupart des codes chiffrés et découvrir ainsi leurs clés. Aussi la complexité du code doit-elle être adaptée afin qu'il soit impossible de le découvrir en un temps raisonnable. Par exemple, des ordres militaires qui ne doivent rester secrets que pendant quelques heures peuvent être cryptés au moyen d'un chiffre qui ne conviendrait pas au codage de rapports diplomatiques exigeant une confidentialité à long terme [17].

Avec ses pages interactives, ses images, ses documents sonores, le réseau Internet a permis le développement d'une forme plus spectaculaire de commerce électronique que celui déjà connu par le minitel. Désormais, les entreprises de vente par correspondance peuvent concevoir des catalogues illustrés sous forme électronique et les achats peuvent s'effectuer au moyen d'une carte de crédit (les jeux téléchargés permettent de faire l'économie du prix de l'emballage).Il existe cependant un obstacle majeur : le seul standard actuel de paiement électronique est la carte bleue. C'est donc ici qu'intervient le cryptage, qui n'est pourtant pas encore légal dans tous les pays. En effet, un problème d'Internet est la question de la sécurité et de la confidentialité. Par nature Internet, étant ouvert à tous, se prête facilement aux piratages de toute nature. Des

logiciels de cryptographie permettent d'assurer une relative confidentialité des échanges [17].

### 2.2.3.1 D.E.S, le chiffrement à clé secrète

**a) Principes :**

C'est un système de chiffrement par blocs de 64 bits uniquement, dont le dernier octet sert de test de parité (pour vérifier l'intégrité des données). Il consiste à faire des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le décryptage). La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés $k_0 k_{15}$. Etant donné que 8 bits de la clé sont réservés pour le test de la parité, " seulement " 56 bits servent réellement à chiffrer, ce qui représente tout de même $2^{56}$ possibilités, c'est-à-dire environ $72 \times 10^{15}$ clés possibles...

Les grandes lignes de l'algorithme sont les suivantes :

DES utilise une clé secrète de 56 bits, qu'il transforme en 16 " sous-clés " de 48 bits chacune. Le cryptage se déroule sur 19 étapes.

La première étape est une transposition fixe (standard) des 64 bits à crypter.

Les 16 étapes suivantes peuvent être divisées en 2 " sous-étapes " chacune. Dans un premier temps, le bloc de 64 bits est découpé en $2 \times 32$ bits, et une substitution est effectuée entre ces deux blocs. En fait, ces deux blocs seront tout simplement échangés l'un avec l'autre. Dans un second temps, le bloc de 32 bits ayant le poids le plus fort (le bloc qui va du bit n°32 au bit n°63) subira une transposition contrôlée par la sous-clé correspondant à l'étape en cours [14].

Les étapes 18 et 19 sont deux transpositions.

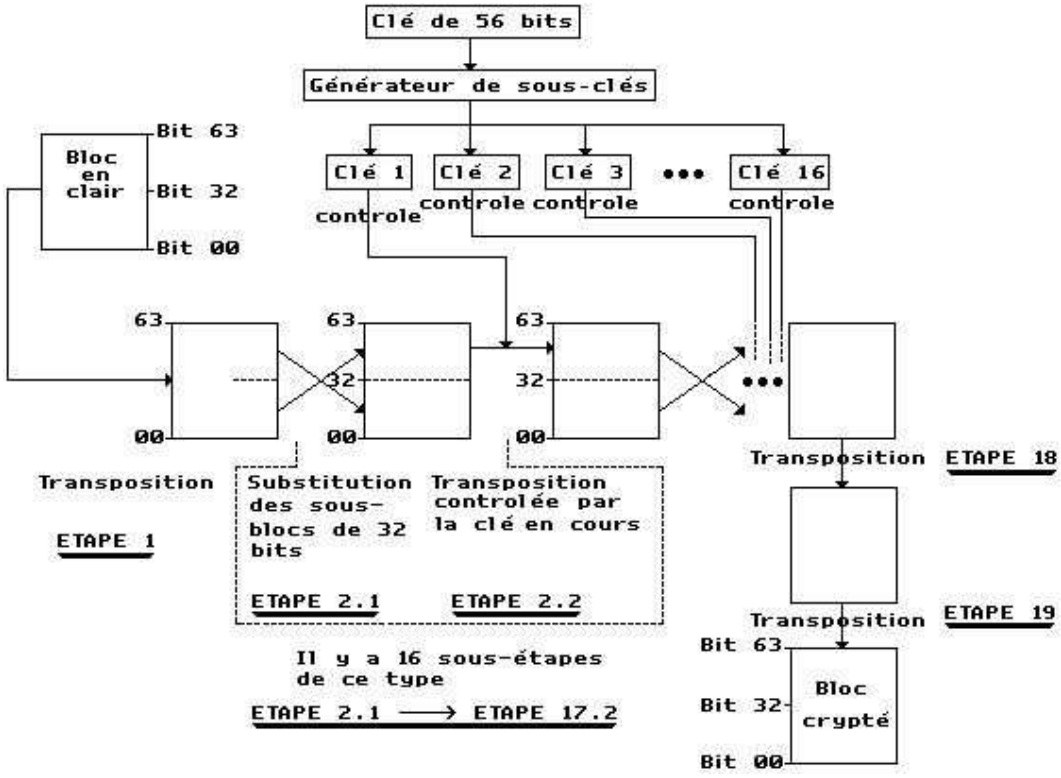



**Fig. 2.3 – Schéma représentant l'algorithme D.E.S.**

## b) Le décryptage avec l'algorithme D.E.S.

Pour décrypter un document auparavant crypté avec D.E.S., il suffit d'effectuer l'algorithme à l'envers avec la bonne clé. En effet, il n'est pas nécessaire d'utiliser un algorithme différent ou une clé différente puisque D.E.S. est comme nous l'avons vu un algorithme symétrique. Il est donc totalement et facilement réversible, si l'on possède la clé secrète.

- **Problèmes de la méthode**

En 1990 Eli Biham et Adi Shamir ont mis au point la cryptanalyse différentielle qui recherche des paires de texte en clair et des paires de texte chiffrées (cette méthode marche jusqu'à un nombre de rondes inférieur à 15 d'où un nombre de 16 rondes dans l'algorithme présenté ci- dessus).

D'autre part, même si une clé de 56 bits donne un nombre énorme de possibilités, des processeurs permettent de calculer plus de $10^6$ clés par seconde. Ainsi, s'ils sont utilisés parallèlement sur un très grand nombre de machines, il peut être possible à un grand organisme (un Etat par exemple) de trouver la bonne clé.

### c) Les modes opérationnels utilisés avec D.E.S.

Comme nous l'avons vu, l'algorithme D.E.S. ne permet que de crypter des blocs de 64 bits. Pour crypter ou décrypter un document complet, il faut donc utiliser D.E.S. en série dans un " mode opérationnel ". Il existe beaucoup de modes opérationnels, nous n'allons voir que le mode ECB et le mode CBC [15].

### d) Le mode opérationnel E.C.B.

ECB signifie Electronic Code Book (" catalogue électronique de codes "). Dans ce mode, on découpe le document à crypter ou à décrypter en blocs de 64 bits qu'on crypte les uns indépendamment des autres. Puisque, à chaque bloc en clair correspond un bloc crypté, pour une clé donnée, cela peut faire penser à un " catalogue de codes ".

### e) Le mode opérationnel C.B.C.

CBC signifie Chain Block Cipher (" Cryptogramme à blocs chaînés "). Comme nous l'avons vu précédemment, le mode opérationnel ECB ne protège pas contre la présence de blocs redondants, puisqu'ils sont cryptés indépendamment les uns des autres. La seconde faiblesse est qu'un bloc en clair, hors contexte, et codé toujours avec la même clé, produira toujours le même bloc crypté.

Le CBC lui, répond à ces deux problèmes. Pour ce faire, avant de crypter un bloc en clair, on va effectuer un " ou-exclusif " entre ce bloc en clair et le bloc précédemment crypté. Cela nous donnera un nouveau bloc en clair que l'on cryptera.

En plus de posséder une clé secrète en commun, les deux interlocuteurs doivent dorénavant se mettre d'accord sur un bloc de 64 bits de départ qu'on appellera " vecteur de départ ", ou " vecteur initial " [15].

### ➢ Procédé PKC (Public Key Cryptosystem).

Bien que moins performant que le DES, il élimine cependant le problème de distribution des clés en utilisant à la fois une clé de chiffrage publique, transmise sans cryptage, et une clé de déchiffrage privée qui n'est accessible qu'au destinataire du message. Il est ainsi possible d'assurer la confidentialité de la transmission tout en authentifiant l'émetteur du message. Il s'agit donc d'une signature électronique,

permettant par exemple la réalisation de transactions commerciales sur un réseau public, notamment sur Internet. La plupart des PKC sont fondés sur les propriétés mathématiques des nombres premiers. Les systèmes de cartes bancaires à puce, qui authentifient leur possesseur par un code secret, sont fondés sur les mêmes principes [18].

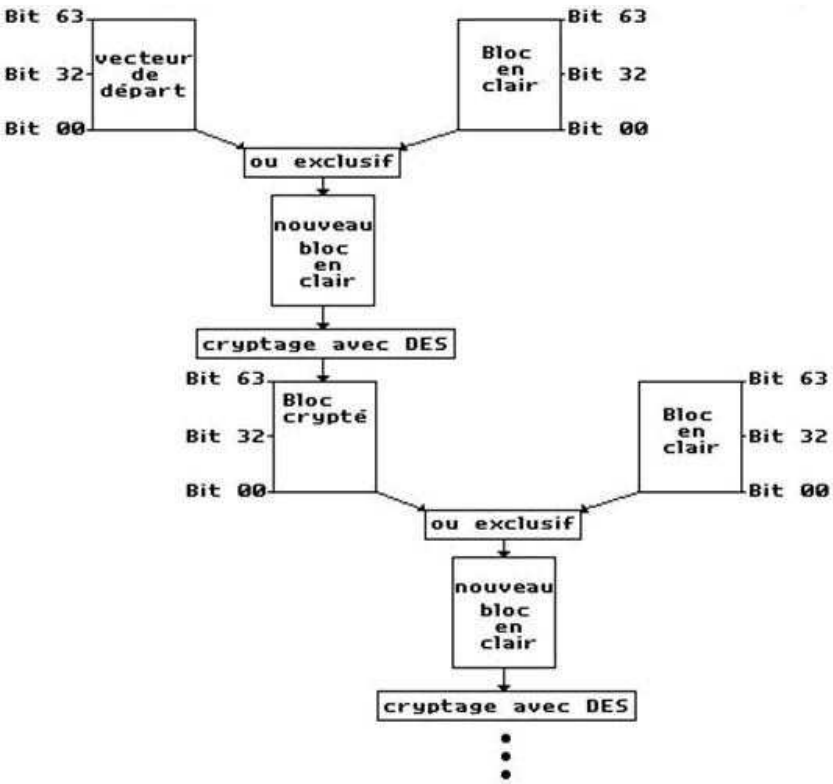



**Fig. 2.4 – Schéma représentant le mode opérationnel C.B.C.**

## 2.2.3.2 R.S.A.

### a) Le système R.S.A.

En 1978, l'algorithme à clé publique de Ron Rivest, Adi Shamir et Leonard Adelman (RSA) apparaît. Il servait encore à l'aube de l'an 2000 à protéger les codes nucléaires de l'armée américaine et soviétique.

### b) Fonctionnement de R.S.A.

Le fonctionnement du crypto système RSA est basé sur la difficulté de factoriser deux entiers. Pour commencer, il nous faut donc choisir deux nombres premiers p et q très grands (de l'ordre de 200 chiffres). Il y a des algorithmes de génération aléatoire de nombres premiers qui existent. Ensuite on trouve le nombre n facilement : $n = p.q$. Puis il nous faut trouver un entier e compris entre 2 et $\phi(n).\phi(n)$ est la fonction indicatrice

d'Euler, c'est en fait le nombre d'entiers inférieurs à n qui sont premiers avec lui, on a $\phi(n) = (p - 1)(q - 1)$.$\phi(n)$ se calcule très facilement ici, puisque l'on a p et q. Maintenant que l'on a n et e, nous sommes prêts à crypter. Les nombres n et e forment ici notre clé publique que l'on notera [n, e]. Il nous faut calculer le nombre d qui sera nécessaire au décryptage. Selon la théorie de RSA, nous devons avoir d tel que (e.d − 1) soit divisible par $\phi(n)$. Comme e et $\phi(n)$ sont premiers entre eux, le théorème de Bezout prouve qu'il existe d et k dans Z tel que $e.d + k.\phi(n) = 1$ [14].

On pourra résoudre l'équation grâce à l'algorithme d'Euclide. Après résolution, on arrivera à une classe de solution de la forme $d = r.\phi(n) + d_0$ (où r appartient à Z) puisque e a été choisi premier avec $\phi(n)$. L'ensemble des solutions d à l'équation diophantienne $e.d + k.\phi(n) = 1$ est une classe de congruence modulo $\phi(n)$, il y a donc une unique solution d comprise entre 2 et $\phi(n)$, donc $d = d_0$. Nous voilà prêts à décrypter. Le nombre d est notre clé privée.

Nous pouvons à présent rendre publique notre clé publique [n, e] et garder secrète notre clé privée. Quant aux nombres p, q, et $\phi(n)$, on doit, soit les conserver secrets, soit les détruire car ils ne serviront plus [14].

## c) L'algorithme R.S.A.

Le brevet de cet algorithme appartient à la société américaine RSA Data Security, qui fait maintenant partie de Security Dynamics et aux Public Key Parteners, (PKP à Sunnyvale, Ca- lifornie, Etats-Unis) qui possèdent les droits en général sur les algorithmes à clé publique. RSA est un algorithme à clé publique qui sert aussi bien à la cryptographie de documents, qu'à l'au- thentification. Grâce au fait qu'il était à clé publique, et au fait qu'il était très sûr, l'algorithme RSA est devenu un standard de facto dans le monde.

## d) Le cryptage avec l'algorithme R.S.A.

Pour crypter un document que l'on aura auparavant transformé en un nombre m inférieur à n il nous faut effectuer l'opération $c = m^e \mathrm{mod} n$. c est ici notre nombre n une fois crypté. La première opération peut être très longue à effectuer à la main, l'utilisation d'un ordinateur et d'un programme spécial est fortement conseillée.

### f) Le décryptage avec l'algorithme R.S.A.

Pour décrypter un document c, il nous faut effectuer l'opération $m = c^d \bmod n$. m sera bel et bien notre nombre décrypté, qu'il ne restera plus qu'à retransformer en texte ou en autre chose. La preuve de cet algorithme de chiffrement est faite avec le théorème de Fermat et le théorème chinois des restes connus depuis quelques siècles !

### g) L'authentification de documents

L'authentification d'un document, c'est le fait d'être sûr de l'identité de l'auteur d'un document. Cette authentification peut s'avérer indispensable pour la justice lors d'un litige sur un contrat par exemple. L'authentification se fait toujours sur un contrat papier par une signature manuscrite, à priori infalsifiable. Le problème de l'authentification d'un document " informa- tique ", est l'impossibilité physique d'y apposer une signature manuscrite à sa fin. On va donc y apposer une signature " digitale ". Pour ne pas être falsifiable, on va crypter cette signature par exemple avec l'algorithme RSA [14].

## 2.3 Méthodes de compression de données

La plupart des méthodes de compression visent à enlever la redondance présente dans l'image de manière à diminuer le nombre de bits nécessaires à sa représentation.

Plusieurs types de redondance en termes de corrélation peuvent être considérés :

- La redondance spatiale entre pixels ou blocs voisins dans l'image ;
- La redondance temporelle entre images successives dans une séquence vidéo ;
- Les méthodes de compression peuvent se regrouper, en deux classes :
- Les méthodes sans perte d'information (sans distorsion) .
- Les méthodes avec perte d'information (avec distorsion).

### 2.3.1 Méthodes sans distorsion des données (sans perte)

Elles permettent de retrouver exactement les pixels de l'image numérique originale.

Voici quelque méthode de distorsion de données sans perte :

### 2.3.1.1Méthodes statistiques (entropiques)

Le principe est d'associer à chaque pixel de l'image un mot de code dont la longueur dépend de la probabilité d'apparition du niveau de gris correspondant.

Pour obtenir un codage efficace, il suffit d'associer les mots de code les plus courts aux niveaux de gris ayant les plus fortes probabilités d'apparition et inversement pour les niveaux présentant une faible probabilité.

Les méthodes statistiques sont aussi connues sous le nom de méthodes VLC (Variable Length Code). Elles peuvent être combinées avec d'autres schémas de codage telles que les méthodes par transformée ou prédictives [19].

- **Codage de Shannon-Fano :**

C.Shannon du laboratoire Bells et R. M. Fano du MIT ont développé à peu près en même temps une méthode de codage basée sur de simples connaissances de la probabilité d'occurrence de chaque symbole dans le message [20].

Le procédé de Shannon-Fano construit un arbre descendant à partir de la racine, par divisions successives. Le classement des fréquences se fait par ordre décroissant, ce qui suppose une première lecture du fichier et la sauvegarde de l'en-tête [21].

Le principe est le suivant :

1. Classer les n fréquences non nulles fi par ordre décroissant.
2. Répartir la table des fréquences en deux sous tables de fréquences proches. Poursuivre l'arborescence jusqu'à ce que toutes les fréquences soient isolées.
3. Attribuer dans l'arborescence le bit 0 à chaque première sous table et le bit 1 à la deuxième sous table correspondante.
4. Attribuer à chaque symbole le code binaire correspondant aux bits de sa description sur l'arborescence.

- **Codage de Huffman :**

D.A. Huffman a inventé en 1952, un algorithme de compression capable, à partir d'une analyse statistique des données, d'associer à celles les plus souvent présentes les

codes les plus courts. Inversement, les données les plus rares se verront attribuer les codes les plus longs [22].

Cet algorithme permet d'obtenir de bons résultats, mais il faut conserver entre la compression et la décompression, le dictionnaire des codes utilisés (voir l'exemple sur annexes).

Le codage de Huffman crée des codes à longueurs variables sur un nombre entier de bits. L'algorithme considère chaque message à coder comme étant une feuille d'un arbre qui reste à construire. L'idée est d'attribuer aux deux messages de plus faibles probabilités, les mots codés les plus longs. Ces deux mots codés ne se différencient que par leur dernier bit. Contrairement au codage de Shannon-Fano qui part de la racine des feuilles de l'arbre et, par fusions successives, remonte vers la racine [21].

Le principe est le suivant :

1. Répartir les fréquences $f_i$ des lettres.
2. Classer les symboles dans l'ordre décroissant des fréquences d'occurrence. Le résultat de l'algorithme ne change donc pas si l'on remplace les fréquences $f_i$ par les probabilités $P_i = f_i / \sum f_i$.
3. Regrouper par séquences les paires de symboles de plus faible probabilité, en les reclassant si nécessaire. Plus précisément : calculer $s = f(i_n) + f(i_{n-1})$, la somme des deux plus faibles fréquences.
4. Choisir le plus petit indice k tel que s soit supérieur ou égal à $f(i_k)$, remplacer k par k+1.
5. Recomposer la table des fréquences en plaçant à la $k^{me}$ position la valeur s et en décalant les autres d'une position vers le bas. Puis décrémenter n d'une unité, poursuivre jusqu'à ce que la table des fréquences ne comporte plus que deux éléments.
6. Coder avec retour arrière depuis le dernier groupe, en ajoutant un 0 ou un 1 pour différencier les symboles préalablement regroupés.

### ➢ Codage arithmétique :

Contrairement aux algorithmes de Huffman et de Shannon-Fano qui associent à des symboles des motifs binaires dont la taille dépend de leur distribution. Le codeur arithmétique traite le fichier dans son ensemble, en lui associant un unique nombre décimal rationnel.

Ce nombre compris entre 0 et 1, possède d'autant moins de chiffres après la virgule que le fichier dont il est redondant. Ces chiffres décimaux dépendent non seulement des symboles du fichier dans l'ordre où ils apparaissent, mais aussi de leur distribution statistique [21].

### ➢ Algorithme du codage arithmétique

1. Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
2. Associer à chaque symbole un sous intervalle proportionnel à sa probabilité, dans l'intervalle [0,1] (l'ordre de rangement des intervalles sera mémorisé car il est nécessaire au décodeur).
3. Initialiser la limite inférieure de l'intervalle de travail à la valeur 0 et la limite supérieure à la valeur 1.
4. Tant qu'il reste un symbole dans la chaîne à coder :
   - largeur = limite supérieure - limite inférieure,
   - limite inférieure = limite inférieure + largeur×(limite basse du sous intervalle du symbole),
   - limite supérieure = limite inférieure + largeur×(limite haute du sous intervalle du symbole),
5. La limite inférieure code la chaîne de manière unique.

## 2.3.1.2 Méthodes Lempel Ziv

Les algorithmes de compression et décompression LZ ont des fonctionnement symétriques : leur principe est fondé sur l'indexation de chaînes dans un dictionnaire qui, dans les deux cas, est construit durant le traitement.

Le dictionnaire est défini comme un tableau de chaînes de tailles variables, repérées par leur adresse ; la taille du tableau est également variable, limitée par le mode de codage des adresses.

Cette méthode, plus performante que celles utilisées auparavant, souffre cependant de quelques difficultés. Sa programmation est assez complexe (gestion de pointeurs " glissants " sur des fenêtres, gestion de tableaux de longueur variable composés d'objets de longueur variable), et le programme peut conduire à des délais de traitement longs, notamment en compression. Le traitement s'effectuant sur une fenêtre, c'est sa taille qui détermine les performances du dispositif : lors du codage, les chaînes du tampon de lecture sont comparées à toutes les positions dans la fenêtre, et une taille réduite conduit à ne pas prendre en compte des chaînes répétées au delà de cette distance ; en revanche, une taille plus importante impose des traitements considérables, en multipliant le nombre de comparaisons nécessaires. De plus, s'il n'y a pas de correspondance, les caractères doivent être transmis individuellement, avec une longueur de séquence à zéro, ce qui peut conduire à une augmentation de la taille des données.

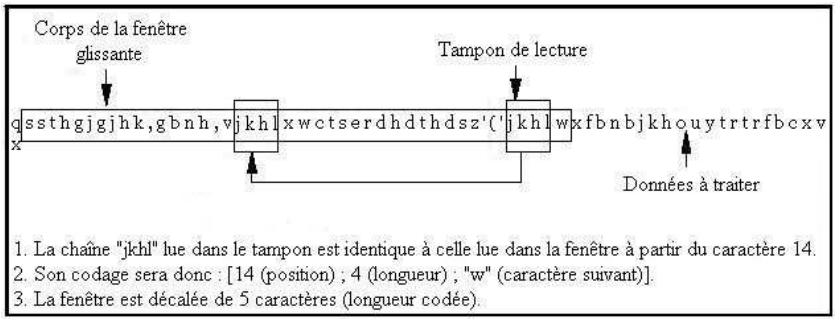



**Fig. 2 .5 – Principe de fonctionnement de LZ77.**

En 1978, Lempel et Ziv ont publié une nouvelle version de leur algorithme [22], abandonnant le concept de fenêtre glissante. Le dictionnaire se construit dynamiquement, tout au long du traitement. Chaque signe transmis par le codeur comprend un index dans le dictionnaire, ainsi que le caractère suivant à coder. La longueur de la chaîne n'a plus à être transmise, puisqu'elle est conservée dans le dictionnaire ; la concaténation de la chaîne répétée avec le caractère suivant est également placée dans le dictionnaire, et devient ainsi disponible pour la totalité du codage ultérieur. Cette nouvelle version permet de dépasser les limites de LZ77, et améliore largement les performances, surtout sur les fichiers longs. Cependant, une nouvelle difficulté apparaît : la taille du dictionnaire est limitée par le mode de codage de ses index (souvent sur 16 bits, soit 65 536 entrées), et il est donc nécessaire de gérer l'évènement " dictionnaire plein ". Une possibilité est de laisser le

dictionnaire en l'état, et de simplement l'utiliser, mais cela conduit à une détérioration des performances dans le cas de longs fichiers, qui peuvent présenter de grands changements dans la nature des données ; une autre solution est de supprimer le dictionnaire existant, et d'en reconstruire un autre, mais, dans le cas de données homogènes, cela peut conduire à une détérioration des performances. La solution généralement retenue est d'analyser le taux de compression obtenue avec le dictionnaire existant, et de le reconstruire seulement si on constate une détérioration.

Terry Welch (le " W " de LZW) a publié en 1984 de nouvelles améliorations [23]. Contraire- ment au dictionnaire de LZ78, qui ne contient au départ qu'une chaîne vide repérée par l'index " 0 ", celui de LZW comprend au départ les codes ASCII, repérés de 0 à 255 ; tous les symboles peuvent ainsi être directement codés selon le dictionnaire, permettant ainsi d'éviter la détérioration des performances pour les fichiers ne présentant que peu de répétitions. De plus, Welch a également défini les principes de communication entre codeur et décodeur : le caractère auto adaptatif du procédé est amélioré par la possibilité d'adapter dynamiquement les paramètres au cours du codage, les modifications étant communiquées au décodeur selon un codage spécifique. Il s'agit par exemple de l'augmentation de la taille des adresses, de la purge partielle ou totale du dictionnaire, de la création temporaire d'un autre dictionnaire, et plus généralement de tout changement de technique de compression en cours de traitement.

La plupart des compacteurs du marché utilisant LZ et ses variantes (Arj, Pkzip...) exploitent conjointement un algorithme statistique limité (issu de Huffman ou de Shannon-Fano), mais en évitant l'analyse préalable de l'ensemble du fichier, trop pénalisante en termes de vitesse d'exécution. Dans la plupart des cas, l'analyse est limitée à une fenêtre, dont la taille varie en fonction des mesures de représentativité statistique des résultats obtenus. D'autres variantes prévoient d'effectuer le compactage par " paquets " de taille fixe, ce qui permet des codage statistiques faciles, tout en ménageant des possibilités d'accès direct et de traitement " au vol " (modems).

Le plus cité des algorithmes LZ, le LZW (perfectionnements divers apportés par Welch en 84) est sans doute aussi le moins utilisé, car il a fait l'objet d'un brevet déposé par Unisys pour une exploitation directe par le système Unix. Dans la plupart des cas, les algorithmes mis en œuvre par les logiciels actuels sont des évolutions de

LZ77 ou LZ78, exploitant quelques uns des principes mis en place par Welch, comme le codage des caractères ASCII ou l'adaptation de taille des dictionnaires.

## ➢ Performances

Les codages Lempel Ziv sont à ce jour ceux qui permettent les meilleurs taux de compactage non destructif sur la plupart des fichiers. Les possibilités de paramétrage et d'association à d'autres techniques (statistiques notamment) offrent une infinité de variations permettant d'optimiser le rapport entre taux de compression et vitesse de traitement. Les taux obtenus sur un jeu d'essai mettent en évidence des compressions deux fois plus efficaces que celle de Huffman [24]. Les tests effectués par Mark Nelson montrent des gains de compression moyens compris entre45 et 50% pour les différentes variantes [25].

## ➢ Implémentations

De très nombreux programmes de tous types utilisent les algorithmes LZ. Outre les utilitaires spécialisés (Arc, Pkzip, Lharc, Arj), on peut citer les logiciels de sauvegarde (QIC-122 pour les sauvegardes à bandes, PC Backup, Norton Backup, MS Backup, Sytos), les protocoles de transmission haute vitesse par modem (V42bis), ainsi que diverses implémentations intégrées aux systèmes d'exploitation (Compress sous Unix, Dblspace de DOS 6). L'ensemble des compacteurs " temps réel " (Dblspace, Stacker...) utilisent ces techniques. Les variantes mises en œuvre sont souvent issues de LZ77, qui se révèle plus performant sur les fichiers courts (le dictionnaire, et donc l'efficacité de LZ78 et des versions postérieures croissent avec la taille du fichier) ; en outre, il a l'avantage d'être dans le domaine public.

Les algorithmes et exemples donnés sont issus de Plume [26]. Les algorithmes de compactage et décompactage utilisent un dictionnaire de taille limitée (1024 entrées). Les codes sont donc toujours de même taille (sur 10 bits). Certaines variantes utilisent des dictionnaires de taille variable, les codes étant eux-mêmes de longueur croissante au long du fichier. Un code réservé indique alors que le nombre de bits de codage est incrémenté.

## 2.3. 2 Méthodes avec distorsion des données

Ces méthodes permettent de retrouver une approximation de l'image numérique. Les pertes sont généralement indécelables à l'œil nu.

### 2.3.2.1 Quantification vectorielle

Les techniques de compression d'images exploitent généralement la redondance statistique présente dans l'image. La quantification scalaire qui associe à une variable continue une variable discrète pouvant prendre un nombre plus faible, et fini de valeurs. Ces valeurs ne sont jamais totalement décorrélées, ou indépendantes. Shannon a montré qu'il était toujours possible d'améliorer la compression de données en codant des vecteurs plutôt que des scalaires [21].

La Quantification Vectorielle (QV), développée par Gersho et Gray [27] a pris une place très importante dans le domaine de la compression d'image que ce soit dans le but de transmission ou d'archivage.

#### a) Principe de la quantification vectorielle

La quantification vectorielle, dans son sens le plus général, est l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète [28]. Elle peut être vue comme une application Q associant à chaque vecteur d'entrée x de dimension K, un vecteur $y = Q(x)$ de même dimension appartenant à un ensemble fini Y appelé DICTIONNAIRE de taille finie N, $Y = (y_j , j = 1...N)$. Elle se décompose en deux applications : codeur, décodeur [27].

#### ➢ Codeur

Le rôle du codeur consiste, pour tout vecteur x du signal en entrée, à rechercher dans le dictionnaire Y le code vecteur $y_j$ le plus proche du vecteur source x. C'est uniquement l'adresse du code vecteur $y_j$ ainsi sélectionnée qui sera transmise ou stockée. C'est à ce niveau donc que s'effectue la compression.

- **Décodeur**

Il dispose d'une réplique du dictionnaire et consulte celui-ci pour fournir le code vecteur d'indice correspondant à l'adresse reçue. Le décodeur réalise l'opération de décompression.

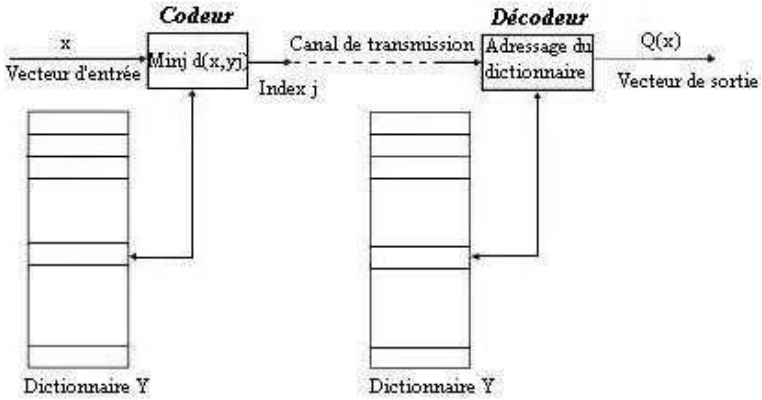



**Fig. 2.6 – Principe du quantificateur vectoriel.**

## 2.3.2.2 Méthodes par transformée

Dans ces méthodes, l'image de dimension NxN est subdivisée en sous images ou blocs de taille réduite (la quantité de calcul demandée pour effectuer la transformation sur l'image entière est très élevée) [23]. Chaque bloc subit une transformation mathématique orthogonale inversible linéaire du domaine spatial vers le domaine fréquentiel, indépendamment des autres blocs (transformée en un ensemble de coefficients plus ou moins indépendants). Les coefficients obtenus sont alors quantifiés et codés en vue de leur transmission ou de leur stockage. Pour retrouver l'intensité des pixels initiaux, on applique sur ces coefficients la transformation inverse [29]. Parmi les transformations linéaires existantes :

- Transformation de Karhunen-Loeve (KLT).
- Transformation de Fourrier discrète (DFT).
- Transformation de Hadamard (HT).
- Transformation en ondelettes (WT).
- Transformation en cosinus discrète (DCT).

Le principe d'un système de codage par transformation est le suivant :

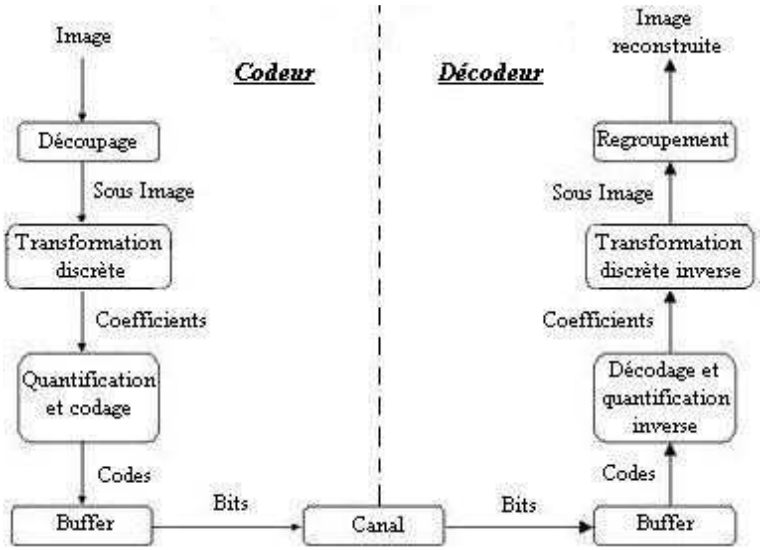



**Fig. 2.7 – Principe d'un système de codage par transformation**.

Exemple de codage par transformation

## ➢ Transformation en Cosinus Discrète (DCT) :

C'est une transformation mathématique qui transforme un ensemble de données d'un domaine spatial en un spectre de fréquence et inversement(IDCT). C'est la plus utilisée parmi les transformations citées.

Elle permet schématiquement de changer l'échelle de mesure, en passant d'une échelle définissant un pixel en fonction de sa position en x et en y à une échelle définissant la fréquence d'apparition de ce pixel dans un bloc de pixels, en effet, il est dés lors possible de supprimer des informations sans pour autant altérer le résultat final, contrairement à un bloc de pixels où la disparition brute de plusieurs éléments est immédiatement visible [29].

La DCT est effectuée sur une matrice carrée N × N de valeurs de pixels et donne une matrice

Carrée N × N de coefficients de fréquence. Le temps de calcul requis pour chaque élément dans la DCT dépend de la taille de la matrice.

Vu la difficulté d'appliquer la DCT sur la matrice entière, celle-ci est décomposée en blocs de taille 8 × 8 pixels.

A la sortie de la matrice de la DCT, la valeur de la position (0,0) est appelée le coefficient continu, cette valeur représente une moyenne de la grandeur d'ensemble de la matrice d'entrée, ce coefficient est plus grand d'un ordre de grandeur à toute valeur dans la matrice de la DCT, par convention, les 64 valeurs transformées (de chaque bloc) sont positionnées d'une certaine manière, ainsi la valeur moyenne de tous ces coefficients est placée en haut à gauche de ce bloc. Plus on s'éloigne des coefficients continus plus leur grandeurs tendent à diminuer. Ce qui signifie que la DCT concentre la représentation de l'image en haut à gauche de la matrice de sortie. Les coefficients en bas et à droite de cette matrice contient moins d'information utile [30].

Les équations qui suivent, donnent respectivement la transformée en cosinus discrète directe et inverse.

## ➢ Transformée Directe

$$F(u, v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x, y)\cos\left[\frac{(2x+1)u\Pi}{2N}\right]\cos\left[\frac{(2y+1)v\Pi}{2N}\right]$$

Pour u, v = 0, 1, 2,...,N-1.

Transformée Inverse

$$f(x, y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)F(u, v)\cos\left[\frac{(2x+1)u\Pi}{2N}\right]\cos\left[\frac{(2y+1)v\Pi}{2N}\right]$$

Pour x, y = 0, 1, 2,...,N-1. Avec :

$$\alpha(u) = \begin{cases} \left(\frac{1}{N}\right)^{\frac{1}{2}} \\ \left(\frac{2}{N}\right)^{\frac{1}{2}} \end{cases}$$

f(x, y) représente une valeur de l'image initiale pour x et y données. F(u,v) représente les coefficients de la DCT. N représente la taille d'un bloc.

La compression JPEG Avec perte (que nous allons voir dans le chapitre qui suit).

## 2.4 Discussion

Les méthodes de cryptage et de compression proposée peuvent intégrer de manière intelligente des techniques de cryptage multi résolution et des techniques de compression existantes (avec ou sans perte). Elle offre à l'utilisateur la possibilité de traiter de façon optimale chaque donnée en fonction de son type et d'obtenir, par conséquent, des taux de compression relativement élevés, une meilleure qualité et de garder une certaine quantité de redondance partiellement cryptée.

# Chapitre III

## 3.1 Préambule

Dans de nombreuses applications : photos satellites, clichés médicaux, photos d'agences de presse, tableaux…, un standard pour archiver ou transmettre une image fixe, en couleur et de bonne qualité est nécessaire. Une première recommandation a été donnée par l'UIT-T en 1980 pour le fac-similé, c'est à dire pour transmettre sur une ligne téléphonique une image en noir et blanc au format A4 (210 x 297 $mm^2$) de l'ISO en environ une minute. La définition est de 4 lignes par mm et de 1780 éléments d'image (pixels, picture éléments) en noir et blanc par ligne. Il y a donc environ 2 Mbits à transmettre. Pendant 1 minute à 4800 bauds (bit par seconde), on transmet environ 300kbits. Le taux de compression doit donc être voisin de 7.

Une image fixe de couleur de qualité télévision réclame de l'ordre de 8 Mbits (640 x 480 x 24). Une image de qualité 35 mm en réclame 10 fois plus. Un effort de standardisation a été effectué : l'association de deux groupes de normalisation, le CCITT et l'ISO (Organisation Internationale de Standardisation), supportée par divers groupes industriels et universitaires, donna naissance au **J.P.E.G.**:(**J**oint **P**hotographique **E**xperts **G**roup). Cette norme comprend des spécifications pour le codage conservatif et non-conservatif. Elle a abouti en 1990 à une première phase d'une recommandation ISO / UIT-T. Les contraintes imposées sont importantes. La qualité de l'image reconstruite doit être excellente, le standard adapté à de nombreuses applications pour bénéficier, entre autre, d'un effet de masse au niveau des circuits VLSI nécessaires, la complexité de l'algorithme de codage raisonnable. Des contraintes relatives aux modes d'opérations ont également été rajoutées. Le balayage est réalisé de gauche vers la droite et de haut en bas. L'encodage est progressif et hiérarchique. Ces deux derniers qualificatifs signifient qu'un premier encodage peut fournir une image reconstruite de qualité médiocre mais que des encodages successifs entraîneront une meilleure résolution. Cela est utile, par exemple, lorsque l'on désire visualiser une image sur un écran de qualité médiocre puis l'imprimer sur une bonne imprimante.

## 3.2 Principe de la compression JPEG

Le principe de l'algorithme JPEG pour une image à niveaux de gris (une image couleur est un ensemble d'images de ce type), est le suivant. Une image est décomposée séquentiellement en blocs de 8x8 pixels subissant le même traitement. Une transformée en cosinus discrète bidimensionnelle est réalisée sur chaque bloc. Les coefficients de la transformée sont ensuite quantifiés uniformément en association avec une table de 64 éléments définissant les pas de quantification. Cette table permet de choisir un pas de quantification important pour certaines composantes jugées peu significatives visuellement, car les informations pertinentes d'une image, caractérisée par son signal bidimensionnel *Img*(*x*, *y*), sont concentrée dans les fréquences spatiales les plus basses. On introduit ainsi un critère perceptif qui peut être rendu dépendant des caractéristiques de l'image et de l'application (taille du document). Une table type est fournie par le standard mais n'est pas imposée.

Un codage entropique, sans distorsion, est enfin réalisé permettant d'utiliser les propriétés statistiques des images. On commence par ordonner les coefficients suivant un balayage en zigzag pour placer d'abord les coefficients correspondant aux fréquences les plus basses. Cela donne une suite de symboles. Le code de Huffman consiste à représenter les symboles les plus probables par des codes comportant un nombre de bits le plus petit possible.


**Figure 3.1_ Principe de la compression JPEG**

## 3. 3 Qu'est-ce qu'une image informatique ?

Une image informatique est constituée de points de couleurs différentes. L'association (point, couleur) est appelée pixel. La mémoire utile pour stocker un pixel peut varier de 1 bit (cas des images monochromes) à 24 bits (images en 16 millions de couleurs)

Les informations sur la luminance (paramètre **Y**) et la chrominance (**I** et **Q**) sont des combinaisons linéaires des intensités de rouge (**R**), vert (**G**), et bleu (**B**) :

***Y = 0.30 R + 0.59 G + 0.11 B I = 0.60 R - 0.28 G - 0.32 B***
***Q = 0.21 R - 0.52 G + 0.31 B***

Soit une image 640x480 RGB 24 bits/pixel. Chacune des ces trois variables est reprise sous forme de matrice 640x480. Cependant, les matrices de **I** et de **Q** (info sur la chrominance) peuvent être réduites à des matrices 320x240 en prenant les moyennes des valeurs des pixels regroupés par carré de quatre. Cela ne nuit pas à la précision des infos sur l'image car les yeux sont moins sensibles aux écarts de couleurs qu'aux différences d'intensités lumineuses. Comme chaque point de chaque matrice est une info codée sur 8 bits, il y a chaque fois 256 niveaux possibles (0-255). En soustrayant 128 à chaque élément, on met à zéro le milieu de la gamme de valeur possible :-128 à +127. Enfin chaque matrice est partagée en blocs de 8x8.

## 3.4 Le schéma général du bloc codec source d'image fixe et le suivant :

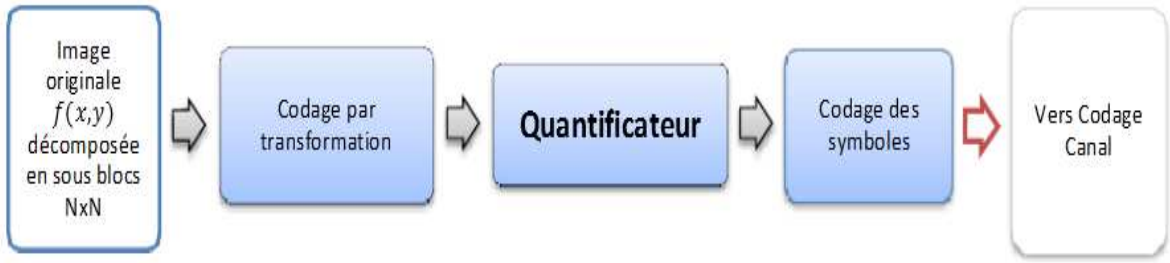



**Figure 3.2 : Figure codage source d'image fixe**

### a) Codage par transformation (Transform Coding):

Après décomposition de l'image en sous-blocs NxN à cause de la non efficacité et la lenteur des calculs d'un seul bloc entier, on applique une transformé linéaire (DFT, DCT,…etc.…) qui permet de transformer les sous blocs NxN de l'image a un format non visuel afin d'obtenir des coefficients dans un domaine fréquentiel (une représentation fréquentiel de l'image) qui seront codés et quantifiés. Le but du codage par transformer est d'avoir un autre aspect de la distribution des données de l'image et ainsi décelé les redondances potentiel comme la redondance inter pixel.

### b) Quantificateur (Quantizer):

Le quantificateur permet de faire une pondération des coefficients de la transformé en se basant sur une table de quantification (une table de luminosité et une autre de chrominance) qui a été établit sur les caractéristiques de la vision humaine, les coefficients de cette table imposeront un seuil de qualité qui décidera des coefficients de la transformé qu'on veut garder et ceux qu'on veut supprimer jugées non important visuellement, d'une autre manière,

cette étape permet de choisir le taux de dégradation visuel qu'on souhaite faire à l'image (facteur de qualité),et le but de tous cela c'est d'éliminer la redondance psycho visuel. Cette étape se trouve seulement dans les compressions avec perte.

### c) Codage des symboles (symbols coding) :

Cette étapes permet de réduire la redondance de codage en utilisant des codes VLC ou autres.

Le décodage source : c'est les mêmes blocs des schémas précèdent mais dans le sens inverse et chaque bloc effectue l'opération inverse de celle au codage.

## 3.5 Notion sur la DCT

Nous allons parler seulement de certains aspects important à connaitre pour l'achèvement de l'objet de ce travail qui est la compression d'image sans entrer dans les démonstrations mathématiques.

### 3.5.1 Définition

La DCT ou discrete cosine transforme, c'est une transformé linéaire qui a été appliquée la 1ere fois dans la publication des professeurs [N.Ahmed, T.Notrojan, and K.R.Rao, "Discrete Cosine Transforme" .ICCC Trans .Computer, janvier 1974].

C'est une variante de la transforme de fourrier discrète, qui permet de garder seulement les cosinus et d'éliminer les sinus, c'est-à-dire, obtenir une représentation fréquentiel purement réel. Cette transforme est très largement utilisée dans la compression audio et de la compression image comme JPEG et MPEG.

### 3.5.2 Variantes de la DCT :

Il existe 8 variables de la DCT, ceux les plus connus sont DCTI, la DCT2D ou DCTll et sa transforme inverse IDCT ou DCTIII qui sont la base de la compression JPEG [68] .

**DCTI :**

Pour une séquence d'élément discrets N, la DCTI s'exprime par :

$$c(u) = a(u) \sum_{x=0}^{N-1} f(x) cos \left| \frac{\pi(2x+1)u}{2N} \right| \quad [\mathbf{1}]$$

$u$ = 0, 1, ….N – 1

$$et: a(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u \neq 0 \end{cases} \quad [\mathbf{2}]$$

**DCTII :**

C'est une extension directe de la DCTI dans deux démentions 2D,N× $N$ , qui seras notre image, sa formule mathématique est donnée par :

$$c(u,v) = a(u)a(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\, cos \left| \frac{\pi(2x+1)u}{2N} \right| cos \left| \frac{\pi(2y+1)v}{2N} \right| \quad [3]$$

$$a(u) et\ a(v) ce\ calcule\ de\ la\ meme\ facon\ que\ dans\ [2]$$

On remarque si :$u = v = 0 \Rightarrow c(u=0, v=0) = \frac{1}{N} \sum_{x=0}^{N-1} f(x,y)$ [4]

Ce coefficient représente le coefficient DC, et le reste des coefficients représente les coefficients AC, nous allons voir après ce que cela signifie en compression JPEG.

**IDCT :**

OU DCTIII, elle est l'inverse de la DCTII, et sa formule est donnée par :

$$f(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} a(u)a(v)c(u,v) \cos \left[ \frac{\pi(2x+1)u}{2N} \right] \left[ \frac{\pi(2y+1)v}{2N} \right] \quad [5]$$

$$u, v = 0,1 \dots \dots \dots N-1$$

## 3.6 Propriétés importantes de la DCT en compression d'image :

### 3.6.1 Décorrélation :

La DCT permet de réduire ou d'éliminer la redondance inter pixel, les pixels sont décorrolés a fin de réduire la variance des pixels voisins dans l'image ce qui permet une compression efficace en codant chaque pixel indépendamment.

### 3.6.1 Concentrations des coefficients :

La DCT est très efficace pour des images fortement corrélée du fait qu'elle permet de compacté les coefficients qui représente les bases fréquences dans une seul partition de la matrice image, cela permet la séparation des fréquences basses des fréquences hautes, et si c'est une image faiblement corrélée, les coefficients sont concentrés dans plusieurs différentes partitions de la matrice image.

### 3.6 .3 symétrie, séparabilité, et orthogonalité :

La DCTII peut être séparée comme suit :

De [3] n aura :

$$c(u,v) = a(u)a(v)\sum_{x=0}^{N-1} cos\left[\frac{\pi(2x+1)u}{2N}\right]\sum_{Y=0}^{N-1}\left[\frac{\pi(2y+1)v}{2N}\right] \quad [6]$$

$$u, v = 0{,}1 \ldots\ldots N-1$$

De [6] on peut sortir la propriété de symétrie qui va nous conduire à une expression simplifiée du calcul de la DCT :

$$F = CfC^T$$

$ou\ f\ est\ la\ matrice\ image\ N \times N\ et\ F\ sa\ transforme\ DCT$

$C\ c'est\ une\ matrice\ N \times N, ces\ element\ c(i,j) se\ calcule\ comme\ suit$:

$$c(i,j) = a(j)\sum_{j=0}^{N-1} COS\left[\frac{\pi(2j+1)i}{2N}\right] = a(j)\sum_{j=0}^{N-1} COS\left[\frac{\pi}{N}i(j+\frac{1}{2}\right] \quad [8]$$

$a(j) est\ calculé\ par$: [1]

A ce stade, on peut conclure aves la propriété d'orthogonalité : $C^{-1} = C^T$ [9]

[9] va-nous permettre d'avoir l'IDCT de la forme suivante :

$$f = C^{-1}FC^{-1} = C^{-1}CfC^TC^{-1} = f \text{ [10]}$$

On remarque qu'on obtient la matrice image original au final. Ces propriétés réduisent considérablement les calculs dans les algorithmes de compression d'image du fait que ces éléments seront déjà calculés à l' avance (calcule de la matrice$C$).

## 3.7 Processus générale d'un codec JPEG d'image fixe :

### 3 .7.1 Définition :

**JPEG :** Joint Photographique Experts Group

C'est un algorithme standard de compression avec perte d'image fixe établie en 1991 base sur le codage par transformation en cosinus discrète DCT, et il existe plusieurs modes opératoires de compression JPEG établies selon la rapidité du codec et le besoin

- Codage base sur la DCT séquentielle
- Codage base sur la DCT progressive
- Codage sans perte
- Codage hiérarchique

### 3.7.2 JPEG basée sur la DCT séquentielle :

Celui que nous allons aborder sur ce travail c'est celui base sur la DCT séquentielle, ou l'image est traitée de gauche à droite et de haut en bas, la figure suivante nous montre le processus d'un codec source JPEG.

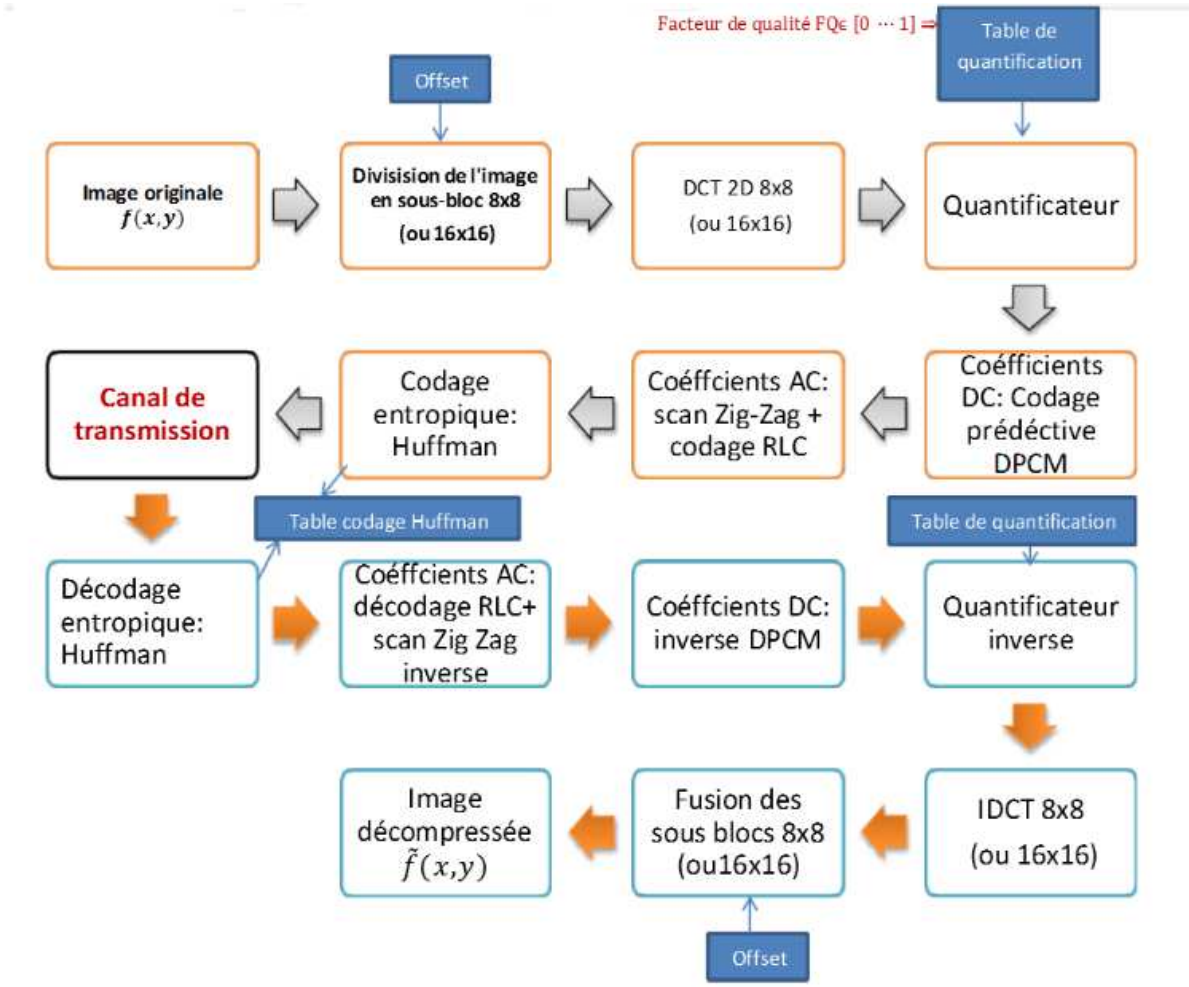



**Figure3.3 Codec source JPEG**

## 3.7.2 .1 Découpage en sous blocs :

L'image au début elle est découpée en sous-bloc de$8 \times 8$ ($ou$ $16 \times 16$), chaque sous – bloc seras traiter individuellement, si c'est une image couleur il faut avant tous faire une transformation couleur ensuite un sous échantillonnage.

## 3.7.2.2Décalage de niveau (Level Shifting) :

Deuxième étape, préréglage (offset), en effectuant un décalage de niveaux, en anglais « Level Shifting » (pour une image monochrome, c'est les niveaux du gris codes sur 8bits), ou chaque éléments de chaque sous-bloc doit être amené a l'intervalle [-128,127], pour faire cela, on soustrait a chaque élément un scalaire 128, cela va permettre d'augmenter la tolérance de précision de coefficient de la DCT.

### 3.7.2.3 Applications de la DCT

La DCT est une transformation linéaire permettant de disproportionner certains coefficients transformés de telle sorte que leur abandon n'entraîne pas de distorsion significative après reconstruction. De plus, la DCT génère des coefficients réels, les plus petits de ceux-ci étant localisés dans une zone fréquentielle où l'œil a une acuité faible.

Une fois la DCT calculée sur un bloc, nous obtenons une matrice carrée des valeurs pour chacune des fréquences. La figure 1 montre un exemple de compression JPEG sur un bloc de 8x8 pixels à 256 niveaux de gris. Les valeurs de la matrice DCT ont été arrondies à l'entier le plus proche. La composante (0,0) est le coefficient continu (1210). Il représente une valeur ``moyenne'' de la grandeur d'ensemble de la matrice d'entrée.

Ce n'est pas exactement la moyenne au sens statistique du terme, l'ordre de grandeur n'étant pas le même, mais c'est un nombre proportionnel à la somme de toutes les valeurs du signal. Les autres valeurs de la DCT représentent des ``écarts'' par rapport à cette moyenne. Les valeurs de la matrice d'indices (0,*j*) (respectivement (*i*,0)) sont les composantes continues le long de l'axe *Y* (resp. *X*) pour la fréquence *j* (resp. *i*) le long de l'axe *X* (resp. *Y*). On remarque une tendance générale des valeurs de la matrice à s'approcher de 0 lorsqu'on s'éloigne du coin supérieur gauche, c'est-à-dire lorsqu'on monte dans les plus hautes fréquences. Cela traduit le fait que l'information effective de l'image est concentrée dans les basses fréquences. C'est le cas de la majorité des images.

Matrice de pixels d'entrée

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

Matrice DCT

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

### 3.7.2.4 La quantification

Le but de la deuxième étape de la méthode JPEG, l'étape de quantification, est de diminuer la précision du stockage des entiers de la matrice DCT pour diminuer le nombre de bits occupés par chaque entier. C'est la seule partie non-conservative de la méthode (excepté les arrondis effectués). Puisque les informations de basses fréquences sont plus pertinentes que les informations de hautes fréquences, la diminution de précision doit être plus forte dans les hautes fréquences. La perte de précision va donc être de plus en plus grande lorsqu'on s'éloigne de la position (0,0). Pour cela on utilise une *matrice de quantification* contenant des entiers par lesquels seront divisées les valeurs de la matrice DCT. Ces entiers seront de plus en plus grands lorsqu'on s'éloigne de la position (0,0). Elle filtre les hautes fréquences.

La valeur d'un élément de la matrice DCT quantifiée sera égale à l'arrondi, à l'entier le plus proche, du quotient de la valeur correspondante de la matrice DCT par la valeur correspondante de la matrice de quantification. Lors de la décompression, il suffira de

multiplier la valeur de la matrice DCT quantifiée par l'élément correspondant de la matrice de quantification pour obtenir une approximation de la valeur de la DCT.

***La matrice obtenue sera appelée matrice DCT déquantifiée.***

Bien que la spécification JPEG n'impose aucune contrainte sur la matrice de quantification, l'organisme de standardisation ISO a développé un ensemble standard de valeurs de quantifications utilisables par les programmeurs de code JPEG. Les matrices de quantifications intéressantes sont celles permettant de ``choisir'' la perte de qualité acceptable. Ce choix a été rendu possible grâce aux tests intensifs des matrices. Habituellement, on prend pour matrice de quantification $Q = (qi,j) AVEC qi,j = \frac{1}{1+k.(1+i+j)}$ avec i l'indice de ligne et j indice de colonne et k facteur de qualité (choisie entre 1 et 25).

On remarque que beaucoup de composantes de hautes fréquences de la matrice quantifiée ont été tronquées à zéro, éliminant leurs effets sur l'image. Par contre, les composantes pertinentes ont été peu modifiées.

Matrice de pixels d'entrée

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

Matrice DCT quantifiée

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

Matrice DCT

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

Matrice DCT déquantifiée (décompression)

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

Matrice de quantification

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

Matrice de pixels de sortie (décompression)

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

**Figure exemple complet de compression et de décompression jpeg**

### 3.7.2.5 Codage des coefficients DC et AC :

Maintenant qu'on a les coefficients DCT quantifiés et arrondis, les coefficients AC et coefficients DC auront deux codages différents avant de subir le codage entropique de Huffman.

### a) Les coefficients DC :

Ils seront code par un codage prédictif DPCM par exemple, ce type de codage permet de prédire la valeur du coefficient DC de chaque sous blocs selon la connaissance de la valeur du coefficient du bloc précédent ou les blocs précédents.

- **Les coefficients AC:**

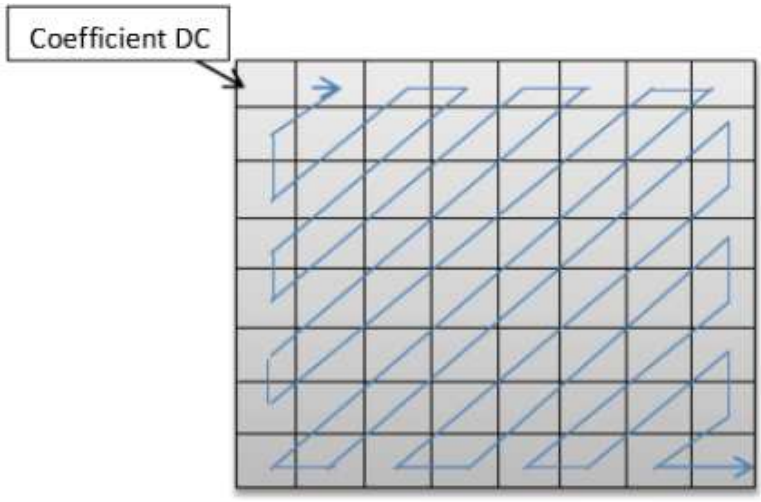



**Figure3.4_ Scan zig-zig**

**b) Les coefficients AC**

Les coefficients AC seront récupérer sur un vecteur par un scan en zig-zig de la matrice de chaque sous bloc afin de mettre les coefficients élevées en premier et les zéros en fin comme illustre dans la figure…Ensuite, ces coefficients seront codé par le codage RLC (Run Length Coding) ou les suites des coefficients AC qui se suivent de même valeurs sont codées par leur nombre de redondance sur la séquence par exemple :

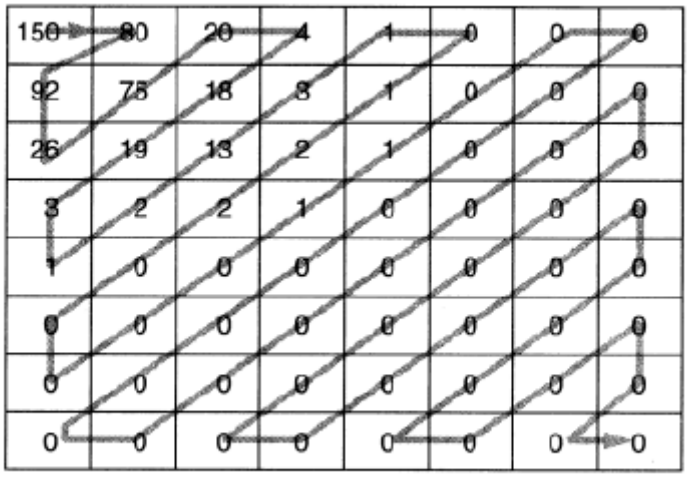



**Figure3.5_ séquence du scan zigzag**

Ce qui donne la suite suivante : 150, 80, 92, 26, 75, 20, 4, 18, 19, 3, 1, 2, 13, 3, 1, 0, 1, 2, 2, 0, 0, 0, 0, 0, 1, 1, 0,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, etc.

Cette séquence a la propriété de parcourir les éléments en commençant par les basses fréquences et de traiter les fréquences de plus en plus hautes. Puisque la matrice DCT quantifiée contient beaucoup de composantes de hautes fréquences nulles, l'ordre de la séquence zigzag va engendrer de longues suites de 0 consécutifs. Deux mécanismes sont mis en œuvre pour comprimer la matrice DCT quantifiée. D'une part, les suites de valeurs nulles sont simplement codées en donnant le nombre de 0 successifs. D'autre part, les valeurs non nulles seront codées en utilisant une méthode statistique de type Huffman ou arithmétique.

### 3.7.2.6 Codage :

Dans cette partie du codage on a porté une modification car la méthode de la compression JPEG avec perte dans la partie codage se fait par une méthode de codage entropique de type codage Huffman est remplacée par une méthode de codage partielle.

Le codage de la matrice DCT quantifier se fait par une méthode de cryptage partielle qui propose de chiffrer que les coefficients fréquentielles relatif aux basses fréquences.

Chaque coefficient fréquentielle est quantifier en basse fréquence est chiffrer a partir d'un générateur de nombre pseudo-aléatoire modulo le spectre de chaque coefficient fréquentielle.

Cette méthode de cryptage partielle est décris par l'équation suivante :

$$E(u,v) = \left(rand(\quad) + F'(u,v)\right) modulo(\frac{2NC(u)C(v)}{Q(u,v)}$$

## 3.8 Discussion

La compression JPEG pour quelle soit efficace doit être effectuée de façon optimale a fin de ne pas perdre trop en qualité de l'image et en parallèle réduire le poids de l'image, c'est à dire trouver un compromis rapport taille /qualité qu'est le standard JPEG.

La compression JPEG en mode DCT est une compression avec perte non adapte au besoin de précision, mais serait le format le mieux adapte pour les mémoires (grâce aux JPEG que les appareilles photo numériques ont fait leurs apparition) ou pour le web(rapidité de transmission),cela dit, un format nouveau a été développer pour les application qui demande une fidélité maximum de l'image traite a l'image réel, comme l'imagerie médicale, le format JPEG2000 qui a les mêmes avantage du JPEG mais base sur une compression sans perte en utilisant une nouvelle transforme appelée la transforme en ondelettes.

# Chapitre IV

## 4.1. Préambule

L'implémentation est la phase la plus importante après celle de la conception (chemin suivie). Le choix des outils de développement influence énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser.

Cette phase consiste à appliquer la norme de compression jpeg qui est très utilise pour le codage des images bitmap et des photos en utilisant quelque transformation que nous allons développer dans la réalisation de cette application, autrement dis appliquer des transformer sur le modèle conceptuel établi précédemment en des composants logiciels formant notre application.

Dans ce chapitre, nous allons commencer par la description de l'environnement de travail puis à dégager et élaborer les composants de notre système (application).

## 4.2. Environnement de travail

L'environnement de travail est constitué par deux parties nommées environnement matériel et environnement logiciel.

### 4.2.1 Environnement matériel

Le développement de l'environnement matériel est caractérisé par :

- Système d'exploitation : Windows XP Professionnel.
- CPU : Pentium M, 1.6 GHz
- Mémoire : 4 Go

### 4.2.2 Environnement logiciel

L'environnement logiciel consiste le composant suivant : Matlab(R2010)

## 4.3. Implémentation

### 4.3.1 Choix de langage de programmation : Java

Pour implémenter notre système, le langage de programmation mathlab est le mieux adapté. En effet, Mathlab s'annonce comme une des évolutions majeures de la programmation. Pour la première fois, un langage efficace, performant, standard et facile à apprendre (et, de plus, gratuit) est disponible.

Il est un langage multi-plate-forme qui permettrait, selon le principa1 proposé par Sun Microsystems, son concepteur, d'écrire des applications capables de fonctionner dans tous les

environnements. L'objectif était de taille, puisqu'il a réussit à gagner une grande popularité auprès des programmeurs grâce à ses avantages. Ce langage offre une portabilité maximale grâce à une indépendance totale par rapport au système.

### 4.3.2 Développement de l'application

Dans cette partie, nous allons présenter les différentes phases de la réalisation de notre application en mentionnant des imprimes écrans de notre application.

#### 4.3.2.1 Chargement automatique de l'image

C'est la partie sur laquelle s'effectue le chargement automatique de l'image sous mathlab

En utilisant le code mathlab suivant :

```
%%% charge et lecture de l'image %%%
%-------------------------------------------------------------------------

% ********************************************************************
nomfich=uigetfile('*.bmp; *.pgm','Merci de charger une image...');
if (nomfich), eval(['X=double(imread(''',nomfich,'''));']);end
```

**Le résultat :**

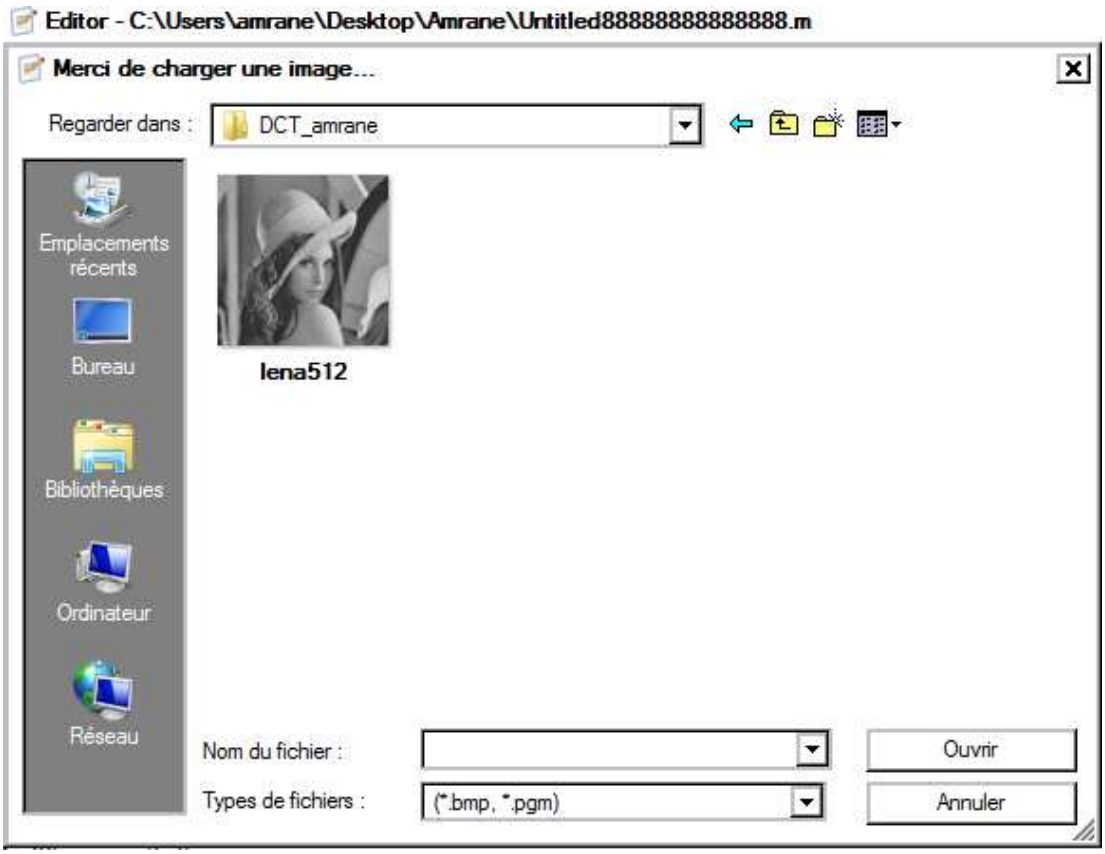



**Figure 4.1_figure chargement automatique sous matlab**

### 4.3.2.2 Affichage de l'image sous matlab :

C'est la partie sur la quelle on affiche notre image « lena512 » sous mathlab. Le code permettant de faire cette instruction sous mathlab est le suivant :

```
image_show(X,256,1,'Image originale');
```

**Le résultat :**


**Figure 4.2_image original lena512**

### 4.3.2.3 Application de la DCT

Dans cette partie l'image est découpée en blocs de taille 8× 8 pixels pour appliquer a chacun d'entre eux une DCT l'instruction « blokproc » permet de découper notre image en blocs de taille [R R] et d'appliquer une fonction DCT sur chacun de ces blocs.

Le code permettant de réaliser cette fonction est le suivant :

```
%
                    %%% Application de la DCT sur l'image %%%
%-------------------------------------------------------------------------
% disp(' input size of bloc RxR: 8x8 or 16x16 or 32x32.')
% R=input('R=');%type de bloc (8*8 or 16*16 or 32*32).

R=8;
Xdct = blkproc(X,[R R],'dct2');%Transformation dct de matrice X.

image_show(Xdct,256,1,'Image DCT');
%-------------------------------------------------------------------------
```

**Le résultat :**


:

**Figure4.3_figure image DCT**

L'image est découpée en bloc de 8 × 8 pixels. Ensuite la Transformée en Cosinus Discrète (DCT) est appliqué sur les pixels de chaque bloc.

On remarque que limage est très uniforme dans le noir et aussi une concentration de pic blanc qui est une représentation de  basses fréquences.

La DCT dans notre cas a pour but de séparer les hautes fréquences des basses fréquences dans l'unique but qui est de diminuer ou d'éliminer les redondances inter pixels, sachant que la transformation seul ans quantification et codage n'est pas une compression, ce n'est juste qu'une étape post-compression.

### 4.3.2.4 La quantification

La  quantification est l'étape de l'algorithme de compression JPEG au cours de laquelle se produit la majeure partie de la perte d'information sachant que cette perte est faite dune manière astucieuse. Parce que le pas de quantification dont dépend la précision de l'image restituée, va dépendre de la position de la valeur dans la matrice. Nous allons prendre un pas relativement petit pour les valeurs importantes (en haut à gauche) et prendre un pas de plus en plus grand au fur et à mesure qu'on descend vers le bas et la droite de la matrice. L'ensemble des pas qui vont être utilisés constituent ce que l'on appelle une matrice de quantification. Une matrice peut être fabriquée grâce une petite formule :

**Q (i, j) = 1 + (1 + i + j) × q, q est le facteur de qualité qui varie de 1 a 25.**

Le code sous Mathlab permettent le calcule de la matrice de quantification est le suivant :

```
%calcul de la matrice de quantification
q=4;%****la valeur de q varie entre 1 à 25***************
for i=0:7
    for j=0:7
        Q(i+1,j+1)=1+((1+i+j)*q);
   end;
end;
```

Afin de réaliser ou d'appliquer cette opération de la quantification qui a pour but d'atténuer les hautes fréquences, c'est a dire celles aux quelles l'œil humain est très peu sensible, car ce sont des fréquences qui possèdent des amplitudes peut sensible. Il faut faire diviser la matrice de la DCT par cette matrice de quantification obtenue précédemment.

Le code sous Mathlab permettant la réalisation de la quantification est comme suit :

```
% (quantification)

  fun = @(x) round(x./Q);


Xdctq = blkproc(Xdct,[R R],fun);


image_show(Xdctq,256,1,'Image dct quantifiée');
```

**Le résultat :**

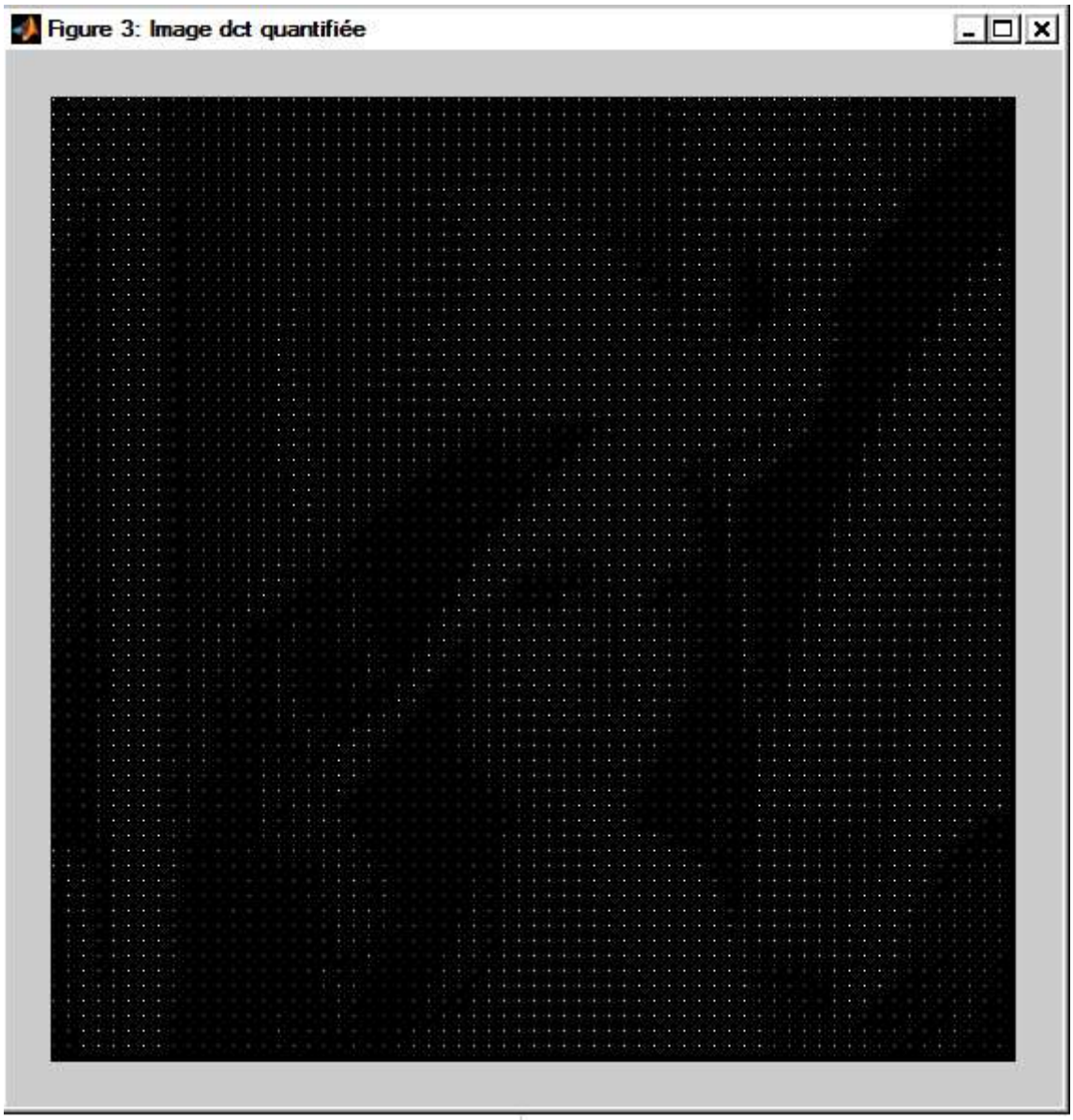



**Figure4.4_image DCT quantifié**

On remarque une augmentation de concentration des pics blanc qui représentent les basses fréquences ce qui signifie que l'atténuation des hautes fréquences est fait.

### 4.3.2.5 Le cryptage partiel

Dans cette partie, l'image est partiellement cryptée, en chiffrant en plus tous les coefficients de la première colonne et la première ligne des blocs 8 X 8.

Le code permettant de réaliser cette fonction est le suivant :

```
%**************cryptage*************************************

function TT=crypt(T)

%calcul de la matrice de quantification
q=4;%****la valeur de q varie entre 1 à 25***************

for i=0:7
    for j=0:7
        Q(i+1,j+1)=1+((1+i+j)*q);
   end;
end;

%*****************************

for i=0:7
    for j=0:7

            if (i==0 && j==0)


                TT(i+1,j+1) = (rand() + T(1,1))* mod(8,Q(1,1));

            else

                TT(i+1,j+1)=T(i+1,j+1);

            end;

    end;

end;
```

**Le résultat**

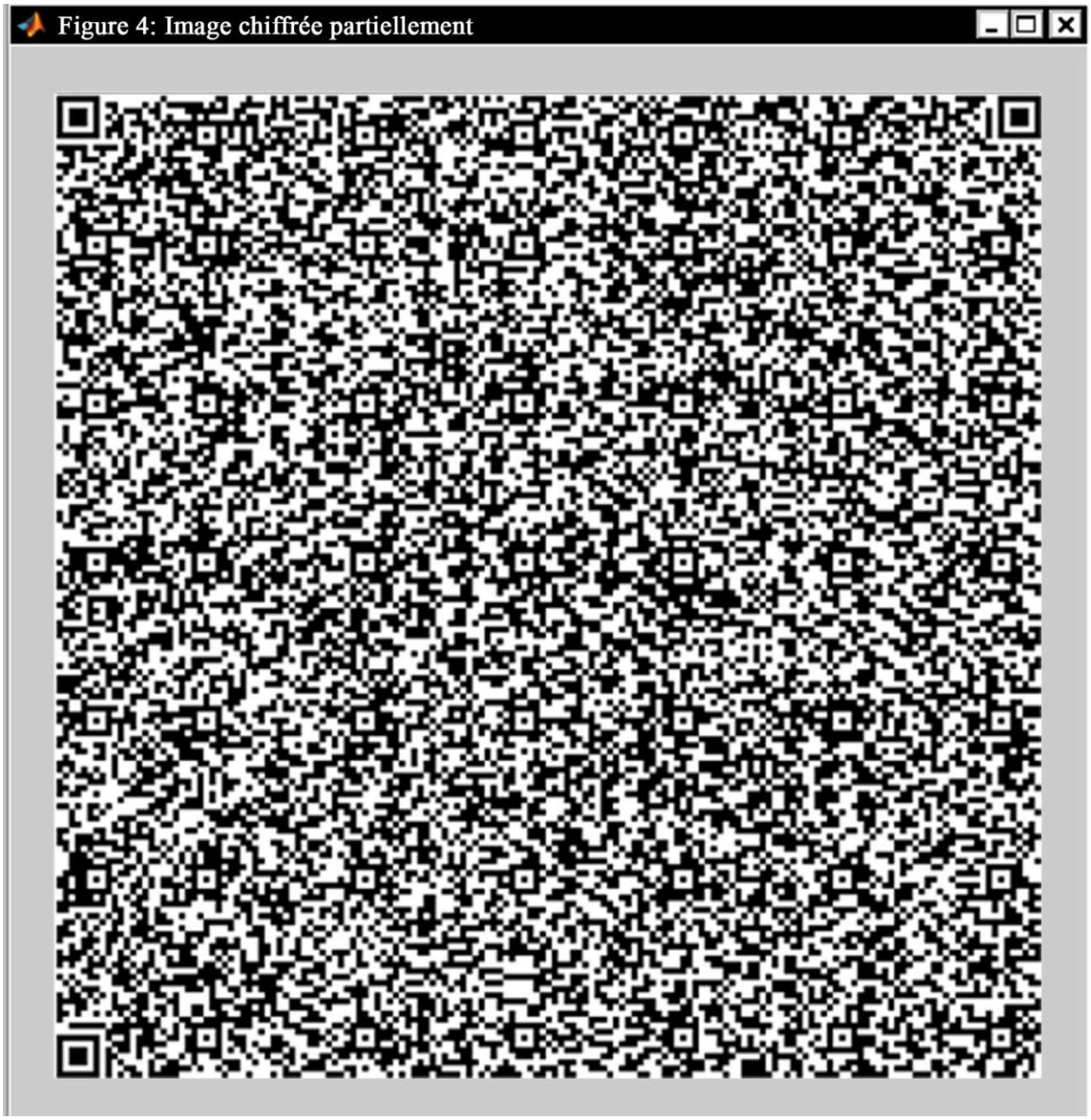



**Figure 4.5_image crypto-comprimée partiellement**

### 4.3.2.6 La déquantification

Qui est une transformer inverse de la quantification. Elle permet le passage d'une image DCT quantifier à une image DCT.

Le code Mathlab qui permet la réalisation de ce passage est le suivant :

```
% (déquantification)

 fun = @(x)(x.*Q);


Xdctqr = blkproc(Xdctq,[R R],fun);

image_show(Xdctqr,256,1,'Image dct déquantifiée');
```

Le résultat

Image DCT reconstruite (déquantifié)


**Figure4.6_image DCT déquantifiée**

### 4.3.2.7 Application de la DCT inverse :

C’est la fonction inverse de la DCT qui permet de retrouver l’image originale reconstruite.

Le code sous Mathlab qui permet de réaliser ce passage est comme suit :

```
            %%% Application de la DCT Inverse %%%
%-------------------------------------------------------------------------------
XR = blkproc(Xdctqr,[R R],'idct2');%Transformation IDCT de matrice X.

image_show(XR,256,1,'Image reconstruite');
%-------------------------------------------------------------------------------
```

Le résultat :


**Figure 4.7_image reconstruite**

## 4.5. Discussion

Dans ce chapitre, nous avons présenté le format de compression jpeg. Ce format de compression est utiliser pour le codage et décodage d'image bitmap et photos, il est très efficace car la perte de qualité d'image occasionnée par l'algorithme de compression peut être maitrisée car le taux de compression des fichiers.

Autrement dis le principal avantage de ce format est le taux de compression réglable qui permet a l'utilisateur de trouver un compromis entre le taux de compression et la qualité de limage.

Ce format de compression jpeg en mode dct est une compression avec perte non adaptè aux besoins de précisions, mais serait le format le mieux adapter pour les memoires (grâce aux jpeg que les appareils photo numériques ont fait leur apparition) ou pour le web (rapidité de transmission).

# Conclusion

## Conclusion générale

Dans cette thèse nous avons présenté une approche de crypto compression qui se base sur une technique de compression sans perte qui est la DCT et quelque méthode de cryptage par mélange de données.

La cryptologie a donc connu une rapide évolution à notre époque du fait, en partie, de deux facteurs essentiels : l'irruption des mathématiques et de l'informatique et le développement des moyens de télécommunication, qui ont eu pour effet de multiplier les activités où intervient la cryptologie.

Le rôle de la cryptologie a évolué au fil des siècles. Auparavant, elle n'avait pour rôle que de protéger un texte écrit. Actuellement, la cryptologie s'étend dans différents domaines tels que l'imagerie, la télémédecine, la téléphonie, le télétraitement, le stockage des données, les communications avec les satellites...

La compression des données est appelée à prendre un rôle encore plus important en raison du développement des réseaux et du multimédia. Son importance est surtout due au décalage qui -existe entre les possibilités matérielles des dispositifs que nous utilisons (débits sur Internet, sur Numéris et sur les divers câbles, capacité des mémoires de masse,...) et les besoins qu'expriment les utilisateurs (visiophonie, vidéo plein écran, transfert de quantités d'information toujours plus importantes dans des délais toujours plus brefs). Quand ce décalage n'existe pas, ce qui est rare, la compression permet de toute façon des économies.

Les méthodes déjà utilisées couramment sont efficaces et sophistiquées (Huffman, LZW, JPEG) et utilisent des théories assez complexes, les méthodes émergentes sont prometteuses (DCT, ondelettes) mais nous sommes loin d'avoir épuisé toutes les pistes de recherche. Les méthodes du futur sauront sans doute s'adapter à la nature des données à compresser et utiliseront l'intelligence artificielle.

La plus part des méthodes de compression ne sont pas figées et définitives. Même si la plus grande part du chemin dans la compression des données semble avoir été accomplie, les recherches se poursuivent pour améliorer encore les performances, au prix d'une complexité toujours accrue, mais compatible avec les progrès parallèles des outils informatiques.

# Conclusion générale

Comme de nombreuses méthodes de compression, le JPEG est basé sur des principes mathématiques très compréhensibles. Mais la difficulté intervient lorsque l'on entre en détail dans les démonstrations de l'algorithme. Théorèmes et principes fondamentaux doivent alors être démontrés, pour expliquer les fondements de la méthode.

L'utilisation de telles méthodes de compression des informations est très répandue et utilisées dans de nombreux domaines : informatique, téléphonie, hi-fi, vidéo,…

A l'heure actuelle la méthode de compression JPEG est parmi les plus utilisées parce qu'elle atteint des taux de compression très élevés sans que les modifications de l'image ne puissent être décelées par l'œil humain.

De plus, beaucoup d'implémentations permettent de choisir la qualité de l'image comprimée grâce à l'utilisation de matrices de quantification paramétrables..

La réputation et donc le nombre d'utilisateurs d'un algorithme de compression dépendent de différents facteurs : le rapport taille/qualité, la vitesse de compression et de décompression. Il est donc intéressant de comparer les différentes méthodes, pour pouvoir choisir la plus adaptées à nos besoins et à son utilisation.

# Annexes

JPEG : Joint Photographique Expert Groupe

LZW : Lampel Ziv-Welch

RLC : Run Length Coding

VLC : Variable Length Coding

DCT: Discrete Cosine Transforme

ISO: International Standards Organisation

CEI: Commission Electronic International

IFF: Identification Friend and Foes

DES: Datta Encryption Standard

ECB: Electronique Code Block

CCB: Chain Block Cipter

PKC : Public Key Cryptosysteme

LZ: Lampel Ziv

QV: Quantification Vectoriel

KLT : Transformation Karhunen Love

DFT : Discrete Fourier Transforme

HT : Hadamord Transforme

WT : Ondelette Transforme

LS : Level Shifting

# Bibliographie

# Bibliographie

[1] Daniel Salles, " Éducation à l'image et aux médias : la liberté de la presse ", Centre de Ressources en éducation aux médias CREM, mars 2005.

[2] Cavet D., " Numérisation des images ", Note technique 013, CNDP, 1994. www.cndp.fr/notestech/13/numerima.htm

[3] AM.Cohen, HL.Resnikoff " Image Compression for Radiology and Telemedicine ", Proc. SPIE 2298, SPIE, Bellingham, Wash., pp. 304-315,1994.

[4] Pascal PLUME, " Compression de données ", Editions Eyrolles, 1993

[5] Charles WAGNER , " De l'image vers la compression ", Rapport de Recherche de l'INRIA, Septembre 1993

[6] www.jpeg.org

[7] M.Abdat, " Etudes des techniques de compression des images fixes et amélioration de la résistance aux erreurs de transmission ",Thèse PhD, Blida-Algerie, Dec.1995.

[8] Jean-Paul Guillois, " Techniques de compression des images " Edition Hermès, 1996.

[9] T.HUFFMAN, D.MULLER, " Techniques de compression des signaux vidéo pour la com- munication multimédia ", Revue des télécommunications 4e édition, 1993.

[10] Encyclopedie Universalis

[11] Encyclopedie Hachette

[12] Encyclopedie Larousse .

[13] Le Quid

[14] La Cryptologie, éditions P.U.F. (III)

[15] http://cryptage.online.fr

[16] Simmons G. Symetric and asymmetric encryption *ACM Computing survey* Vol 11 No 4 pp. 305-330

[17] http ://www.encarta.fr

[18] Science et avenir

[19] M.Nelson, " La compression des données ", éditions Dunod, 1993.

[20] O. Robineau, " Compresser les images fixes ", le monde de l'informatique, 14 Mars 1997.

[21] M.Abdat, " Etudes des techniques de compression des images fixes et amélioration de la résistance aux erreurs de transmission ",Thèse PhD, Blida-Algerie, Dec.1995.

[22] F.Falzon, " Analyse Multi- échelle, Détection des singularités et Caractérisation de la Régularité des Images ", thèse de doctorat, Nice - Sophia- Antipolis, Décembre 1994.

[23] H.C.Andrews, W.K.Pratt, " Transform image coding ", Processing communication, pp 63,84. 1969.

[24] Ziv J. et Lempel A., " Compression of individual sequences via varaible rate coding ", IEEE Transactions on Information Theory, Vol. 24, septembre 1978.

[25] H.C.Andrews, W.K.Pratt, " Transform image coding ", Processing communication, pp 63,84. 1969

[26] Marseau X., " Compression et cryptage en informatique ", Editions Hermès, Collection Traité des nouvelles technologies, série informatique, 1992.

[27] M.Nelson, " La compression des données ", éditions Dunod, 1993.

[28] Pascal PLUME, " Compression de données ", Editions Eyrolles, 1993.

[29] Gray R. M., " Vector quantization ", IEEE ASSP Magazine, pages 4-29, April 1984.

[30] Dossiers de l'ingénierie éducative, " De l'analogique au numérique ", CNDP, Les cartes vocales, N_ 13, juin 1993, p. 16-18.

[31] N.Ahmed, K.R.Rao, " Orthogonal transforms for digital ", Signal processing NY, 1975.

[32] A. Bijaoui, J.L.Strack, F. Murtagh, " Restauration des images MultiEchelles par l'algorithme à trous ", INIST-CNRS, I-Revues, Traitement du signal et des Images, vol.11, pp.229- 243, 1994.

[33] M.Abdat, " Etudes des techniques de compression des images fixes et amélioration de la résistance aux erreurs de transmission ",Thèse PhD, Blida-Algerie, Dec.1995.