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

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

Un système de crypto-compression d'images basé sur le Stream Cipher et la compression RLE

Option : Réseaux et Sécurité Informatique
Thème
Présenté Par :

DJEFFALI Khaled

Devant le jury :

Mr.MEKHAZNIA Tahar	*MCA*	*Univérsité Larbi Tébessi*	*Président*
Mr.NOUIOUA Tarek	*MAA*	*Univérsité Larbi Tébessi*	*Examinateur*
Mr.MENASSEL Rafik	*MCA*	*Univérsité Larbi Tébessi*	*Encadreur*

Date de soutenance : Juin 2020

Remerciements

Tout d'abord, je remercie le Dieu, notre créateur de m'avoir donné la force, la Volonté et le courage afin d'accomplir ce travail modeste.

Je tiens à exprimer mes profondes gratitudes et mon immense respect à mon encadreur ***Dr. MENASSEL Rafik*** *pour la qualité de son encadrement, sa disponibilité, ses hautes qualités morales et scientifiques et pour m'avoir découvert un domaine de recherche si passionnant et aussi pour ses conseils précieux et son soutien affectif durant mon étude et la 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 vous PAPA.

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 Mon frère et ma grand-mère 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 Larbi Tebessi de Tébessa et en particulier, ceux du département

Mathématiques et Informatique.

A tous mes amis et camarades et tous ceux qui ont contribué à la réalisation de ce travail.

Résumé La compression s'impose comme une étape incontournable pour optimiser l'utilisation des grands volumes d'informations dans les réseaux informatiques. L'objectif principal de la compression d'image est de réduire la quantité d'information nécessaire à une représentation visuelle fidèle à l'image originale. En générale on différencie les méthodes de compression selon la perte d'informations. Les méthodes réversibles, utilisent uniquement le principe de la réduction de la redondance et n'engendrent pas de perte. Celles irréversibles, définissent une représentation approximative de l'information.

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

Dans ce travail de Master, nous allons mener, une étude plus détaillée au sujet des système de crypto-compression basé sur une compression sans perte RLE, et un algorithme de chiffrement moderne RC4 basé sur la technique du Stream Cipher ou cryptage par flots

Mots clés : Compression d'image, Cryptographie, Crypto-compression, Stream Cipher, RLE.

Abstract

Compression is an essential step to optimize the use of large volumes of information in computer networks. The main objective of image compression is to reduce the amount of information necessary for a visual representation faithful to the original image. In general we differentiate the compression methods according to the loss of information. Reversible methods use only the principle of reducing redundancy and do not cause loss. The irreversible ones define an approximate representation of the information.

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

In this work of Master, we will lead, a more detailed study about the crypto-compression system based on a lossless compression RLE, and a modern encryption algorithm RC4 based on the technique of Stream Cipher or encryption by streams

Keywords: Image compression, Cryptography, Crypto-compression, Stream Cipher, RLE.,

définissent une représentation approximative de l'information.

ملخص

يعد الضغط خطوة أساسية لتحسين استخدام كميات كبيرة من المعلومات في شبكات الكمبيوتر. الهدف الرئيسي لضغط الصورة هو تقليل كمية المعلومات اللازمة للتمثيل المرئي المخلص للصورة الأصلية. بشكل عام نحن نميز طرق الضغط وفقًا لفقدان المعلومات. تستخدم الطرق القابلة للعكس فقط مبدأ تقليل التكرار ولا تسبب الخسارة. تحدد تلك التي لا رجعة فيها تمثيلًا تقريبيًا للمعلومات.

من ناحية أخرى ، يتم وصف تشفير البيانات بشكل عام على أساس اتصال سري للمعلومات بين محاورين. في نظام الكمبيوتر ، تحدث هذه السرية في عدة أشكال ، ولا سيما في حماية التخزين والوصول ونقل المعلومات.

في مذكرة الماستر هذه ، أجرينا دراسة أكثر تفصيلاً حول نظام ضغط التشفير مبنية على ضغط RLE بدون ضياع وخوارزمية تشفير حديثة RC4 استنادًا إلى تقنية التشفير بواسطة التدفقاتStreamCipher

الكلمات المفتاحية: ضغط الصور ، التشفير ، ضغط التشفير ، Stream Cipher، RLE

Table Des matières

Liste des figures

Liste des tableaux

Introduction générale

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

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

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

Dans ce mémoire de Master nous nous intéressant à la combinaison des deux techniques, à savoir la compression et la cryptographie dans l'objectif principale est de mettre en œuvre un nouveau crypto système basé sur une compression sans perte RLE et un algorithme de cryptage RC4 fondé sur la méthode de chiffrement par flots (Stream Cipher).

Ce mémoire de Master est organisé ainsi :

Nous commençons tout d'abord notre mémoire par une introduction générale qui va mettre en claire le contexte de notre étude. Ensuite….

Le premier chapitre est réservé aux généralités théoriques sur les images, la compression et la cryptographie, où on a essayé de faire familiariser les notions de composition de l'image (pixel) pour commencer, ensuite on a passé aux différents procédés de compression d'images toute en glissant par les mesures de performances utilisées pour choisir une méthode de compression par rapport aux autres suivant les conditions d'usage.

Le deuxième chapitre introduit les systèmes de crypto compression, où on fera le tour d'horizon sur les différents systèmes existants, tout en mettant l'accent sur les différences

majeurs qui existent entre ces systèmes. A la fin de ce chapitre nous détaillons notre approche proposée, qui s'articule sur une méthode de compression sans perte « RLE » et la méthode de chiffrement par flots « Stream Cipher ».

Le dernier chapitre est considéré comme un dossier technique, comportant l'environnement logiciel et matériel utilisé pour développer notre application. Ce chapitre contient aussi l'ensemble des tests et résultats obtenus en appliquant le modèle proposé de crypto compression sur des images de différentes nature.

Nous terminerons ce mémoire par une conclusion générale, résumant notre travail et mettant en surface les difficultés rencontrées tout au long la période de l'élaboration de ce modeste travail.

Chapitre 1 : Généralités sur l'Image, la Compression et la Cryptographie

Chapitre 1 : Généralités sur l'image, la compression et la cryptographie.

1.1 Introduction

Ce premier chapitre se compose en deux parties primordiales, la première se concentre sur des généralités concernant l'image toute en passant par une étude bibliographique des méthodes de compression d'images, tandis que la deuxième se penche sur la cryptographie des donnée et les algorithmes .

Dans la première partie on s'intéresse en premier lieu à une définition de l'image, en suite on passe par les différents systèmes de représentation de couleurs ainsi que quelques mesures de performances, et on termine par l'étude détaillée des différents algorithmes de compression de données des deux familles ***Avec*** et ***Sans*** perte de données.

La deuxième partie se focalise sur les méthodes de cryptage, où on verra en détails la notion de cryptographie ainsi que les différents algorithmes utilisés dans ce sens afin de sécuriser et assurer la transmission de données.

1.2 Notions d'images et de compression

L'image est une reproduction d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est autant un ensemble organisé d'informations qui, après visualisation sur un écran, ont un sens pour l'œil humain.

Une image peut être symbolisée sous un aspect vectoriel ou sous configuration d'une matrice de points, bitmap.

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

Pour bien saisir le fonctionnement de la compression d'image, nous devons tout d'abord savoir quelles sont les différentes représentations informatiques d'une image et autant de choses qu'il est nécessaire de détailler dans une première partie.

Pour comprendre comment fonctionne la compression d'image, nous devons tout d'abord savoir quelles sont les différentes représentations informatiques d'une image, par quels moyen peut-on réduire la taille des fichiers, comment représenter les couleurs bref autant de choses qu'il est nécessaire de détailler dans une première partie.

1.2.1 Notion de pixel [1]

En infographie, une image est composée de points, appelés pixel (abréviation de PICTure Element). Ces pixels sont rassemblés en lignes et en colonnes afin de former un espace à deux dimensions. Chaque point sera représenté par ses coordonnées (X,Y), avec X l'axe orienté de gauche à droite, et Y l'axe orienté de haut en bas. La lettre A, par exemple, peut être affichée comme un groupe de pixels (Figure1).

Figure 1.1 : Pixels

1.2.2 Taille et définition d'une image

Pour connaître la définition (en octets) d'une image, il est nécessaire de compter la quantité de pixels que renferme cette dernière, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. La taille (ou poids) de l'image est alors le nombre de pixels que multiplie la taille de chacun de ces éléments. Les définitions les plus répandues sont 640 x 480, 600 x 800, 1024 x 768 pixels…

- Prenons l'exemple d'une image 1024 x 768, dont la couleur est codée sur 24 bits (1 octet pour les nuances de rouge, 1 pour le bleu et 1 octet pour le vert, codage True color ou RGB)
- Nombre de pixels :
    - 1024 x 360 = 786432 pixels
- Poids de l'image :
    - 786432 x 3 = 2359296 octets
- Soit une image de 2359296 / 1024 = 2304 Ko, ou 2304 / 1024 = 2,25 Mo, ce qui est assez conséquent, surtout lorsqu'on veut transmettre l'image…

1.2.3 Mesures de performances [1]

1.2.3.1 Taux de compression

Le taux de compression soumis à une image est directement proportionnel à la quantité de redondance d'information qu'elle possède. Le taux de compression est utilisé pour mesurer le résultat d'un procédé de compression. Il est représenté soit comme une formule, équation (a), soit comme un facteur, équation (b). Dans les équations (a) et (b), Io est la taille de l'image

originale en octet et Ic la taille de l'image comprimée. Le taux de compression peut être aussi quantifié par le nombre moyen de bits par pixel (bpp), équation (c). L'élément BitsIc est le nombre total de bits de l'image comprimée et PixelsIo est le nombre total de pixels de l'image originale.

$$CR = \frac{Io}{Ic} \text{ (a);} \qquad CR = (Io/Ic) \text{ (b);} \qquad CR = \frac{BitsIc}{PixelsIo} \text{ (c)}$$

1.2.3.2 Redondance

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

1.2.3.3 Entropie : [1]

L'entropie est une grandeur qui caractérise la quantité d'information que contient une image. Par exemple une image dont tous les pixels ont la même valeur contient très peu d'information car elle est extrêmement redondante, son entropie est faible. En revanche, une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'information, son entropie est forte. Ceci est comparable à l'entropie en thermodynamique qui croit avec le désordre.

En pratique, l'entropie d'une image numérique est inversement liée à la probabilité d'apparition des niveaux de gris dans l'image. Plus une valeur de gris k est rare, plus sa probabilité d'apparition p(k) est faible, et cela contribue à une entropie globale plus grande. Par définition, l'entropie d'ordre zéro H_0 est donnée par :

$$H_0 = -\sum_{k=0}^{2^R-1} p(k) * ln\, p(k)\ ppb$$

1.2.3.4 Mesures de distorsion : [1]

La distorsion (*D*) est l'erreur introduite par l'opération de compression, due au fait qu'éventuellement l'image reconstruite n'est pas exactement identique à l'image originale.
La mesure de distorsion, utilisée généralement en compression d'image, est l'erreur quadratique moyenne MSE *(Mean Square Error).* Cette grandeur est définie par la moyenne carrée e_{mn}^2 entre le pixel (m,n) de l'image originale I(m,n), et le pixel (m,n) de l'image reconstruite I'(m,n)

$$MSE = \frac{1}{M * N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(m,n) - I'(m,n)]$$

1.2.4 Compression

La compression s'exige semblablement comme une étape inévitable pour améliorer l'utilisation de ces grands volumes d'informations dans les différentes applications informatiques. L'objectif principal de cette discipline est de réduire la quantité d'information nécessaire à une reproduction visuelle fidèle à l'image originale. En générale on différencie les méthodes de compression selon la perte d'informations. Les méthodes réversibles « sans perte », utilisent uniquement le principe de la diminution de la redondance et n'engendrent pas de perte. Les méthodes irréversibles « avec perte », définissent une reproduction approchée de l'information.

1.2.4.1 Méthodes sans perte

La compression sans perte ou codage entropique ou codage réversible permet de retrouver la valeur exacte du signal comprimé. En fait, la même information est réécrite d'une manière plus concise. Le processus de codage sans perte crée des "mots-codes" à partir d'un dictionnaire statique ou d'un dictionnaire construit dynamiquement. Ces processus s'appuient sur des informations statistiques de l'image. Les codes statistiques les plus répandus sont le codage d'Huffman et le codage arithmétique. Le codage statistique permet de s'approcher au mieux de l'entropie [1]. Ils ont pour principe d'associer aux valeurs les plus probables les mots binaires les plus courts.

a) Méthode d'Huffman

Huffman [2] a inspiré une méthode statistique qui permet d'attribuer un mot-code binaire aux différents symboles (pixel) à compresser. La probabilité d'occurrence du symbole dans l'image est prise en compte en attribuant aux plus fréquents des codes courts, et aux plus rares des codes longs, VLC - Variable Length Coding. La suite finale de pixels codés à longueurs variables sera

plus petite que la taille originale. Le codeur Huffman crée un arbre ordonné à partir de tous les symboles et de leur fréquence d'apparition. Les branches sont construites récursivement en partant des symboles les plus fréquents. Le code de chaque symbole correspond à la suite des codes le long du chemin allant de ce caractère à la racine. Plus le symbole est profond dans l'arbre plus la quantité de bits pour le représenter est important. La figure 3 présente un exemple de codeur d'Huffman. Le tableau de gauche montre que nous avons au total 21 symboles qui sont représentés en octets équivalant à 168 bits. Après la construction de l'arbre d'Huffman nous pouvons constater un taux de compression de $\boldsymbol{\sigma = \frac{45}{168} = 26.79\%}$.

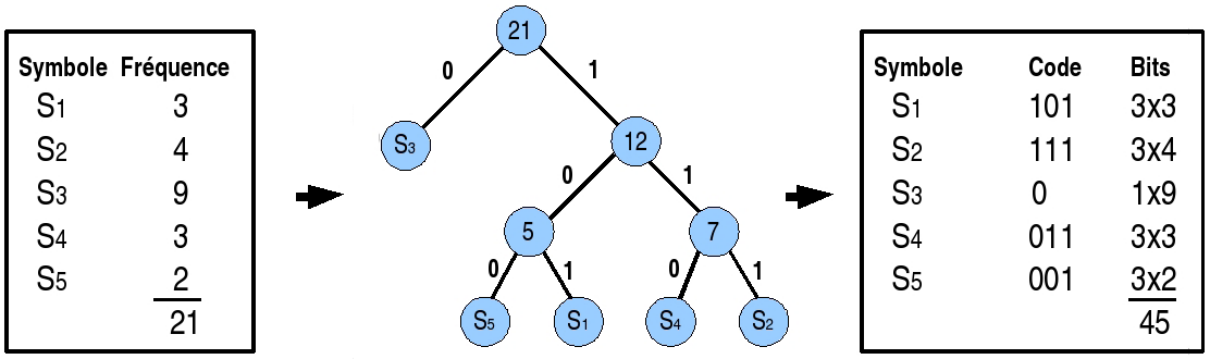


Figure 1.2 : Algorithme d'Huffman.

b) Méthode arithmétique

Le codage arithmétique (CA) [3] est un codage statistique qui attribue à une suite de symboles une valeur réelle. Il consiste à découper l'intervalle des réels [0, 1) en sous intervalles, dont les longueurs sont fonctions des probabilités des symboles. Le codage arithmétique n'attribue pas un code à chaque symbole comme Huffman et les autres codages par blocs, mais un code au message tout entier. Les tableaux suivants présentent un exemple de codage arithmétique avec le message AAOEU.

Alphabet	Probabilité	Probabilité Cumulée	Partition Initiale
A	0,2	0,2	[0 0,2)
E	0 ,4	0,6	[0,2 0,6)
I	0,1	0,7	[0,6 0,7)
O	0,2	0,9	[0,7 0,9)
U	0,1	1,0	[0,9 1,0)

Message	Gauche G	Taille T	Droite D
A	0,0000	0,2000	0,2000
A	0,0000	0,0400	0,0400
O	0,0280	0,0080	0,0360
E	0,0296	0,0032	0,0328
U	0,0325	0,0003	0,0328

Tableau 1.1: Codage arithmétique

Soit l'alphabet {A,E,I,O,U} avec les probabilités {0,2 0,4 0,1 0,2 0,1}. Le codage arithmétique est fait à partir de l'intervalle initial [0, 1) et au fur et à mesure du codage, la longueur de l'intervalle diminue en tenant compte du sous-intervalle précédent.

Nous nous servons des formules (G = GPI +GM_TPI) et (T = TPI _TM) pour construire le nouvel intervalle où les lettres signifient : G-gauche, T-taille, M-message et P-précédant.

Le premier symbole A du message réduit l'intervalle initial à [0 0,2). Le deuxième symbole A du message réduit ce dernier intervalle à [0 0,04) (1/5 de l'intervalle précèdent).

Le symbole O réduit l'intervalle à [0,028 0,036). Le symbole E diminue l'intervalle à [0,0296 0,0328). Enfin, le symbole final U réduit à [0,03248 0,0328). Finalement, tout réel dans l'intervalle [0,03248 0,0328) codera le message AAOEU. Le codage arithmétique est présent dans la norme JPEG (dans les modes Extended DCT-based processes et Lossless processes) et JPEG2000.

Les méthodes de codage statistiques construisent les mots-codes à partir d'un dictionnaire prédéfini, basé sur les statistiques de l'image elle-même. Ce dictionnaire est indispensable pour le décodage. Des nouvelles études et améliorations pour cette approche ont été proposées, nous citons les travaux [4] [5] [6].

Les deux méthodes que nous allons présenter, codage par substitution, n'exigent pas de connaissance à priori de l'image, comme les probabilités d'apparition des pixels par exemple. Elles construisent des dictionnaires dynamiques dont les mots-codes créés sont indépendants de la source.

c) L'algorithme LZW

Lempel et Ziv [7] ont exposé un schéma (LZ77) qui est à la base de tous les algorithmes à dictionnaire dynamique utilisés actuellement. Welch a amélioré leur algorithme et a entreposé un brevet en créant l'algorithme LZW qui génère un dictionnaire dynamique qui contient des motifs du fichier. Son principe consiste à substituer des motifs par un code d'affectation en construisant au fur et à mesure un dictionnaire. Celui-ci est initialisé avec les valeurs de la table ASCII. Chaque octet du fichier est comparé au dictionnaire. S'il n'existe pas, il est ajouté au dictionnaire. L'algorithme LZW fait partie du format d'image, aussi breveté, GIF (Graphics Interchange Format).

d) Codage par plage

Le codage par plage ou RLE Run Length Encoding est recommandé lorsque nous observons des répétitions de symboles consécutifs. Il est utilisé par de nombreux formats d'images (BMP, TIFF, JPEG) [8]. L'idée est de regrouper les pixels voisins ayant la même couleur.

Chaque groupement définit un couple de valeurs P = (plage, n) où plage est le nombre de points voisins ayant la même valeur, et n est cette valeur. Le RLE est d'autant plus performant que les groupements sont étendus, il n'est pas applicable dans tous les cas. Il est recommandé pour les images avec de larges zones uniformes.

La compression d'une image peut être effectuée de manière adaptative : dans les régions uniformes le RLE est appliqué, et dans les zones non uniformes des règles particulières sont créées. Par exemple, au moins trois éléments se répètent consécutivement alors la méthode RLE est utilisée, sinon un caractère de contrôle est inséré, suivi du nombre d'éléments de la chaîne non compressée. D'autres caractères de contrôle peuvent aussi être utilisés pour définir la fin de ligne ou la fin de colonne.

e) Codage par prédiction linéaire

Les algorithmes qui utilisent le codage par prédiction exploitent la redondance spatiale. Il s'agit de prédire la valeur d'un pixel en fonction de la valeur des pixels voisins et de ne coder que l'erreur de prédiction. Le gain en compression est accompli par la variation faible entre pixels voisins, sauf pour les pixels situés sur les contours. Le voisinage peut être défini selon sa connexité (4-connexité ou 8-connexité) ou selon l'ordre du parcours choisi pour accéder aux pixels voisins. L'une des techniques de prédiction la plus simple est la DPCM (Differential Pulse Code Modulation) [9]. Cette technique effectue une prédiction à base d'une combinaison linéaire des valeurs des pixels voisins. Une version adaptative, ADPCM, qui utilise différentes formes de prédiction et de voisinage selon le contexte et le contenu de l'image a été présentée par Kyung et al. [10]. Récemment, Babel et al. [11] ont proposé un codage progressif et multi résolution LAR - (Locally Adaptive Resolution). Il s'agit d'un codeur qui associe le DPCM à une décomposition multicouches suivi d'une transformée Mojette3. La profondeur de la décomposition détermine le type de compression, avec pertes ou sans perte pour le huitième niveau.

1.2.4.2 Méthodes avec pertes

Les méthodes avec pertes (lossy) ou irréversibles sont des méthodes qui tirent parti d'une corrélation (ou redondance) existante dans l'image. L'information perdue est due à l'élimination de cette redondance, ceci rend possible une compression plus importante. Par ailleurs, la perte d'information est toujours discutable et nous nous posons alors la question de la limite acceptable. Cette limite est définie par le type d'application, comme les images médicales ou satellites par exemple.

a) Codage prédictif avec pertes

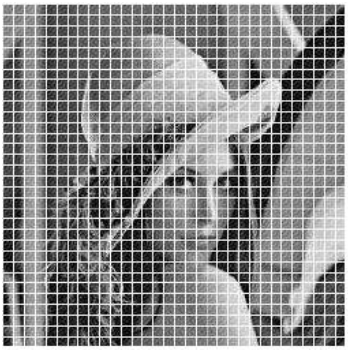
Il existe des techniques qui exploitent la redondance spatiale, cependant la prédiction est faite par approximation. Ces algorithmes ont comme objectif de rechercher un modèle de représentation le plus adéquat de l'information à coder afin d'obtenir un coût de codage minimal. L'idée est de coder l'erreur de prédiction au-dessus d'un seuil. Ce seuil peut être défini par rapport à la qualité de l'image ou le niveau de compression espéré.

b) Codage par transformation

Les méthodes qui utilisent cette technique utilisent des transformations pour produire une décorrélation des redondances spectrales. Les pixels passent d'un espace où ils sont fortement corrélés dans un autre espace où leur corrélation est moindre. Lors de chaque transformation, le signal d'origine est remplacé par sa représentation dans un autre domaine. Dans divers algorithmes cette transformation d'espace est accompagnée d'une quantification et d'un codage entropique pour accomplir la compression de l'image. Ceci est le cas des normes standards de compression : l'algorithme JPEG qui utilise la transformation type DCT et l'algorithme JPEG2000 qui utilise la transformation en ondelettes DWT.

c) Fractales

La compression par fractale est une technique de compression avec pertes encore peu utilisée. Une fractale est une structure géométrique qui se reproduit, dans une boucle infinie, par transformation affine (translation, rotation et mise à l'échelle). Cette structure se refait à toutes les échelles de forme réduite et légèrement déformée. La compression par fractale est basée sur le principe qu'il existe des similarités entre différentes régions isolées d'image. Elle exploite les récurrences des motifs qui, après quelques traitements, peuvent permettre une compression. La figure 6 présente un exemple d'exploitation des motifs



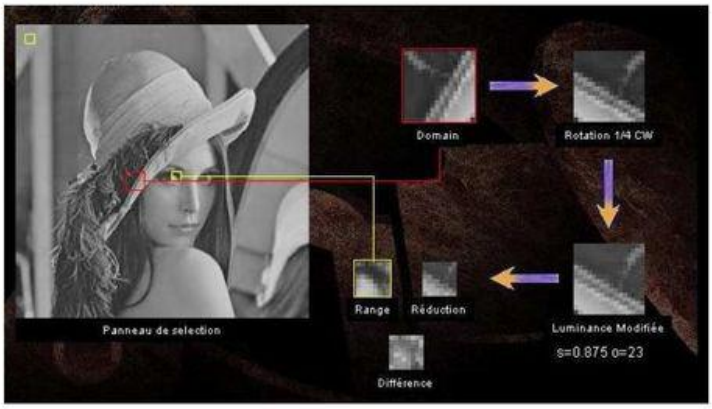


Figure 1.3: Compression par fractales.

d) JPEG [1]

Le comité Joint Photographic Expert Group a été créé en 1986 par la jonction (Joint) de plusieurs groupes qui travaillaient sur la photographie. Ce comité a produit la norme de compression d'images photographiques qui a été standardisée (ISO/IEC/10918-1/1994) et a reçu son nom JPEG. Il est devenu le format le plus populaire très rapidement parce qu'il a été conçu avec différentes contraintes :

- L'algorithme JPEG doit être implémenté sur une grande variété de types de CPU (unité centrale de calcul) et sur des cartes plus spécialisées (appareil photo numérique et téléphone portable par exemple).
- Il doit pouvoir compresser efficacement tout type d'images réelles (images photographiques, médicales) avec pertes et sans perte.
- Il possède quatre modes de fonctionnement : séquentiel (Baseline), progressif (Extended DCT-based), sans perte Lossless, hiérarchique hierarchical.

Entre les 4 modes de compression de la norme JPEG, le séquentiel ou Baseline est le mode principal le plus répandu. Il est basé sur la transformation DCT, quantification scalaire et

le codage d'Huffman sur pixels de 8 bits par plan de couleur. La figure 7 expose une synthèse du mode séquentiel.

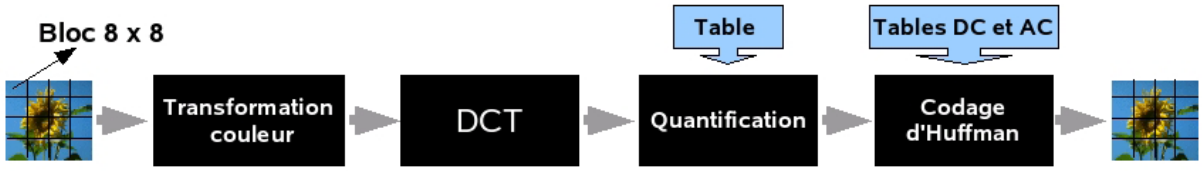


Figure 1.4: Compression JPEG.

e) JPEG2000 - [ISO/IEC/15444-1/ 2000]

Le JPEG2000 [1] remplace le JPEG comme le format standard pour la compression des images. Il a été réalisé dans la perspective de répondre aux exigences des nouvelles applications les plus diversifiées, comme la multi résolution par exemple. La compression JPEG2000 est composée de plusieurs étapes selon les schémas avec pertes et sans perte.

La résistance aux erreurs est une caractéristique particulière du JPEG2000. Après le codage entropique plusieurs caractères de contrôle (segment marks, resynchronising marks) sont insérés dans le flux de bits. Cette démarche est faite pour synchroniser les informations, limiter la taille du segment et éviter la propagation des erreurs.

Une autre fonctionnalité importante du JPEG2000 est la compression par région d'intérêt (ROI). Ceci permet d'avoir des taux de compression différents dans certaines régions de l'image. Les zones importantes peuvent être compressées quasi sans pertes et les zones moins importantes avec un fort taux de compression.

Malgré ces nombreuses fonctionnalités, le JPEG2000 possède quelques inconvénients. Il nécessite entre deux et six fois plus de cycles de CPU que JPEG6 et il n'est pas indiqué pour les machines avec faibles ressources comme les appareils photos numériques par exemple. L'algorithme JPEG est beaucoup moins complexe et il peut être implémenté en hardware.

f) Compression par Ondelettes

La compression par ondelettes est une technique relativement nouvelle, La transformation par Ondelettes est une technique inventée par Summus Ltd et qui consiste à décomposer une image en une myriade de sous-bandes, c'est à dire des images de résolution inférieure. On distingue 4 étapes différentes pour procéder à la transformation:

- Moyenner les pixels de l'image originale deux à deux suivant l'axe horizontal.
- Calculer l'erreur entre l'image originale et l'image sous-échantillonnées dans le sens horizontal.
- Pour chacune des deux images intermédiaires, moyenner les pixels deux à deux suivant l'axe vertical.
- Pour chacune des deux images intermédiaires, calculer l'erreur suivant l'axe vertical.

Le résultat est une image d'approximation qui a une résolution divisée par deux et trois images de détails qui donnent les erreurs entre l'image originale et l'image d'approximation. Cette transformation est répétée autant de fois que nécessaire pour obtenir le nombre voulu de sous-bandes.

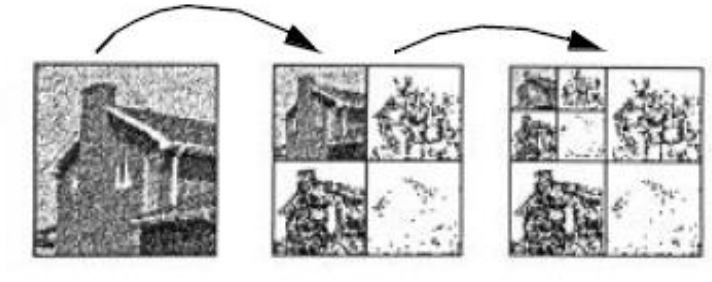


Figure 1.5: Transformation par ondelettes

Il n'y a pas de pertes à ce stade de la transformation. Les pertes subviendront lors de la compression. Les étapes de compression sont les suivantes :

- Transformations par ondelettes
- Quantification : les valeurs des images de détails inférieures à un certain niveau sont éliminées, en fonction de l'efficacité recherchée. C'est cette étape qui introduit des pertes.
- Codage des valeurs restantes.

La transformation inverse par ondelettes reconstruit une image originale. La construction de l'image à partir des sous-bandes restitue l'image en mode progressif. L'affichage de l'image peut s'effectuer en deux modes :

- Soit la taille de l'image augmente au fur et à mesure de la lecture du fichier compressé.

- Soit la résolution de l'image augmente au fur et à mesure de la lecture du fichier compressé.

1.3 La cryptographie [12]

D'énorme quantité d'informations sont échangées sur les réseaux de communication numériques, ces informations peuvent être de différents types (texte, audio, image, vidéo, . . . etc.). Cependant, sécuriser ces informations s'impose comme une étape primordiale et incontournable pour de garantir la confidentialité et de prévenir toute modification ou exploitation non désirée des données.

La cryptographie est une technique qui sert à chiffrer des contenus, c'est-à-dire permettant de les rendre incompréhensibles sans une action spécifique, cette discipline est essentiellement basée sur le calcul (Mathématiques): Il s'agit dans le cas d'un texte de changer les lettres qui composent le message en une succession de chiffres (sous forme de bits dans le cas de l'informatique car le fonctionnement des ordinateurs est basé sur le binaire), puis ensuite de faire des calculs sur ces chiffres pour les modifier de telle façon à les rendre inintelligibles. Le résultat de cette modification (le message chiffré) est appelé cryptogramme (en anglais Ciphertext) par opposition au message initial, appelé message en clair (en anglais plaintext), et de faire en sorte que le destinataire saura les déchiffrer.

On peut donc dire que la cryptographie est une science à part entière qui croise les **mathématiques** l'**informatique**, et parfois même de la **physique**, elle permet en toute évidence le maintien du secret.

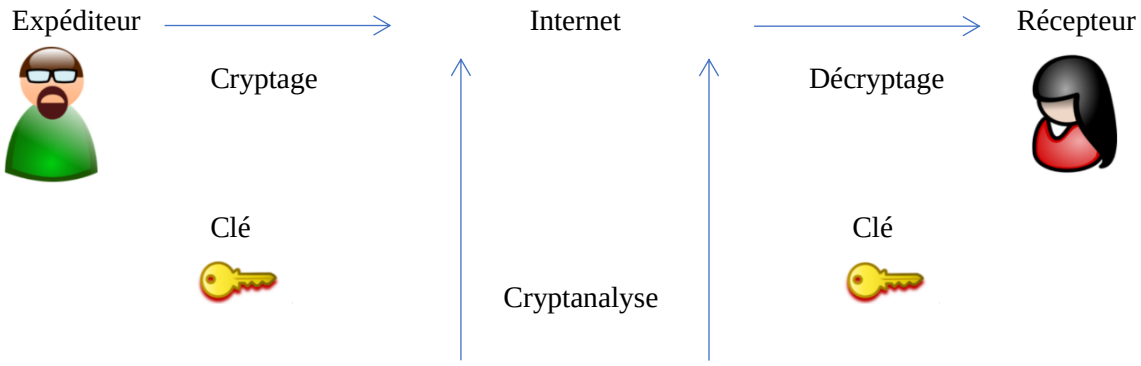


Figure 1.6: Principe de cryptographie.

1.3.1 Historiques des crypto systèmes :

La cryptographie n'est pas une technique neuve comme on l'en pense. Effectivement, l'homme a eu toujours le besoin de dissimuler ses informations secrètes. Evidemment depuis le début la cryptographie a amplement évolué.

Au travers les temps, plusieurs crypto systèmes ont été créés, mais au commencement les algorithmes étaient loin d'être aussi complexes et astucieux qu'à notre époque et on trouve que la contribution de l'informatique est remarquable dans l'amélioration de ces systèmes.

Au débuts, les méthodes de cryptage s'articulait sur deux principes radicaux : la substitution et la transposition.

1.3.1.1 Système de César

L'un des plus anciens et plus simples systèmes est le codage par substitution mono alphabétique (ou alphabets désordonnés). Il admet d'échanger toute lettre par une autre lettre. De cette façons on peut réaliser 26 manière de chiffrer un message, ce système a été beaucoup utilisé par les armées de l'ancienne époque.

Ce système était très sûr à l'époque est tout de même douteux car il nécessite que les interlocuteurs retiennent tous les deux la clé. De plus, il est quasiment simple nulle d'être déchiffré par n'importe quelle individu en mettant le temps qu'il le faut.

Le procédé le plus ancien admis par la substitution alphabétique est connu le code de César, consistant en un décalage simple de lettres. Par substitution si l'on remplace le A par le E, alors le B devient F, le D un G, etc.… César utilisait ce code simple pour communiquer via un message des instructions à ces généraux d'armées sans qu'il puisse être exploité par un quelconque rival dans le cas où le message serait décelé.

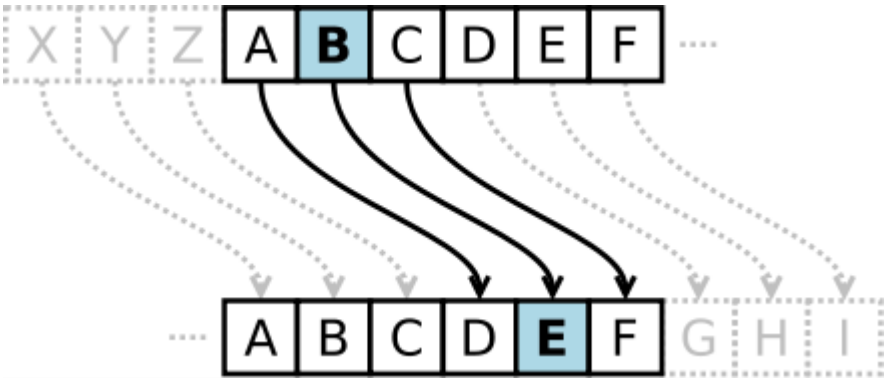


Figure 1.7: Principe du code de César.

1.3.1.2 Système de Vigenère

Un autre système de cryptographie dont la substitution est faite de plusieurs lettres alphabétiques. L'algorithme de substitution poly alphabétique plus connu sous l'appellation du code de Vigenère, mis au point par Blaise de Vigenère en 1586, qui fut utilisé pendant plus de 3 siècles. Son principe se repose sur le code de César, mais en changeant le décalage à chaque fois. Il utilise alors un carré composé de 26 alphabets alignés, décalés de colonne en colonne d'un caractère. Il place également au-dessus de ce carré, un alphabet pour la clé et à sa gauche un autre alphabet pour le texte à coder. Il suffit alors, pour chiffrer un message, de choisir un mot de longueur quelconque, de l'écrire sous le message à coder (de façon répétée s'il le faut) et de regarder dans le tableau l'intersection de la lettre à coder et de la lettre de la clé.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1.8: Table de Vigenère.

Il existe d'autres systèmes quasi aussi antiques fondés pareillement sur des techniques par remplacement mais moins connus que ceux vus précédemment. Il s'agit des systèmes par substitution de polygrammes. En effet au lieu de substituer des caractères, on substitue par exemple des diagrammes (des groupes de lettres le plus souvent). Le système de « Playfair »

inventé par Sir Charles Wheatstone, popularisé par L.Playfair utilise ce stratagème au moyen d'une table.

Nous citons aussi, le système ADFG(V)X, qui mélange transpositions et substitutions, il faut alors , ranger les 26 lettres de l'alphabet et les 10 chiffres dans un tableau de 6 cases sur 6. Au-dessus et à côté de ce tableau est ajouté le mot ADFGVX.

1.3.2 Crypto systèmes modernes

1.3.2.1 Cryptographie à clés privés

La cryptographie à clés privées, appelée aussi cryptographie symétrique est utilisée depuis déjà longtemps. C'est l'approche la plus authentique du chiffrement de données et qui pose le moins de problèmes mathématiques. La clé employée pour crypter les données peut être aisément déterminée si l'on connaît la clé servant à décrypter et vice-versa.

Dans la plupart des systèmes symétriques, la clé de chiffrage et la clé de déchiffrage sont les mêmes. Les principaux types de crypto systèmes à clés privés utilisés aujourd'hui se distribuent en deux grandes catégories : les crypto systèmes par flots (Stream Cipher) et les crypto systèmes par blocs (Block Cipher).

a) Crypto systèmes par flots (Stream Cipher) :

Dans un crypto système par flots, le cryptage des messages se fait caractère par caractère ou bit à bit, par le biais des substitutions de type César générées hasardeusement : la taille de la clé est donc égale à la taille du message.

L'exemple le plus illustratif de ce principe est le chiffre de Vernam. Cet algorithme est aussi appelé « One Time Pad » (masque jetable), c'est à dire que la clé n'est utilisée qu'une seule fois. Voici un exemple simple de l'application du chiffre de Vernam.

Message en clair: "SALUT"

=> (conversion en binaire)

01010011 01000001 01001100 01010101 01010100

XOR

Clé (générée aléatoirement)

01110111 01110111 00100100 00011111 00011010

=

00100100 00110110 01101000 01001010 01001110

=> (conversion en caractère)

"Message chiffré: $6jJM"

Le mathématicien Claude Elwood Shannon a prouvé l'impossibilité de retrouver un message crypté par le principe de Vernam sans connaître la clé. Ce qui ferait en théorie du chiffre de Vernam un crypto système infrangible. Mais en réalité, le Stream Cipher pose des difficultés relatives aux :

- Canaux sûrs de distribution des clés,
- Clés de tailles encombrantes et surtout caractère aléatoire des générateurs de bits de clés utilisées.

En revanche, un des avantages du système est qu'il est insensible aux phénomènes de diffusion d'erreurs : un bit erroné donne une erreur à la réception ou à l'émission, mais sans impact sur les bits suivants.

b) Crypto systèmes par blocs (Block Cipher)

Dans ce type de chiffrement, le texte clair est fragmenté en blocs de longueur identique à l'aide d'une clé unique. Les méthodes de cryptage par blocs sont généralement construites de façon itérative, où on emploie souvent une fonction F qui prend en paramètres une clé k et un message de n bits. F est réitérée un nombre de fois, on parle de ronde. A chaque ronde, la clé k employée est modifiée et le message que l'on code est le résultat de l'itération préalable.

$$C_1 = F(k_1, M)$$
$$C_2 = F(k_2, C_1)$$
$$\dots$$
$$C_n = F(k_n, C_{n-1})$$

Dans ce type de crypto système l'émetteur et le récepteur possèdent une clé K discrète. L'algorithme qui engendre les clés k_i à partir de K s'appelle l'algorithme de cadencement des clés.

c) Algorithme Data Encryption Standard (DES)

Le DES est une méthode de cryptage symétrique par blocs, divulgué en 1977 par le NBS (National Bureau of Standards), admettant de coder des mots de 64 bits à partir d'une clé de 56 bits (56 bits employés pour coder + 8 bits de parité vont servir à vérifier l'intégrité de la clé), le DES est préconisé pour les organisations à caractère fédéral, commercial ou privé. tire son origine des travaux menés par le groupe cryptographique d'IBM dans le cadre du projet LUCIFER.

d) Rijndael (AES)

En octobre 2000 la NIST (National Institute of Standards and Technology) adopte Rijndael, parmi 5 propositions : MARS, RC6,Rijndael, Serpent et Twofish, comme nouveau standard qu'on nomme aussi AES (Advanced Encryption Standard).

C'est une méthode de cryptage par blocs à plusieurs tours similaire à DES mais avec une taille de blocs et de clés supérieures et variables, choisis entre 128, 196 et 256 bits.

1.3.2.2 Cryptographie à clés publiques

L'idée de base des crypto systèmes à clés publiques a été présentée par ***Diffie et Hellman*** en 1976. Le principe de base est d'utiliser des clés de cryptage et décryptage distinctes, non déductibles l'une à partir de l'autre :

- une clé publique pour le cryptage
- une clé secrète pour le décryptage

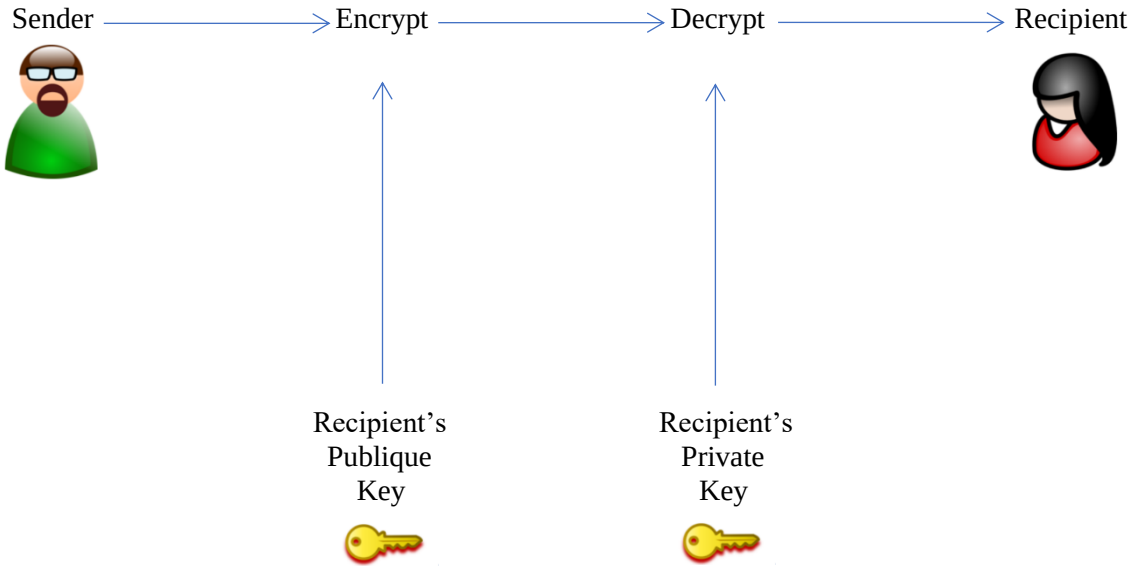


Figure 1.9 : Principe de cryptographie à clé publique.

Ce système est basé sur une fonction à sens unique, soit une fonction facile à calculer dans un sens mais très difficile à inverser sans la clé privée. Pour faire une explication imagée, la clé publique joue le rôle d'un cadenas. Imaginons que seul Bob possède la clé (clé secrète), Alice enferme son message dans une boîte à l'aide du cadenas et l'envoie à Bob. Personne n'est en mesure de lire le message puisque seul Bob possède la clé du cadenas. Le gros avantage de ce système est qu'il n'y ait pas besoin d'avoir partagé un secret au préalable pour s'échanger des messages cryptés. En revanche les implémentations de tels systèmes (RSA, ElGamal,…) ont un inconvénient majeur : leur lenteur par rapport à leurs homologues à clés secrètes qui tournent eux jusqu'à près de mille fois plus vite.

1. RSA

L'algorithme le plus célèbre d'algorithme à clé publique a été inventé en 1977 par Ron Rivest, Adi Shamir et Len Adleman, à la suite de la publication de l'idée d'une cryptographie à clé publique par Diffie et Hellman. Il fut appelé RSA, des initiales de ces inventeurs. RSA est basé sur la difficulté de factoriser un grand nombre en produit de deux grands facteurs premiers. L'algorithme fonctionne de la manière suivante : Imaginons que Bob souhaite recevoir d'Alice des messages en utilisant RSA.

1. **génération des clés** :
   a. p et q, deux grands nombres premiers sont générés au hasard grâce à un algorithme de test de primalité probabiliste, avec n = pq.
   b. Un nombre entier e premier avec (p-1)(q-1) est choisi. Deux nombres sont premiers entre eux s'ils n'ont pas d'autre facteur commun que 1.
   c. L'entier d est l'entier de l'intervalle [2, (p-1)(q-1)[tel que ed soit congrue à 1 modulo (p-1)(q-1), c'est-à-dire tel que ed-1 soit un multiple de (p-1)(q-1).
2. **distribution des clés** : le couple (n, e) constitue la clé publique de Bob. Il la rend disponible à Alice en lui envoyant ou en la mettant dans un annuaire. Le couple(n, d) constitue quant à lui sa clé privée.
3. **chiffrement du message** : Pour crypter le message Alice représente le message sous la forme d'un ou plusieurs entiers M compris entre 0 et n-1. Elle calcule $C= M^e \bmod n$ grâce à la clé publique (n, e) de Bob et envoie C à Bob.
4. **déchiffrement du message** : Bob reçoit C et calcule grâce à sa clé privée $C^d \bmod n$. Il obtient ainsi le message initial M.

1.4 Conclusion

Dans ce chapitre, on a essayé de rassembler les prérequis théoriques de deux domaines de recherche différents mais qui se croisent souvent lorsqu'on est appelés à traiter des informations avec de grands volumes nécessitant un mécanisme de stockage très efficace (avec un compromis taille/dégradation) et un transfert sécurisé qui fait appelle à des techniques très avancées afin d'assurer la confidentialité de ces informations.

Effectivement, la première partie de ce chapitre à été consacrée à un type très spécifique des information, à savoir les images. On a essayé d'introduire les notions de composition de l'image (pixel) comme point de départ, pour passer ensuite aux différentes méthodes de

compression d'images toute en passant par les mesures de performances utilisées pour favoriser une méthode de compression par rapport aux autres selon le contexte d'utilisation.

La deuxième partie a été consacrée au volet de sécurisation de l'information, où on s'est penché sur la cryptographie qui joue un rôle primordial dans l'assurance des transactions et informations dans le monde entier, tout étant aujourd'hui informatisé.

Nous avons pu aussi voir un panoramas de méthodes de cryptage de plus anciennes au plus modernes tout en mettant l'accent sur les lacunes et les difficultés rencontrées dans chaque méthode, et cela pour mettre en évidence le choix d'une méthode au détriment d'une autre.

Dans le chapitre qui suit nous allons voir la mise en œuvre d'une possible combinaison entre la compression et la cryptographie afin d'optimiser et de sécuriser l'information qui sera manipulée et transmise via les réseaux de télécommunication.

Bibliographie

[1] : CARACTÉRISTIQUES D'UNE IMAGE NUMÉRIQUE, Sylvain Argentieri (ISIR).

[2]: D. A. Huffman. A Method of the Construction of Minimum Redundancy Codes. In IRE, volume 40, pages 1098_1101, 1952.

[3]: Paul G. Howard and Jeffery Scott Vitter. Arithmetic Coding for Data Compression. Technical Report Technical report, DUKE_TR_1994_09, 1994.

[4]: B. Fong, G.Y. Hong, and A.C.M Fong. Constrained error propagation for efficient image transmission over noisy channels. IEEE Transactions on Consumer Electronics, 48(1): 49_55, 2002.

[5]: A. Guyader, E. Fabre, and C. Guillemot. Joint source-channel turbo decoding of VLC encoded Markov sources. In GRETSI, septembre 2001.

[6]: B.-J. Shieh, Y.-S. Lee, and C.-Y. Lee. A new approach of group-based VLC codec system with full table programmability. IEEE Transactions on Circuits and Systems for Video Technology, 11(2): 210_221, 2001.

[7]: J. Ziv and A. Lempel. A universal algorithm for sequential data compression. IEEE Transactions on Information Theory, 23 :337_343, 1977.

[8]: Richard E. Woods Rafael C. Gonzalez. Digital Image Processing. Addison-Wesley Pub Co, ISBN: 0201180758, Paris, January 2002.

[9] : N. Moreau. Techniques de compression des signaux. Masson, Paris, 1995.

[10]: Kyung Sub Joo, D. R. Gschwind, and T. Bose. ADPCM encoding of images using a conjugate gradient based adaptive algorithm. In ICASSP - IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, volume 4, pages 1942_1945, 1996.

[11]: M. Babel, O. Déforges, and J. Ronsin. Décomposition pyramidale à redondance minimale pour compression d'images sans perte. In GRETSI, volume 1, 2003.

[12] : TECHNIQUES DE CRYPTOGRAPHIE, cours de licence informatique, Jonathan BLANC, Adrien DE GEORGES.

Chapitre 2 : Approche Proposée

Chapitre 2 : Approche proposée

2.1 Introduction

La compression et le cryptage de données sont deux technologies dont l'importance croit d'une manière exponentielle dans une myriade d'applications. Par contre, l'usage exagéré des réseaux informatiques afin de transmettre les données doit certainement obéir à un double objectif : la diminution du volume des données afin de désembouteiller le maximum possible les réseaux de communication et la confidentialité pour assurer une sécurité optimale des données transmises.

Dans ce sens et afin d'assurer l'optimisation et la sécurisation de la transmission et du stockage des images, un certain nombre de méthodes de cryptage et de compression d'image ont été développées au cours des dernières années. Pour assurer la sécurité et l'efficacité de la conduction de l'image numérique lors de la transmission et du stockage, ces méthodes de cryptage et de compression des images numériques sont combinées de telle sorte que le schéma de chiffrement et de compression des images vise une sécurité robuste et une compression efficace.

Les techniques de cryptage et de compression sont souvent utilisées d'une manière disjointe, la combinaison de ces deux procédés peut être faite des façons suivantes:

La première façon d'envisager un système de crypto-compression est d'effectuer une compression suivie d'une cryptographie « compression-cryptage » pour des raisons que les partisans de cette catégorie voient logiques et pertinentes :

- Un bon chiffrement devrait faire en sorte que toutes les données d'entrée (en particulier les données redondantes) apparaissent aléatoires. Mais la compression fonctionne en supprimant la redondance et ne fonctionne pas bien sur les données aléatoires.
- La compression fonctionne en réduisant la redondance des données. Une méthode de cryptanalyse courante est l'analyse de fréquence, qui repose sur la recherche de données répétées. Le comprimer devrait réduire son efficacité.

La deuxième alternative consiste à effectuer une cryptographie suivie d'une de compression « cryptage-compression » : lorsqu'on des informations sensibles à transmettre sur un réseau avec une bande passante faible, il est nécessaire d'inverser l'ordre c-à-d, crypter

les informations pour les cacher puis l'opérateur du réseau se charge de les comprimer afin de les transmettre via le réseau.

La troisième technique qui est d'actualité, est d'utiliser la compression et le cryptage en parallèle « compression-cryptage hybride » : en général dans ces types de techniques certains algorithmes spéciaux sont utilisés pour que les deux processus puissent être traités en une seule étape.

Pour se positionner bien dans le contexte, nous allons citer quelques travaux d'actualité qui se sont distingués les uns des autres par l'ordre d'application des techniques aussi par le type de compression et de cryptage utilisées dans le système de crypto-compression.

2.2 Travaux similaires

En 2006, William P et al. [1] présentent une méthode de cryptage partiel ou sélectif pour les images JPEG, qui est basée sur le cryptage de certains coefficients DCT quantifiés en basses et hautes fréquences. Ils ont combiné la compression et le cryptage afin de dissimuler complètement les informations visuelles de l'image et de voir l'image en basse résolution.

En 2010, Masmoudi et al. [2] proposent une nouvelle méthode qui emploie conjointement le cryptage avec la compression qui se base sur l'échange des sous-intervalles associés aux symboles d'un codeur arithmétique binaire de façon aléatoire tout en exploitant un générateur de nombres pseudo-aléatoire. Les résultats expérimentaux montrent que la technique proposée est efficace, sécurisée et conserve le taux de compression obtenu par le codage arithmétique et ceci quel que soit le modèle statistique employé : statique ou adaptatif.

En 2014, Sharma et al.[3] ont proposé une nouvelle méthode de crypto-compression fondée sur une basée sur une transformée de Fourier fractionnaire discrète (DFRFT) combinée avec une cryptographie symétrique (la compression ici est sans perte à l'aide de méthodes zig-zag, Run Length et Huffman). La méthode proposée améliore considérablement la sécurité des données ainsi que la qualité de l'image décompressée. Les tests numériques basés sur le rapport signal / bruit de crête (PSNR), le rapport de compression (CR) démontrent la validité et l'efficacité de ce schéma.

En 2015, Bobby J. et al. [4], dans ce système proposé, une chaîne codée est créée à partir d'une chaîne d'entrée de symboles et de caractères basée sur une technique de codage entropique comme le codage arithmétique qui peut être utilisé pour atteindre un niveau de

compression élevé dans les topologies de réseau actuelles pour l'échange de données avec plus de sécurité et de compression.

En 2017, [5] Dans ce travail, une nouvelle approche concernant l'intégration du cryptage RSA dans un processus de compression basée sur la TCD.

En 2018, [6] Hajjaji et al, proposent un nouvel algorithme de crypto-compression d'images médicales basé sur les réseaux de neurones artificiels (RNA) et le système chaotique. L'objectif principal de cet algorithme est d'améliorer la sécurité des images médicales et de préserver les informations qu'elles contiennent. Les auteurs proposent de compresser l'image à l'aide des réseaux de neurones artificiels. Ensuite, les cartes d'Arnold ont été utilisées pour mélanger la matrice de poids et enfin, les cartes chaotique linéaire par morceaux sont utilisées pour modifier la valeur de la couche cachée du réseau. L'algorithme proposé a été appliqué sur des images médicales, de différents types, comme l'IRM, les images échographiques et radiographiques. Les résultats expérimentaux, confirment les performances et l'efficacité de l'algorithme proposé en termes de sécurité et de qualité des images non compressées.

En 2019, [7], Mr. Iyad présente une approche adaptée qui combine une excellente compression, nommée SPIHT (Set Partitioning in Hierarchic Tree), avec un chiffrement sélectif intégré dans le cycle du processus de compression. L'approche est adaptée et capable d'être utilisée dans les WMSN et de limiter les ressources de ces appareils. Les résultats obtenus, prouvent la haute performance l'approche proposée avec un surcoût inférieur à 0, 2914% et un débit de transfert constant.

En 2020, [8] RODRIGUES et al. Ont proposé une nouvelle méthode de compression-cryptage d'images 2D dont la qualité est démontrée par une reconstruction précise d'images 2D à des taux de compression plus élevés. La méthode est basée sur la transformation d'ondelettes discrète DWT où des sous-bandes haute fréquence sont connectées avec un nouvel algorithme de crypto-compression Hexadata au stade de la compression et un nouvel algorithme de recherche à correspondance rapide au stade du décodage. La nouvelle méthode de crypto-compression comprend quatre étapes principales:

1) un DWT à cinq niveaux est appliqué à une image pour effectuer un zoom arrière sur la sous-bande basse fréquence et augmenter le nombre de sous-bandes haute fréquence pour faciliter le processus de compression;

2) L'algorithme de compression de données Hexa est appliqué à chaque sous-bande haute fréquence indépendamment en utilisant cinq touches différentes pour réduire chaque sous-bande à 1/6 de sa taille d'origine;

3) Construire une table de correspondance des données de probabilité pour permettre le décodage des sous-bandes haute fréquence d'origine, et

4) Appliquer un codage arithmétique aux sorties des étapes (2) et (3). Au stade de la décompression, un algorithme de recherche à correspondance rapide est utilisé pour reconstruire toutes les sous-bandes haute fréquence. Nous avons testé la technique sur des images 2D, y compris le streaming à partir de vidéos (YouTube). Les résultats montrent que la méthode de crypto-compression proposée donne des taux de compression élevés jusqu'à 99% avec des images de haute qualité perceptuelle.

Dans cette recherche [9], Jean-Claude B et al. ont proposé deux crypto-systèmes, le premier d'entre eux est un algorithme très rapide de chiffrement par blocs, le TEA (Tiny Encryption Algorithm) et le second est un chiffrement par flots constituant une variante du chiffrement de Vigenère. Nous indiquons les différences qui existent entre ces deux systèmes, en particulier en ce qui concerne la combinaison du cryptage d'images et de la compression. Des résultats appliqués à des images médicales illustrent les deux méthodes.

Les méthodes de crypto compression se multiplient du point de vue de la nécessité de ce type de systèmes pour optimiser et sécuriser le transfert des informations via les réseaux de télécommunication. Nous avons pu voir quelques systèmes (les plus récents) parmi tous ceux qui existent pour prendre idée sur la forme que prendra notre système de crypto-compression.

2.3 Système de crypto-compression proposé

Pour notre système de crypto-compression on a proposé de combiner la méthode de cryptage Stream Cipher avec un algorithme de chiffrement RC4 et une compression sans perte RLE .

2.3.1 L'algorithme RC4

L'algorithme RC4, pour "Rivest Cipher 4", RC4 est un protocole de génération de bits pseudo aléatoires, permet, à partir d'une clé secrète, d'obtenir une suite de bits qui servent à encoder un message en faisant simplement un XOR bit à bit avec les données du message. Plus précisément, étant donnée K (la clé) un tableau d'octets de longueur Len ≤ 256,

l'algorithme commence par générer une permutation S aussi aléatoire que possible à l'aide de l'algorithme suivant :

1. La clef RC4 permet d'initialiser un tableau de 256 octets en répétant la clef autant de fois que nécessaire pour remplir le tableau.
2. Des opérations très simples sont effectuées : les octets sont déplacés dans le tableau, des additions sont effectuées, etc.
3. Le but est de mélanger autant que possible le tableau.
4. On obtient une suite de bits pseudo-aléatoires.
5. Cette suite peut être utilisé pour chiffrer les données via l'opération XOR.

Pour le déchiffrement on applique le même algorithme de chiffrement puisque l'opération XOR est une opération symétrique.

Algorithme KSA

```
function [ S ] = KSA( key )

key = char(key);
key = uint16(key);

key length = size(key,2);
S=0:255;

j=0;
for i=0:1:255
    j =  mod( j + S(i+1) + key (mod(i, key length) + 1), 256);
    S ([i+1 j+1]) = S ([j+1 i+1]);
end

end
```

Algorithme PRGA (génération de flux de clés)

```
function [ key ] = PRGA( k, n )
S = KSA(k);
i = 0;
j = 0;
key = uint16([]);
%each iteration we will append one key value
while n> 0
    n = n - 1;
    i = mod( i + 1, 256);
```

```
    j = mod(j + S(i+1), 256);
    S([i+1 j+1]) = S([j+1 i+1]);
    K = S( mod( S(i+1) + S(j+1)   , 256)  + 1  );
    key = [key, K];
end
key = uint8(key);
end
```

2.3.2 L'algorithme RLE

La compression RLE pour 'Run Length Encoding' est une compression sans perte d'information, elle consiste à remplacer les suites de caractères ou bits pareils par un nombre représentant le nombre de répétitions de caractères ou bits suivit de celui-ci.

Algorithme de compression RLE

```
function compressData = Compress(pic)
tic;
countVariable =1;
compressData1=zeros(10,1);
column=size(pic,2);
row= size(pic,1);
r=2;
compressData(1,1)=row;
compressData(1,2)=column;
k=1;
for j=1:column
    for i=1:row
      compressData1(k,1)=pic(i,j);
      k=k+1;
    end
end
   for k=1:size(compressData1,1)
      if size(compressData1,1) == k
          compressData(r,1)=compressData1(k,1);
            compressData(r,2)=countVariable;
      elseif compressData1(k,1)== compressData1(k+1,1)
            countVariable=countVariable+1;
       else
            compressData(r,1)=compressData1(k,1);
            compressData(r,2)=countVariable;
            countVariable=1;
            r=r+1;
```

```
        end
    end
t=toc;
end
```

Algorithme de décompression RLE

```
function samepic = deCompress(pic)
g = size(pic,1);
row = pic(1,1);
column = pic(1,2);
newArray=zeros(row,column);
a=0;
countArray=1;
count=1;
 for k=2:(g)
     for o=1:pic(k,2)
     if(count==row+1)
        countArray=countArray+1;
        count=1;
     end
     newArray(count,countArray)=pic(k,1);
     count=count+1;
     end
 end
     samepic=newArray;
 end
```

2.4 Architecture du système proposé

Pour notre système on va essayer d'appliquer les deux techniques de compression et de cryptage dans un ordre où le cryptage sera en premier lieu, suivi de la compression et la décompression et enfin le décryptage. Donc l'architecture du système va être comme la figure 2.1 le montre :

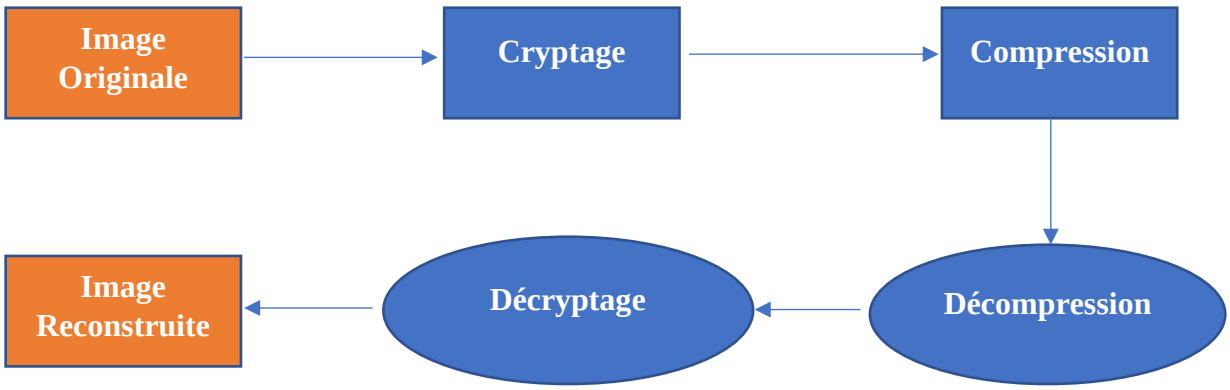


Figure 2.1 : Architecture générale du système.

Une autre idée nous est survenue, est celle d'inverser les modules de cryptage et de compression et voir l'impact sur les résultats l'

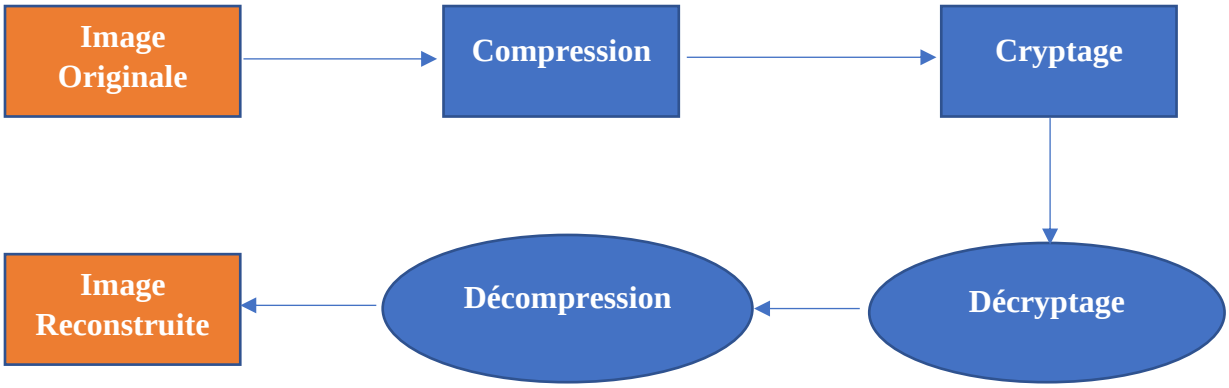


Figure 2.2 : Deuxième variante du système.

2.4.1 Le module de cryptage : comme son nom l'indique, ce module permet de crypter l'image sélectionnée par l'utilisateur avec la méthode de chiffrement Stream Cipher avec un algorithme RC4, ceci va nous aider à continuer le reste du processus dans un contexte plutôt sécurisé.

2.4.2 Le module de compression : après la sécurisation des données, une compression sans perte de type RLE est appliquée à celles-ci afin de réduire la taille sans pour autant les perdre.

2.4.3 Le module de décompression : la décompression intervient pour restaurer l'image cryptée (pas encore claire) après sa transmission bien sûr.

2.4.4 Le module de décryptage : la dernière étape vient pour reconstruire l'image originale chez le récepteur.

2.5 Conclusion

Dans ce chapitre on a effectué un survol sur les méthodes les plus récentes de combinaison de la compression et le cryptage.

On a essayé de proposer une méthode basée sur un algorithme RC4 associé à une compression sans perte de données RLE dans deux ordres différents.

A première vue et théoriquement parlé les deux variantes de notre système ont le même impact sur l'image d'entrée, reste à prouver ceci en calculant les mesures de performances et les comparées.

Bibliographie

[1] : William Puech, José Marconi Rodrigues. Crypto-Compression d'Images Médicales par Cryptage Partiel des Coefficients DCT. JSTIM: Journées Sciences Technologies et Imagerie pour la Médecine, Mar 2005, Nancy (France), pp.149-150. lirmm-00106477.

[2] : Masmoudi, Atef. Elaboration et analyse de nouveaux algorithmes de crypto-compression basés sur le codage arithmétique. Montpellier : 2010. Université de Montpellier 2, Université de Sfax (Tunisie) : thèse de doctorat, Informatique.

[3] : Deepak Sharma, Rajiv Saxena, and Narendra Singh. Hybrid encryption compression scheme based on multiple parameter discrete fractional fourier transform with eigen vector decomposition algorithm. International Journal of Computer Network & Information Security, 10 :1–12, 2014.

[4]: Crypto-Compression System: An Integrated Approach using Stream Cipher Cryptography and Entropy Encoding Bobby Jasuja, Abhishek Pandya, Published 2015, Computer Science, International Journal of Computer Applications.

[5]: Med Karim Abdmouleh and Med Salim Bouhlel , Nouvelle Approche basée sur la TCD Pour la Crypto-Compression des Images Médicales : Application à la Télémédecine, 29th IBIMA Conference, 2017.

[6] : Hajjaji, M.A., Dridi, M. & Mtibaa, A. A medical image crypto-compression algorithm based on neural network and PWLCM. Multimed Tools Appl 78, 14379–14396 (2019).

[7] : Iyad M. Hraini, Joint Crypto-Compression Based on Selective Encryption for WMSNs, Thesis submitted in partial fulfillment of the requirements of the degree Master of Science in Informatics, Palestine Polytechnic University

[8]: RODRIGUES Marcos and SIDDEQ Mohammed (2019). A Novel Hexadata Encoding Method for 2D Image Crypto-Compression. Multimedia Tools and Applications, 79 (9), 6045-6059.

[9]: Crypto-compression using TEA's algorithm and a RLC compression, Jean-Claude Borie, William Puech and Michel Dumas.

Chapitre 3 : Résultats et Discussions

3.1 Introduction

Ce dernier chapitre se compose de deux parties, la première partie se concentre sur la présentation de l'environnement de travail qu'on a utilisé, que ce soit coté matériel ou coté logiciel.

La deuxième partie se focalise sur les résultats et les interprétations des résultats obtenus par le système de crypto-compression réalisé.

3.2 Environnement de travail

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

3.2.1 Environnement matériel

Le développement de l'environnement matériel est caractérisé par :

- Système d'exploitation : Windows 10 Professionnel.
- CPU : Pentium M, 1.6 GHz.
- Mémoire : 4 Go.

3.2.2 Environnement logiciel

Pour implémenter notre système, le langage de programmation Matlab est le mieux adapté. En effet, Matlab 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.

3.3 Aperçu sur l'interface de système réalisé

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

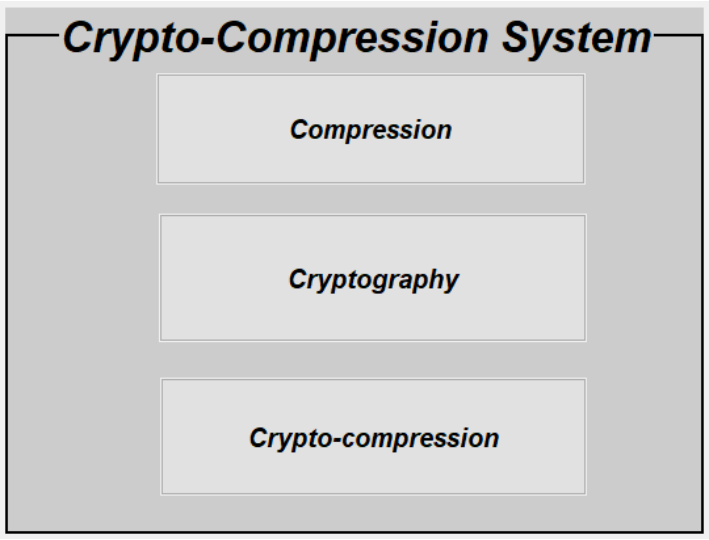


Figure 3.1 : L'interface de l'application réalisée

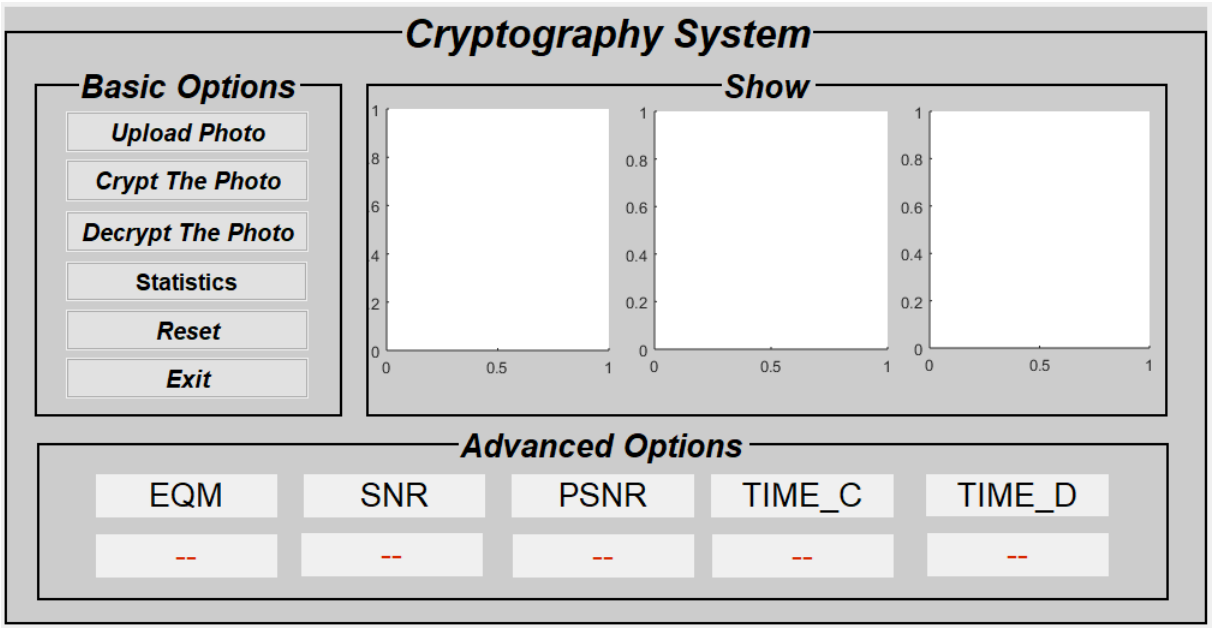


Figure 3.2 : Interface de cryptographie

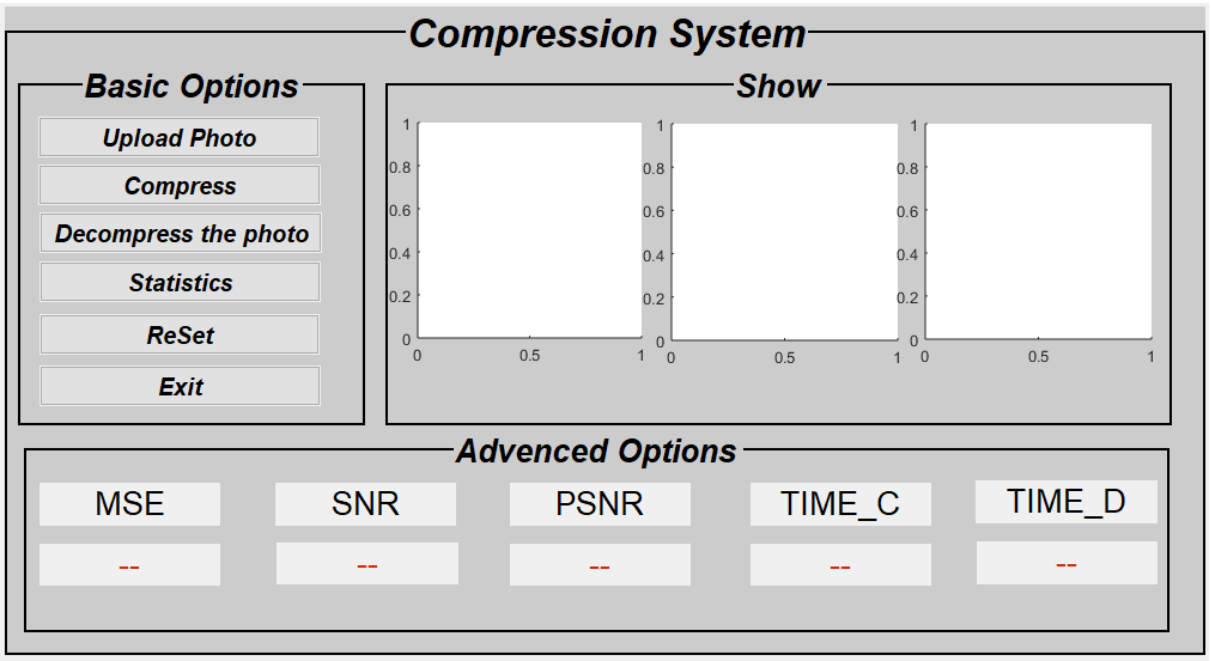


Figure 3.3 : Interface de compression

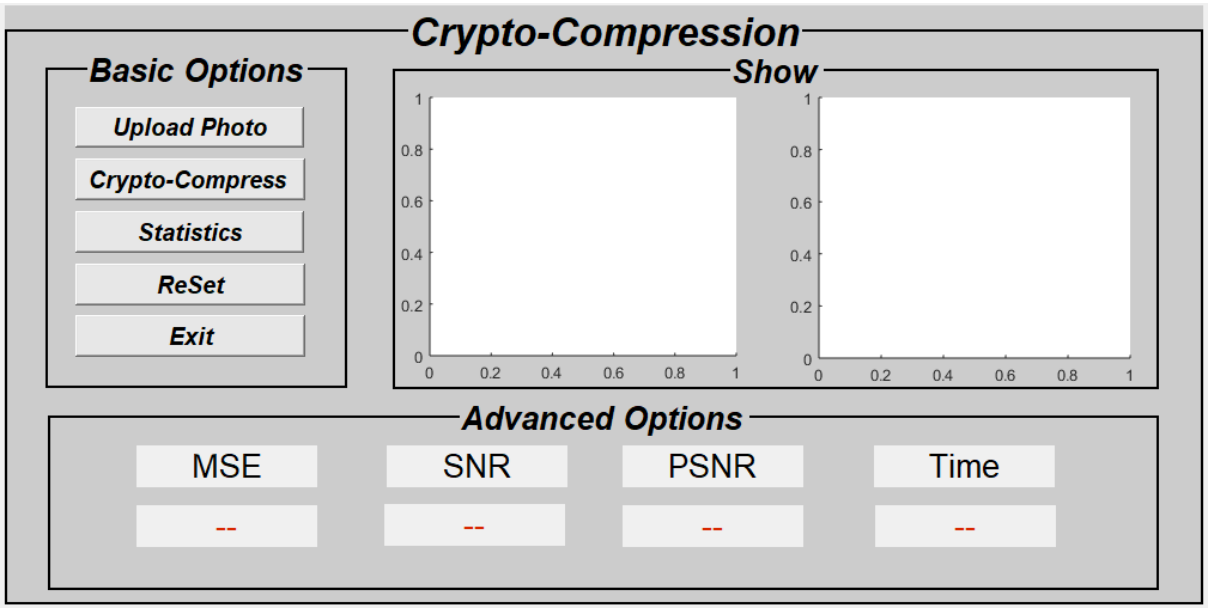


Figure 3.4: Interface de crypto-compression

3.4 Base d'images

Voici la collection des images avec lesquelles on a travaillé durant l'exécution de notre système :

Images de tests			
Nom : Lena **Taille Physique : 3.6 ko** **Dimension : 128 X 128** **Type de fichier : BMP**		**Nom : Avion** **Taille Physique : 3.55 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**	
Nom : Barbara **Taille Physique : 3.63 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**		**Nom : Clown** **Taille Physique : 3.58 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**	
Nom : Fruit **Taille Physique : 3.58 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**		**Nom : Pimen** **Taille Physique : 3.76 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**	
Nom : Baboon **Taille Physique : 4.11 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**		**Nom : House** **Taille Physique : 3.78 ko** **Dimension : 128 X 128** **Type de fichier réel : BMP**	

Tableau 3.1 : Base des images de tests

3.5 Tests expérimentaux

3.5.1 Résultats du première variante de système

Nous présentons dans ce qui suit, quelques résultats d'application du système proposé sur les images de la base de test.

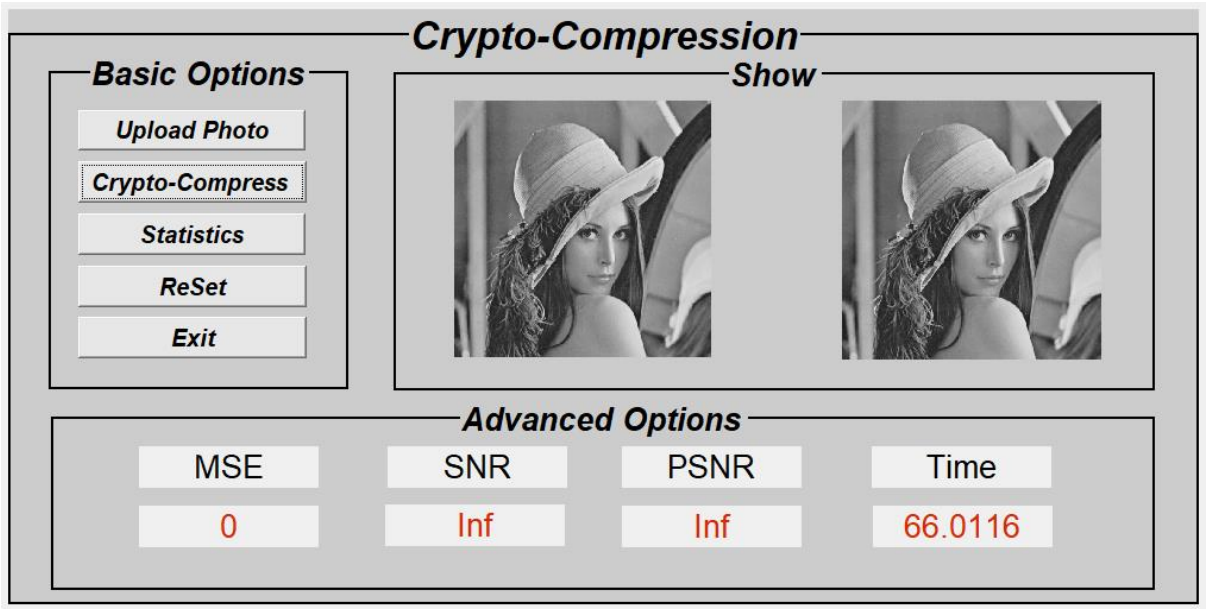


Figure 3.5 : crypto-compression de Lena

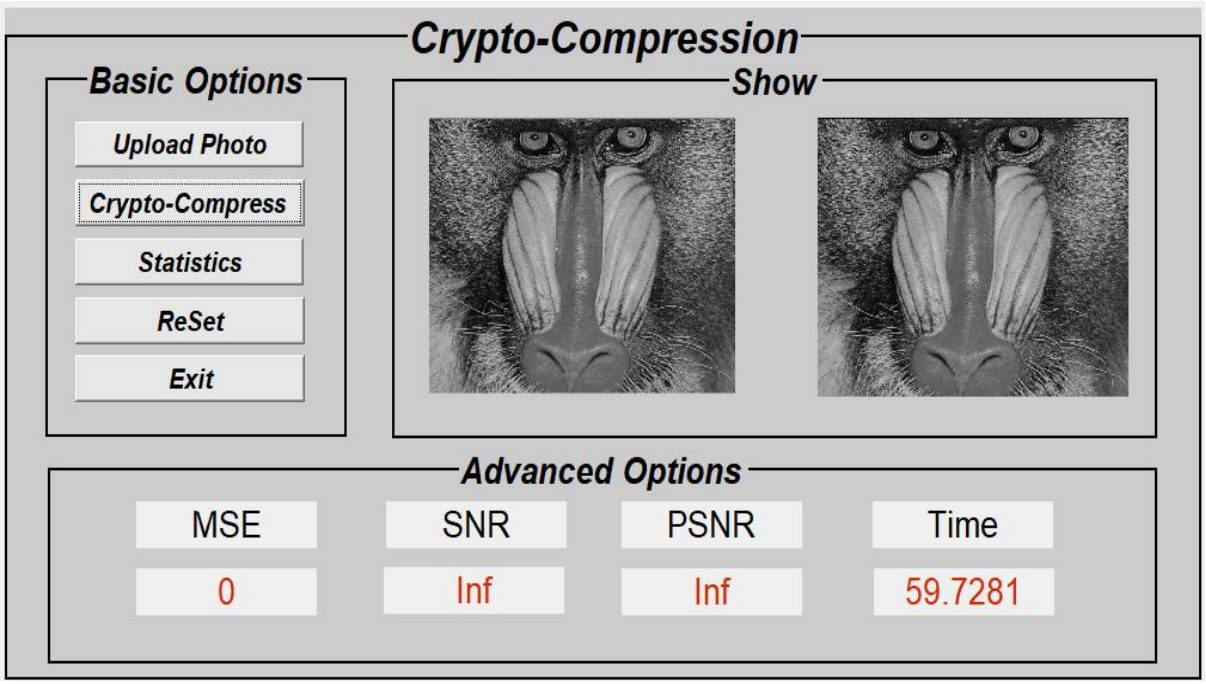


Figure 3.6 : crypto-compression de baboon

L'image				Paramètres de performance						
Nom	**Dimension**	**Type**	**Test**	**MSE (Err)**	**SNR**	**PNSR (db)**	**Temps Cryptage (S)**	**Temps compression (S)**	**Temps Décompression (S)**	**Temps Décryptage (S)**
Lena	128 X 128	BMP	01	0	INF	INF	1	60	4.01	1
Barbara	128 X 128	BMP	02	0	INF	INF	0.5	50	3	1.5
Clown	128 X 128	BMP	03	0	INF	INF	1.5	62.5	1	3
Avion	128 X 128	BMP	04	0	INF	INF	2	75.14	4	1
Fruit	128 X 128	BMP	05	0	INF	INF	0.5	89	3	1
House	128 X 128	BMP	06	0	INF	INF	1	55	3	2
Boat	128 X 128	BMP	07	0	INF	INF	1.25	68 .25	5	2
Pimen	128 X 128	BMP	08	0	INF	INF	2 .5	69	1	1

Tableau 3.2 : Résultats sur différentes images

Interprétation des résultats

- On remarque que le PSNR et le SNR sont des grands nombres (+inf dB) dans notre system de Crypto-Compression que l'approche appliqué pour les images avec un taux d'erreur égale à zéro ce qui nous prouve que notre système fonctionne parfaitement sans perte d'information.
- **Remarque** : Une valeur de PSNR inférieure à 30 dB traduit généralement une image présentant des dégradations perceptibles.
- Et concernant le temps que le système a pris pour la Crypto-Compression des images se varient selon la taille de l'image et le nombres de répétition des bites de cette image.
- Donc On a réussi a développé un system de Crypto- Compression d'image sans perte d'information et en toute sécurité qui peut être utilisé dans plusieurs domaine grâce à sa rapidité et sa sécurité.

3.5.2 Résultats du système de la deuxième variante

- Nous avons essayé d'appliquer le système d'une autre façon où on a appliqué la compression avant le chiffrement : D'après les résultats obtenus on a remarqué, qu'on a des problèmes dans la phase de chiffrement puisque le résultat de la compression est

un vecteur de deux dimensions. Cela a causé une erreur lors de l'opération XOR de l'algorithme de chiffrement.

3.6 Conclusion

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

Nous avons testé plusieurs types d'image à l'entrée de notre système, et nous avons enregistré les résultats.

Dernièrement on a discuté les résultats obtenus, et d'après les résultats présentés, on remarque bien qu'on a réussi a développer un système de crypto-compression sans perte d'information.

Conclusion générale

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

On a vu dan ce mémoire la complémentarité entre la compression et la cryptographie, on a vu aussi à travers plusieurs travaux que la combinaison entre les deux axes est possible et le système résultant s'appelle un système de crypto-compression.

Les systèmes de crypto-compression existants on combiné la compression et la cryptographie dans des ordres différents selon le contexte d'application de ce système.

Dans ce mémoire de Master on s'est intéressé à la combinaison de ces deux techniques, à savoir l'objectif principale est de mettre en œuvre un nouveau crypto système basé sur une compression sans perte RLE et un algorithme de cryptage RC4 fondé sur la méthode de chiffrement par flots (Stream Cipher).

D'après les résultats obtenus, on a pu constater que l'ordre de combinaison dans le système proposé est obligatoirement imposé comme suit : Cryptage → Compression → Décompression → Décryptage.

On a vu aussi que le cryptage n'affecte pas l'image reconstruite, mais aussi la compression n'a aucun effet sur le décryptage.

On a essayé d'inverser la compression et le cryptage, malheureusement, ça n'a pas marché, faute des résultats engendrés par l'étape de cryptage.

Nous envisageons d'essayer de trouver une solution pour que la deuxième variante de notre système soit exploitable en adaptant les résultats des différentes étapes afin qu'il puissent être exploitables par les étapes suivantes.