MA-004-73-1

Université Saâd DAHLAB de Blida

Faculté des sciences

Département d'informatique

En vue d'obtenir le diplôme de **Master**

Domaine : Sécurité informatique.

Filière : Informatique.

Spécialité : Informatique.

Option : Génie des logiciels.

Sécurisation des données biométriques par tatouage numérique.

Mémoire Présenté par :

Mr : BENAISSA Zakaria.
Mr : BENKORTBI Abdenour.

Promotrice : Mme Karima Ait Saadi

Organisme d'accueil : CDTA (Centre de Développement des Technologies Avancées).

Division : Architecteur Des Système Multimédia

Soutenue le:....., devant le jury composé de :

Nom. Président du jury, M..Benoour	Président
Nom examinateur 1, ..Z.ai..Mostapha	Rapporteurs
Nom examinateur 2, .Me...Re.Zoug	Examinateurs

2010 - 2011

Sommaire

CHAPITRE VI : tests et résultats.

Liste des tables

CHAPITRE I : Le tatouage numérique.

CHAPITRE II : Wavelet Scalar Quantization (WSQ)

CHAPITRE III : la protection des images biométriques par tatouage numérique.

CHAPITRE VI : tests et résultats.

Introduction Générale

Depuis quelques années, la croissance mondiale des technologies biométriques est incontestable, elle présenterait des revenus globaux de l'ordre de 5.7 Milliard dollars en 2010 selon (International Biometric Group) [1], permettant d'obtenir une meilleure sécurité et une diminution des risques de fraude à l'identité. Cette importante amélioration de la sécurité apportée par la biométrie conduit aujourd'hui les gouvernements à lancer un projet ambitieux dont le but est d'établir des nouveaux titres d'identité biométriques (carte d'identités et passeports biométriques). Ces titres comportent, outre les informations démographiques de l'individu, des modalités biométriques, à savoir : photo de visage, signature numérique et empreintes digitales. Ces dernières seront gérées par un système AFIS (système d'identification des images biométriques), où ses différentes stations de prélèvement seront déployées au niveau des daïras à travers le territoire national.

En effet, le système AFIS à base des images biométriques (le visage et d'empreinte digitale) comporte un nombre important d'images stockées dans les bases de données. Ces images sont généralement des fichiers volumineux et leur exploitation telles qu'elles sont, influent sur le temps de leur transmission et aussi sur la capacité du stockage (réduction de l'espace de stockage). Pour remédier à ça, le FBI a développé une technique de compression basée sur la transformée en ondelette, dite WSQ (Wavelet Scalar Quantization), qui permet de réduire considérablement la taille des images en introduisant de faibles déformations visibles. Cette technique, qui est devenue le standard d'échange des images d'empreintes digitales, est actuellement adoptée par toutes les agences de police (FBI, Interpol, etc.) et est utilisée dans les systèmes professionnels biométriques à base d'images biométriques.

Néanmoins, ces systèmes ne sont pas à l'abri des erreurs et des attaques qui tentent d'exploiter les failles pour déstabiliser ces systèmes. L'une des attaques ou erreurs jugée fatale est la modification ou suppression des codes des fichiers d'images biométriques, car en analysant ces fichiers, on constate que les codes de ces fichiers sont les seuls liens entre les personnes (plus précisément, l'identité de la personne) et ces images. En d'autres termes, les images biométriques sont enregistrées en utilisant des codes qui ne sont que l'identificateur de l'individu, alors si ces codes sont modifiés, le système ne pourra jamais extraire les fichiers associés. Ce qui causera un disfonctionnement fatal du système.

Une solution possible à ce problème est d'utiliser les techniques du tatouage numérique. Le tatouage numérique est une approche de sécurité très récent qui consiste à insérer une donnée

secrète (code, matricule, nom, prénom, etc.), d'une façon invisible dans la donnée à protéger. Seul le propriétaire de la donnée connait l'existence, l'emplacement et la nature de la donnée insérée. Cette approche est essentiellement utilisée pour sécuriser les données multimédia (image, son, vidéo).

Le but de ce projet est la conception et le développement d'un module pour sécuriser les images biométriques compressées par le standard WSQ du FBI en utilisant le tatouage numérique. Ce projet s'inscrit dans le projet « Sécurisation des documents multimédias par tatouage numérique » de la division Architecture des Systèmes et Multimédia (ASM) du Centre de Développement des Technologies Avancées (CDTA).

La structure de ce mémoire reflète la logique de notre objectif. Il comprend quatre (04) chapitres qui sont présentés comme suit :

Chapitre I : Le tatouage numérique.

Dans le premier chapitre, nous allons présenter des généralités sur le tatouage numérique, son principe, ses domaines d'utilisation, la classification de ses techniques, ainsi que les caractéristiques. Il expliquera aussi comment les techniques de tatouage numérique peuvent être introduites pour conforter la sécurité des systèmes d'identification à base d'images biométriques.

Chapitre II : Le standard de compression WSQ.

Ce chapitre décrire le standard de compression WSQ, en expliquant et détaillés ses différents modules tels que : la transformé en ondelette discrète (TOD), la quantification et le codage de Huffman. De plus, la présentation du flux compressé (.wsq).

Chapitre III : La protection des images biométriques par tatouage numérique.

Une présentation détaillée du système de sécurisation d'images biométriques par tatouage numérique fera objet du troisième chapitre. Le système proposé comportera deux procédures à savoir : la procédure d'insertion et la procédure d'extraction de la marque.

CHAPITRE IV : Résultats et tests.

Ce chapitre commencera par présenter les différents résultats des tests effectués pour évaluer les performances de la méthode proposée. Suivie par la conception et l'environnement de programmation ainsi que l'interface de l'application et les différentes fonctionnalités offertes par le système.

CHAPITRE I :
LE TATOUAGE NUMERIQUE

I.1 Introduction

Dans le but de sécuriser la circulation des œuvres « copyright», le tatouage des images est apparu au début des années quatre-vingt-dix. Depuis ce temps, l'image a changé de forme et elle est passée au format numérique ce qui a facilité son traitement et sa manipulation mais en même temps la numérisation de l'image là rendu plus vulnérable aux attaques que peut subir n'importe quelle information numérique telles que : la modification, la distribution et la copie illégale. Face à ce changement, le tatouage des images est passé lui aussi au numérique pour devenir le tatouage numérique des images qui consiste à insérer une information invisible dite « *Marque* » dans une image, cette marque doit être suffisamment imperceptible pour ne pas infecter la qualité de l'image et assez robuste pour être récupérée après une éventuelle manipulation de l'image.

I.2 Historique

L'idée de cacher une information existait depuis bien longtemps, il existe quelques événements historiques qui illustrent ce concept, un exemple a été cité dans le livre « *The Histories of Herodotus* », où « *histiaeus* » a rasé le crane d'un de ses esclaves, et a tatoué un message sur son crane, il a attendu que ses cheveux grandissent, ainsi le message a été caché, et cela dans le but de lancer une guère contre l'empire perse [2].

Le premier travail du tatouage sur papier a été publie vers 1282 en Italie où les marques étaient constituées par l'addition d'un modèle de fil fin au module du papier. Le papier était devenue légèrement fin, et donc plus transparent. Néanmoins, le tatouage n'avait pas les mêmes objectifs que ceux de nos jours. Le but derrière l'insertion de la marque était incertain, la marque a été utilisée probablement pour identifier le moule avec lequel le papier a été fabriqué, ou pour représenter de signes de religion, ou simplement pour l'utiliser comme un outil de décoration [3].

Au début de 21ème siècle, l'utilisation du tatouage de papier est devenue de plus en plus claire en Europe et en Amérique et les marques servaient comme des marques de fabrication pour noter la date de fabrication du papier, et indiquer la taille d'origine de la feuille. Dans cette même période, les marques ont été

utilisées aussi comme moyen contre la falsification des billets monétaires et d'autres documents confidentiels [4].

Le terme marque « watermark » a été utilisé pour la première foi à la fin du 21ème siècle, et parait comme étant une dérivée de la langue allemande et précisément du terme « *mark* » qui veut dire la marque. Cette appellation de la marque ne veut pas dire que l'eau est nécessaire pour la création de la marque, mais probablement que la marque ressemble à l'effet de l'eau sur le papier [3].

En 1954, Emil Hembrooke de la Muzak Corporation a rangé un tatouage dans un travail musical. Un code d'identification a était inséré dans la musique par un filtre passe-bas à une fréquence de 1KHz [5].

En 1979, Szrpanski a décrit une machine qui détecte des modèles placés dans des documents. Neuf ans plus tard, Holt et al. ont décrit une méthode pour insérer un code d'identification dans un signal audio [5]. Et c'était en 1988 que Komatsu et Tominga ont utilisé le terme tatouage numérique pour la première fois [6].

I.3 Définitions

Le tatouage numérique, digital watermarking en anglais, consiste à insérer une information secrète dite « *marque* » dans une donnée numérique (image, son, vidéo...), dite « *hôte* » ou « *porteuse* ». L'insertion s'effectue dans les composantes perceptibles (comme la luminance des pixels d'une image), et non dans l'entête d'un fichier par exemple. La marque insérée doit pouvoir être détectée et décodée, mais doit être imperceptible, c'est-à-dire que la déformation doit être suffisamment faible pour que l'être humain ne puisse pas différencier la donnée tatouée de la donnée d'origine [6].

Le contenue d'une marque, typiquement quelques bits d'information, peut être de différentes natures, dont les plus utilisées sont :

(i) informations sur le propriétaire de la données (par exemple : Id, nom ou logo).
(ii) informations sur les permissions attachées à la donnée (par exemple : copier illimité, interdite).
(iii) information sur la (les) personne(s) ayant accès à la donnée (c-à-d : la personne à qui le propriétaire a donné une copie) [7].

Un modèle général d'un système de tatouage numérique est représenté dans la figure I.1. La fonction d'insertion de la marque (f), qui a comme entrées la marque *b* et la clé secrète *k*, est appliquée à la donnée d'origine *I*, afin d'obtenir la donnée tatouée *I'*. Cette dernière est transmise dans un canal qui peut être non sécurisé. Lors de cette transmission, d'éventuelles attaques ou modifications peuvent surgir et la donnée tatouée *I'* peut être changée en *I''*. A la réception de *I''*, le destinataire peut effectuer deux opérations à savoir :

(i) détecter l'existence de la marque à l'aide de la fonction *d* qui a comme entrées *I''* et la clé *k*, et en sortie une valeur booléenne (0,1) qui nous renseigne sur la présence/absence d'une marque dans la donnée reçue *I''*.

(ii) extraire la marque *b'* à l'aide de la fonction d'extraction (*e*) à partir de la donnée reçue *I''* en utilisant la clé secrète *k* utilisée dans l'insertion de la marque [3].

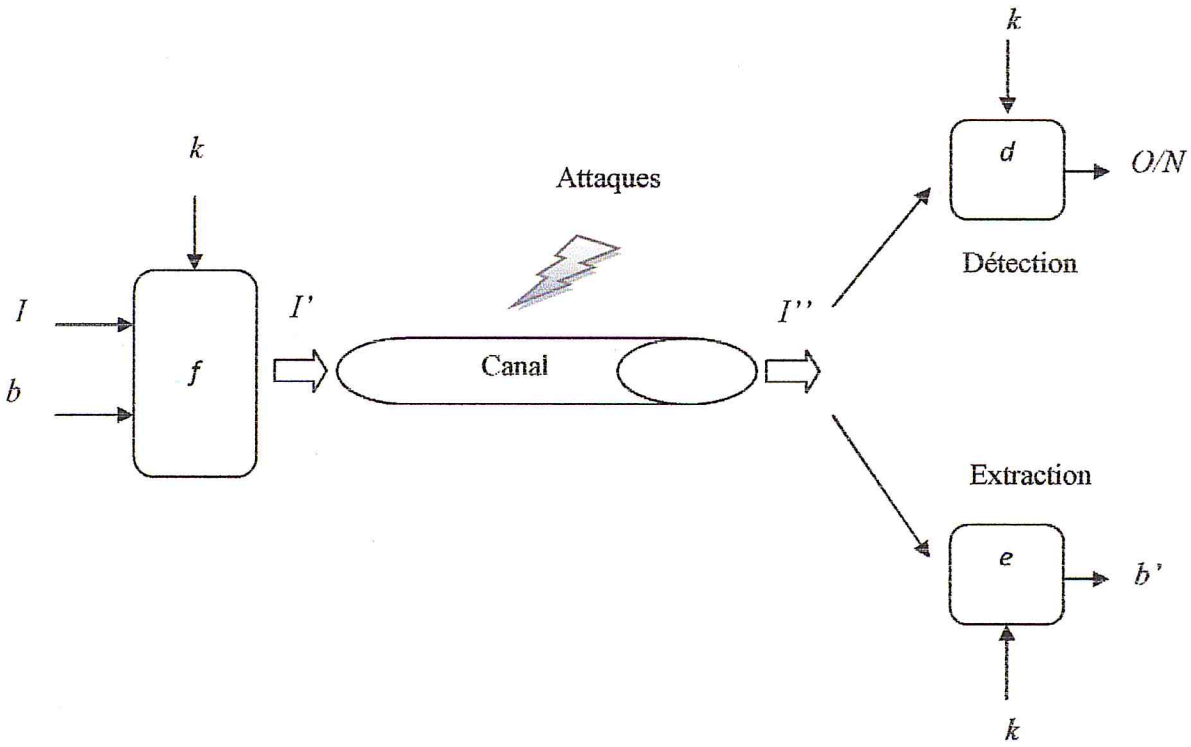

Figure I.1 Modèle général d'un système de tatouage numérique.

I.4 Applications du tatouage numérique

Le domaine d'application du tatouage numérique est vaste. Les principales applications du tatouage numérique sont décrites ci-dessous :

- **Protection des droits d'auteur (copyright)**

C'est l'une des applications la plus porteuse : le tatouage numérique offre une alternative intéressante à la cryptographie, car il permet de protéger le document, même lorsque celui-ci est diffusé. La protection des droits d'auteur, quant à elle, l'application la plus courante. L'objectif est d'incruster une information dans la donnée source, typiquement le copyright du propriétaire, afin de prévenir toute revendication frauduleuse de propriété. Cette signature ne doit pas être connue que de la personne et de l'organisme de tatouage. Elle dépend donc d'une clé secrète, qui permet son insertion et sa détection. Cette application nécessite la mise en place d'un algorithme ayant une robustesse très élevée [6].

- **Traçage des copies des pirates**

Le but de cette application est de pouvoir contrôler et faire le suivi des copies de document et tracer les copies illégales de ce dernier (suivi des pirates). Ce type d'application engendre un marquage unique pour chaque document distribué. Cette application nécessite d'un algorithme de tatouage robuste, pour pouvoir résister à des attaques ayant pour but de détruire la marque [2].

- **Contrôle de copie et de lecture**

Cette application vise à protéger les supports de stockage (CD et DVD) contre la copie ou la lecture illégale. Le message porté par la marque comporte des informations concernant la permission à la copie et/ou la lecture des données stockées. Un module sécurisé peut être rajouté aux équipements (lecteurs CD et DVD) pour extraire la marque [8].

- **Authentification**

L'objectif de cette application est de détecter si la donnée ou le document a subi des manipulations. Ceci peut être fait en utilisant une marque fragile. Si le document d'origine est modifié ou manipulé, la marque sera détruite, dans ce cas,

on peut certifier que le document a été modifié et donc il n'est plus authentique [2].

- **Indexation**

L'indexation des documents consiste à classer de manière automatique ces documents selon leurs contenus, et cela dans le but de faciliter la recherche et l'accès à ces documents. Les techniques classiques utilisées consistent à effectuer un traitement automatique du document, de manière à dégager les composantes essentielles du contenu. Alors que, le tatouage d'un document permet aussi d'insérer une information (contenant peu de bits) permettant de garantir sommairement son contenu ou d'insérer un pointeur vers une description plus complète [9].

I.5 Caractéristiques du tatouage numérique

Les caractéristiques du tatouage numérique diffèrent d'une application à une autre selon les besoins de ces applications. Certaines propriétés sont définies pour un grand nombre d'applications telles que :

I.5.1 Imperceptibilité

Elle désigne la similarité perceptuelle entre la donnée d'origine et celle tatouée. La procédure d'insertion doit assurer que la marque est imperceptible pour toutes les personnes. La marque est vraiment imperceptible si les humains ne peuvent pas faire la différence entre la donnée d'origine et la copie tatouée.

Certaines approches, dites perceptuelles, ont été proposées afin d'assurer l'imperceptibilité en exploitant les caractéristiques du Système Visuel Humain (SVH) dans le cas des images et celles du Système Audible Humain (SAH) dans le cas des signaux audio [8].

I.5.2 Robustesse

Etre robuste c'est équivalent à pouvoir retrouver la marque insérée quelles que soient les modifications que peut subir l'objet marqué. Ces modifications sont dues à plusieurs types de traitements (attaques) dont deux sont essentiels. Le premier concerne les attaques simples telles que la compression, le filtrage, l'addition de bruit, l'impression, etc. Cette classe ne change pas la géométrie de

l'image alors que la deuxième catégorie la modifie. On y trouve la rotation, le changement d'échelle et les transformations affines. La plupart des systèmes de marquage résistent à ces transformations simples mais ne supportent pas la combinaison de plusieurs d'entre elles [8].

I.5.3 Capacité

Cette notion désigne le nombre de bits qui peuvent être insérés dans le document à protéger tout en respectant les autres propriétés. Bien qu'une grande capacité soit désirée, son insertion cause des dégradations en termes de robustesse et d'imperceptibilité. Plus la capacité est grande, plus la qualité perceptuelle de la donnée tatouée sera dégradée et moins robuste sera la marque insérée [10].

I.5.4 Sécurité

Cette caractéristique représente la sécurité de l'information insérée. Une technique du tatouage numérique est dite sécurisée, si la connaissance exacte des processus d'insertion et d'extraction de la marque ne sert pas à enlever ou détruire la marque par une personne non-autorisée. Dans la plupart des systèmes, cette sécurité est assurée en utilisant des clés secrètes et pseudo-aléatoires. Ces clés servent à générer la séquence représentant la marque ou à déterminer les emplacements ou la marque est insérée [8].

Les trois premières caractéristiques sont fortement liées les unes aux autres et un schéma de tatouage numérique s'inscrit donc dans un compromis entre l'imperceptibilité, la robustesse et la capacité. (Une application de tatouage numérique très robuste a une très grande variété d'attaques ne peut pas avoir une capacité très importante (pour les techniques visant la protection des droits d'auteurs, souvent 64 bits sont utilisés) [10]. A l'inverse, une méthode permettant de cacher plusieurs milliers de bits dans une image de taille standard ne peut pas être très robuste. Ce compromis est illustré dans la figure I.2.

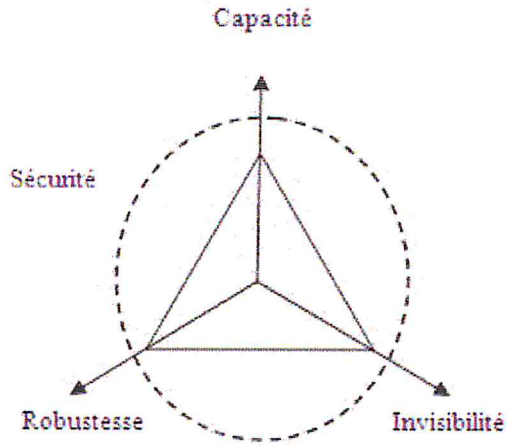

Figure I.2 Illustration graphique du compromis existant entre les caractéristiques du tatouage numérique.

I.6 Classification des systèmes de tatouage numérique

Les systèmes du tatouage numérique sont classifiés selon plusieurs critères, ces critères sont représentent dans la figure II.3 :

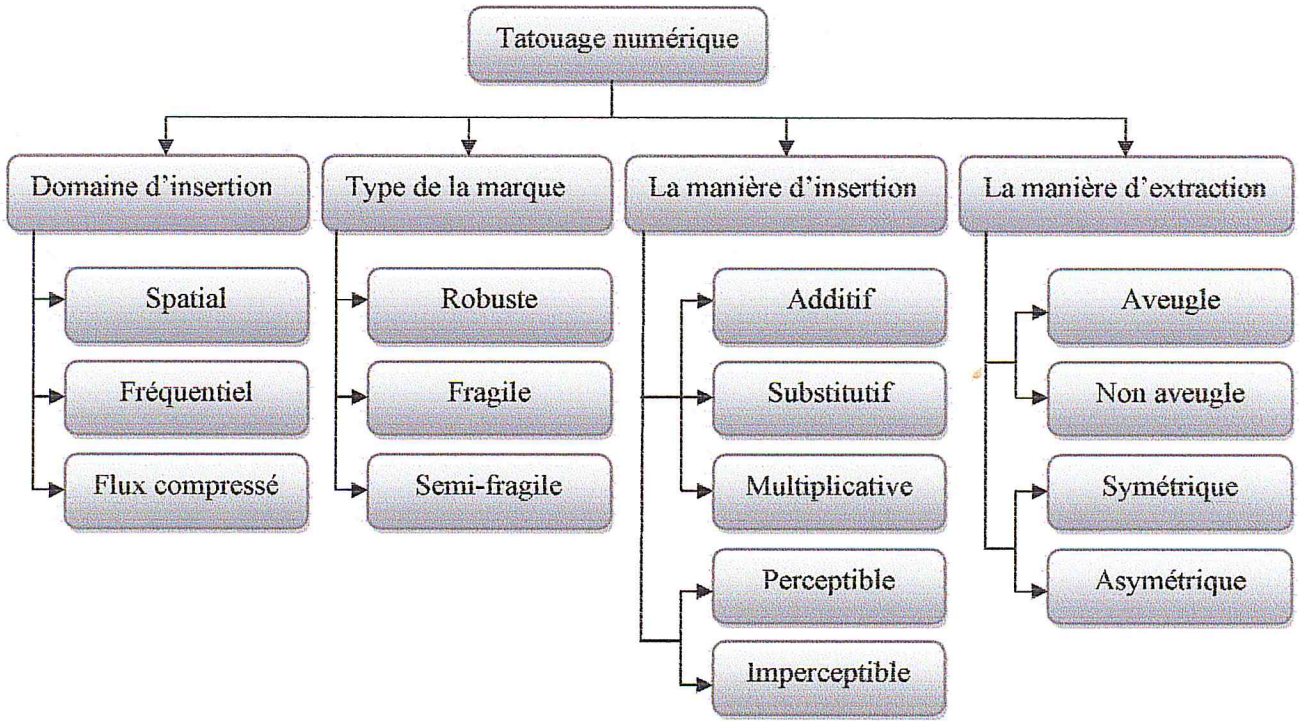

Figure I.3 Schéma de classification du tatouage numérique.

I.6.1 Classification selon le domaine d'insertion

I.6.1.1 Le domaine spatial

Les méthodes qui viennent en premier à l'esprit sont celles du domaine spatial, où elles modifient et agissent directement sur la luminance des pixels. Comme aucun traitement initial n'est requis, ces algorithmes sont très rapides et permettent de travailler en temps réel [11]. Leur problème réside dans leur fragilité face aux attaques. La plupart d'algorithmes utilisent le LSB bit (Least Significant Bit) pour l'insertion [12]. Il y a beaucoup de variantes de cette technique. Elle implique essentiellement d'insérer la marque en remplaçant les bits les moins significatifs des données du document avec un bit de la marque [9].

I.6.1.2 Le domaine fréquentiel

Le domaine transformé ou domaine fréquentiel est obtenu du domaine spatial en appliquant une transformée en une ou deux dimensions. La transformée peut se réaliser sur toute la donnée ou sur des blocs obtenus par une subdivision de celle ci [13]. Les transformées les plus utilisées dans le domaine de tatouage numérique sont : Transformée en Cosinus Discrète (TCD), Transformé de Fourier discrète (TFD) et Transformée en Ondelette Discrète (TOD). L'avantage principal de ce domaine par rapport au domaine spatial est que l'insertion de la marque se fait dans les coefficients de la transformée, et ainsi, elle assure que les modifications introduites sur un sous ensemble de ces coefficients seront propagées à tout les pixels dans le domaine spatial, ce qui rend ces modifications imperceptibles [14]. Le tableau I.1 illustre l'étude comparative entre les deux domaines d'insertion [11].

	Domaine spatial	**Domaine fréquentiel**
Coût de calcul	*Bas*	*Haut*
Qualité perceptuelle	*Haut*	*Bas*
Robustesse	*Fragile*	*Plus robuste*
Capacité	*Haut (dépend de la taille de l'image)*	*Bas*

Tableau 1. 1. Comparaison entre le tatouage dans le domaine spatial et fréquentiel.

I.6.1.3 Le domaine compressé

Le domaine compressé (bitstream) est obtenu du domaine fréquentiel, les algorithmes de tatouage sont fragiles et appliquées directement sur le flux binaire compressé. La plupart d'algorithmes utilisent le code de longueur variable VLC (Variable Length Code) pour l'insertion. Les avantages principaux de ce domaine est que les méthodes est sans perde et les marques insérés sont invisibles. Au plus, il ya aucune augmentation dans la taille de fichier [13].

I.6.2 Classification selon le type de la marque

I.7.1 Tatouage robuste

Il a pour objectif de transmettre une information malgré la modification du document. Lors de la lecture de la marque, certains algorithmes permettent d'extraire un message complet (une suite de symbole), tandis que d'autres indiquent simplement si le document a été marqué ou pas (on parle de détection de marque). Le tatouage robuste est particulièrement adapté au suivi et à la gestion de droits. Même si un fraudeur modifie le document, il est possible de retrouver l'auteur initial en insérant un numéro d'identification par tatouage robuste [15].

I.7.2 Tatouage fragile

Il permet de vérifier l'intégrité du document marqué. Il est fragile aux modifications, et permet de vérifier que le document n'a pas été manipulé et donc de l'authentifier. Néanmoins, certains systèmes de tatouage fragile sont tout de même résistants aux traitements les plus usuels (compression avec perte notamment) afin de ne détecter que les modifications les plus préjudiciables vis-à-vis de l'interprétation du document. Ce type de schéma de tatouage est dit **semi-fragile** [13].

I.7.3 Classification selon la manière d'insertion

I.6.3.1 Schéma additif

Ce schéma se résume dans l'extraction des coefficients à modifier du document d'origine puis le tatouage de ce dernier, s'effectue par l'ajout de la marque à ces coefficients. L'insertion peut s'effectuer soit directement sur le

document, dans le domaine spatial, soit dans un domaine transformé. De ce fait, adapter la marque au document d'origine est une contrainte essentielle à respecter pour que le signal qu'elle représente ne soit ni trop faible (problème de robustesse) ni trop fort (dégradation du signal original) [12].

La règle additive est définie par l'équation suivante :

$$Y_i = X_i + \lambda * W_i \quad telque \quad i = 1,2,\dots,N. \tag{I.1}$$

Où :

$X_i = \{X_1, X_2, \dots, X_n\}$ est la séquence de la donnée d'origine, $W_i = \{W_1, W_2, \dots, W_n\}$ est la séquence de la marque, λ est la force de marquage et $Y_i = \{Y_1, Y_2, \dots, Y_n\}$ représente la séquence de la donnée marquée.

I.6.3.2 Schéma multiplicative

Dans ce schéma, la marque à insérée est multipliée à ces coefficients [16]. Avec la même notation utilisée précédemment, la règle multiplicative est définie par l'équation suivante :

$$Y_i = X_i(1 + \lambda * W_i) \quad telque \quad i = 1,2,\dots,N. \tag{I.2}$$

I.6.3.3 Schéma substitutif

Dans les modes substitutifs, l'information à insérer est substituée à certaines caractéristiques du document. Par exemple, **Bas** et **Chassery** proposent dans [9] une méthode basée sur l'insertion de similarités. L'idée de base consiste donc à insérer une signature en modifiant le contenu structurel du document. Ainsi, l'étape d'insertion consiste d'une part à détecter les points d'intérêt et d'autre part à insérer des similarités autour de ces points. A la suite, la détection de marque s'effectue par recherche de ces similarités.

I.6.3.4 Tatouage imperceptible

Dans ce type de tatouage, on n'observe pas l'existence de la marque. En conséquence, elle n'affecte pas la qualité du document et ce dernier garde sa qualité commerciale [13].

I.6.3.5 Tatouage perceptible

Par contre, dans ce type de tatouage, la marque est bien visible dans le document. Il est utilisé plus dans l'application non commerciale [14].

I.6.4 Classification selon la manière d'extraction

Aussi, les schémas de tatouages peuvent être classés suivant les éléments nécessaires pour l'extraction (lecture du message depuis le document) de la marque.

I.6.4.1 Schéma aveugle

Dans ce cas, le document d'origine n'est pas nécessaire pour l'extraction de la marque. C'est le cas le plus favorable pour des applications pratiques [17].

I.6.4.2 Schéma non aveugle

Ce type de schéma nécessite le media d'origine pour pouvoir extraire le message inséré. Ces types sont de moins en moins étudiés, les applications concrètes sont rares [17].

I.6.4.3 Schéma symétrique

Dans ce schéma, La marque insérée est issue du codage du message à transmettre. Il est dépendant d'une clef. Si cette même clef est nécessaire au décodage pour l'extraction du message [18].

I.6.4.4 Schéma asymétrique

Le tatouage asymétrique repose sur l'utilisation de deux clés : une clé k_I privée pour l'insertion et une clé k_D publique pour la détection. K_D est issue de k_I par une transformation non inversible. N'importe quel utilisateur peut détecter la marque en connaissant K_D, mais seule la connaissance de k_I permet d'enlever ou modifier la marque [18].

I.7 Les attaques

Nous allons aborder la question des attaques que peut subir un tatouage numérique. La sensibilité d'un marquage vis à vis des attaques est très importante.

On peut distinguer deux grandes familles d'attaques, les bienveillantes et les malveillantes [17].

I.7.1 Attaques bienveillantes

Il s'agit de traitement qui n'a pas initialement pour objectif d'empêcher la détection de la marque. Il peut s'agir des dégradations dues à une compression, à un changement de résolution, à des filtrages passe-bas, etc....

I.7.2 Attaques malveillantes

Ce type d'attaque vise explicitement à rendre le tatouage inopérant. Ces attaques, comme souvent dans le domaine numérique, sont difficile à prouver d'un point de vue juridique. Toutefois, une attaque malveillante qui a réussi devra produire un contenu à la fois lavé de son tatouage et encore exploitable.

I.8 Application du tatouage numérique au système d'identification à base des images biométriques

Le problématique de ce travail concerne la sécurisation des images biométriques (le visage et d'empreinte digitale) qui sont stockés dans les bases de données des systèmes biométriques d'identification automatique AFIS (Automated Fingerprint Identification System) aux transmissent entre les différents modules de ces systèmes. En analysant ces images, on trouve que les seuls liens entre elles et les personnes correspondantes sont les codes de ces images. En d'autres termes, pour extraire une image de visage ou d'une empreinte digitale d'une personne quelconque de la base des données, on réfère toujours au code de cette image. Si ce lien (c-à-d. le code d'une image) est modifié soit par des fausses manipulations faites par des opérateurs ou par des attaques de virus ou programmes malicieux, alors cette image sera inutilisable car on ne saura jamais c'est l'image de qui. Et le pire est que, dans le cas d'une attaque de virus, une grande partie de code sera modifiée, causant ainsi un dysfonctionnement fatal du système. Récemment, les techniques du tatouage numérique ont été proposées pour renforcer la sécurité des systèmes d'identifications à base des images biométriques (AFIS). Dans la littérature, trois cas de figure sont envisagées sont reportés [19]. Ces cas sont :

- Augmenter la sécurité des images biométriques une fois transmises :
  - (i) du module d'acquisition au module d'extraction (sécurité 1).
  - (ii) du module d'acquisition au module du stockage (sécurité 2).
  - (iii) du module d'extraction au module du stockage (sécurité 3).

Pour réaliser cela, l'expéditeur (c.à.d. module d'acquisition, agences) insert une marque dans l'image biométrique et transmet l'image marquée au récepteur (c.à.d. module d'extraction ou base de données), qui essaye de détecter/extraire la marque insérée. Si celle ci est détectée/extraite, l'image biométrique est authentique et peut être traitée par le récepteur, autrement, elle est jugée truquée et rejetée par le récepteur (figure. I.4).

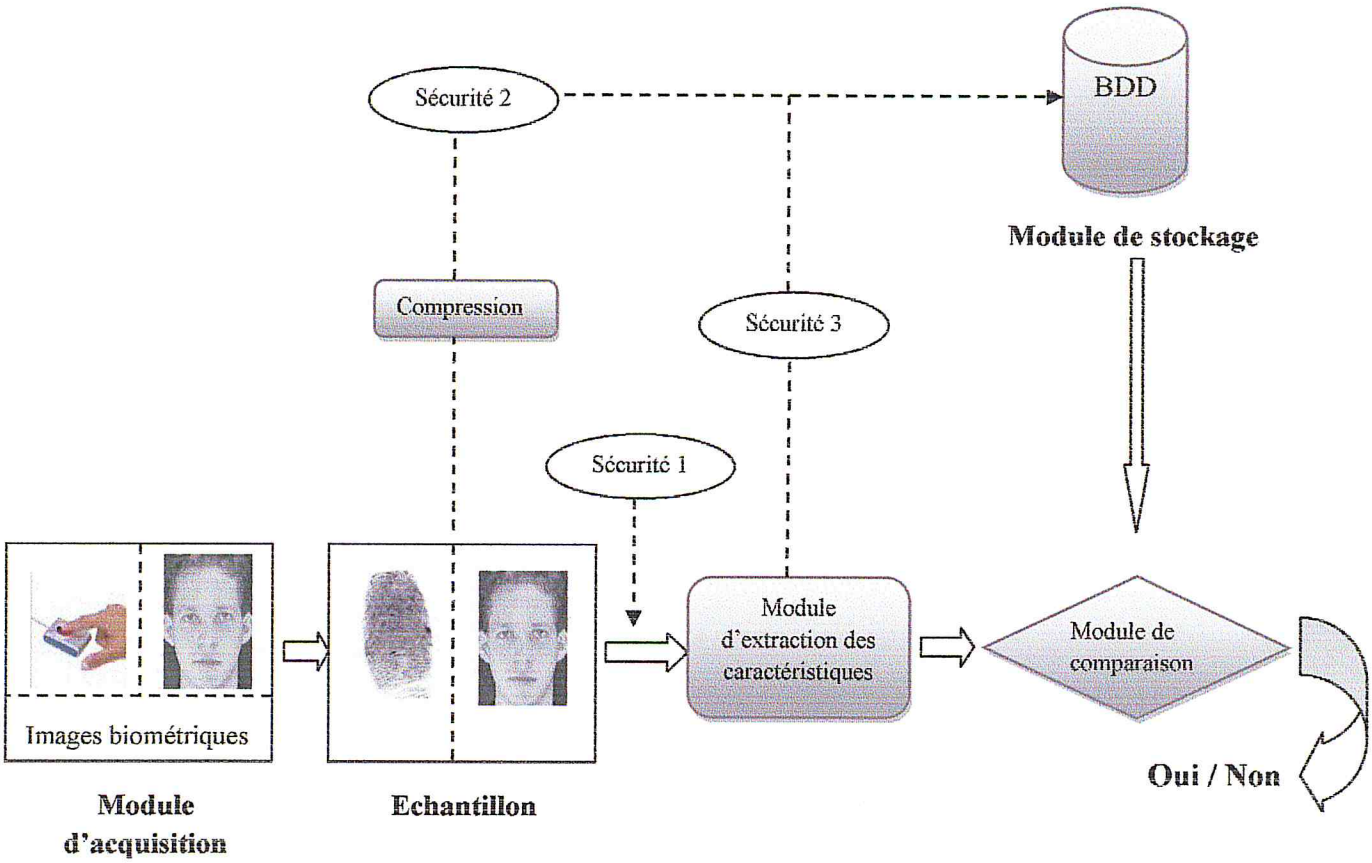

Figure I.4 Protection de la transmission des images biométriques par le tatouage numérique.

- Protéger l'originalité des images biométriques stockées dans les bases de données. Toutes ces images sont tatouées par des différentes marques (c.à.d. chaque image avec sa propre marque). Un attaquant ne peut pas enregistrer d'autres images dans la base de données, car ces images ne contiennent pas

les marques. Et donc la base de données est protégée contre toute essai d'ajouter frauduleusement des images (figure I.5).

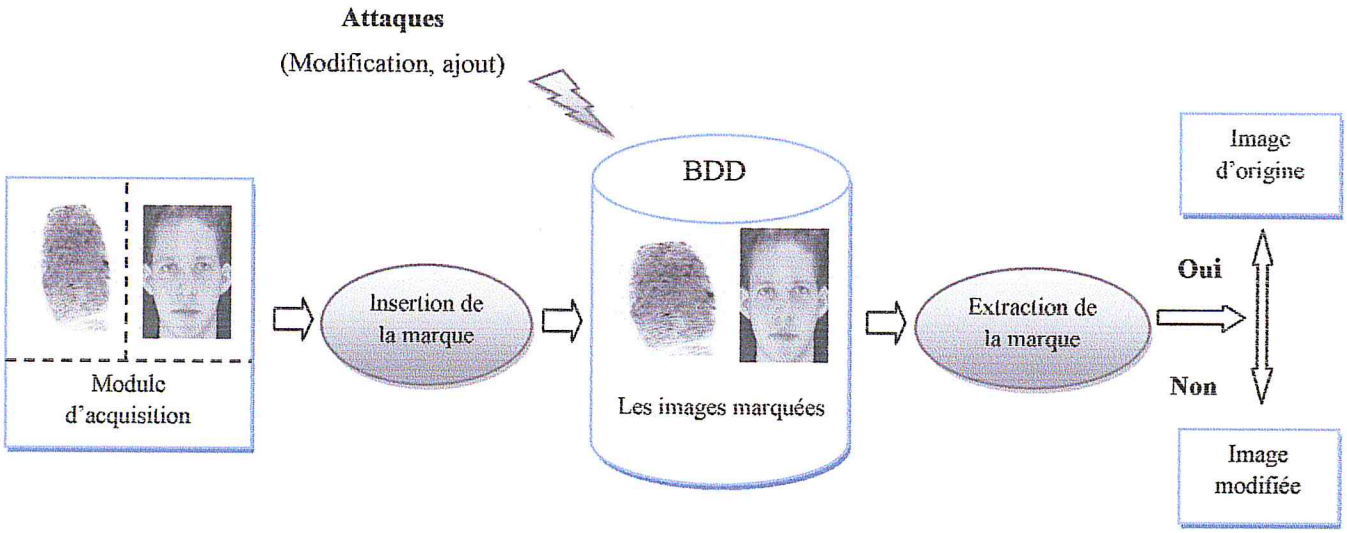

Figure I.5 Protection des images biométriques stockées par le tatouage numérique.

- Détecter des fraudes et des changements sur les images biométriques stockées dans la base de données. Ceci peut être accompli au moyen des méthodes du tatouage fragile dans lesquelles les marques ne résistent à aucune manipulation sur les images marquées. La détérioration ou la perte totale de la marque indiquera des fraudes ou manipulations possibles sur les images de visage stockées (figure I.6).

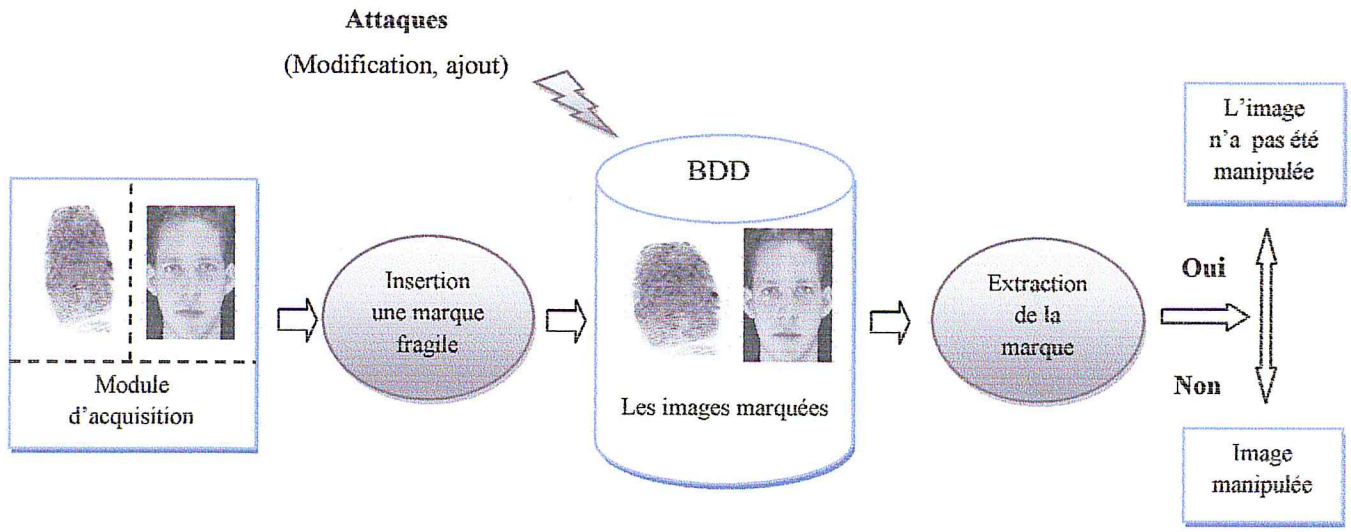

Figure I.6 Détection des fraudes sur les images biométrique par le tatouage numérique.

I.9 Conclusion

Dans ce chapitre, une introduction générale au tatouage numérique à été présentée, avec les définitions de base, les principaux concepts et les différentes applications envisagées en cette technologie.

Dans le prochain chapitre, nous allons présenter le codec WSQ appelé « Wavelet Scalar Quantization », dédié à la compression des images biométriques. Son principe de fonctionnement est basé sur la transformation en ondelette, la quantification scalaire adaptive, et le codage entropique qui feront l'objet du prochain chapitre.

CHAPITRE II :
WAVELET SCALAR QUANTIZATION ((WSQ))

II.1 Introduction

Les systèmes d'identification à base d'empreintes digitales, comme le système AFIS de la Gendarmerie Nationale ou celui qui sera utilisé dans le cadre des titres d'identité biométriques et électroniques, traitent, stockent et transmettent un nombre important d'images d'empreintes digitales. Ces images sont généralement des fichiers volumineux qui épuisent rapidement l'espace du stockage et augmentent le temps de leurs transferts. Ainsi, il est préférable de compresser ces images (réduire leurs tailles) plutôt que d'augmenter les capacités du stockage et de communication.

Des chercheurs du laboratoire National de Los Alamos, supportés par le Bureau Fédéral d'Investigation (FBI) et National Institute of Standards and Technology (NIST) [20], ont développé un algorithme de compression des images d'empreintes digitales, dit Wavelet Scalar Quantization (WSQ). Comme son nom l'indique, cet algorithme est basé sur la quantification scalaire des coefficients de la transformation en ondelette discrète. Il présente des performances meilleures par rapport à d'autres algorithmes de compression, comme le JPEG, surtout en termes de taux de compression et de préservation de la qualité visuelle des images. Et c'est pour ces raisons que cet algorithme est vite devenu un standard international d'échange des images d'empreintes digitales. Ce standard est spécifique aux images à niveaux de gris à 8 bits et avec une résolution de 500ppp et il est actuellement adopté par toutes les agences de police comme le FBI et l'Interpol, et utilisé dans les systèmes professionnels d'identification à base d'empreintes digitales, à savoir : AFIS de Cogent, BLUECHECK de Cogent, AFIS de Sagem.

II.2 Généralités

Le standard WSQ est basé sur une méthode de compression avec perte (i.e. l'image décompressée n'est pas identique à l'image d'origine). Cette méthode utilise un mécanisme dit « taux de compression cible» afin de contrôler la qualité des images compressées. De faibles taux de compression permettent d'avoir des images moins volumineux, mais aussi des qualités visuelles fortement dégradées. Le FBI spécifie une valeur de taux de compression cible de 0.75 bits/pixel, qui donne un rapport de compression typique de 15 :1 [20].

La classe de codeurs du standard WSQ utilise une *Transformation en Ondelette Discrète* (TOD), implémentée par un banc de filtres multi-cadences, pour décomposer l'image d'empreinte digitale en un ensemble de sous-bandes, où chacune représente une information dans une bande de fréquences particulière. Après, les coefficients de chaque sous-bande sont quantifiés en utilisant des valeurs d'une *table de quantification*. Finalement, les coefficients quantifiés sont concaténés en un nombre de blocs de données, qui sont transmis à une procédure de *codage de Huffman* peut être compressé. La procédure de codage utilise des codes de *la table de Huffman*. La procédure de compression est illustrée dans la figure II.1.

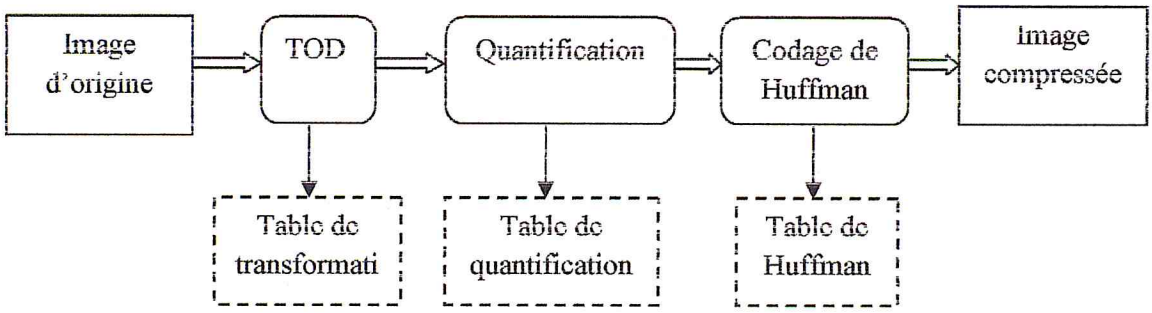

Figure II.1 Procédure de codage de l'algorithme de WSQ.

L'algorithme de WSQ est symétrique, donc les complexités des procédures de codage (Figure II.1) et de décodage (Figure II.2) sont similaires. Les tables spécifiées à la procédure de codage (i.e. table de quantification et table de Huffman) sont des tables dépendantes de l'image (c.à.d. les valeurs de chaque table changent avec l'image à compresser). Donc, les mêmes tables utilisées pour compresser une image quelconque doivent être fournies au décodeur pour décompresser cette image.

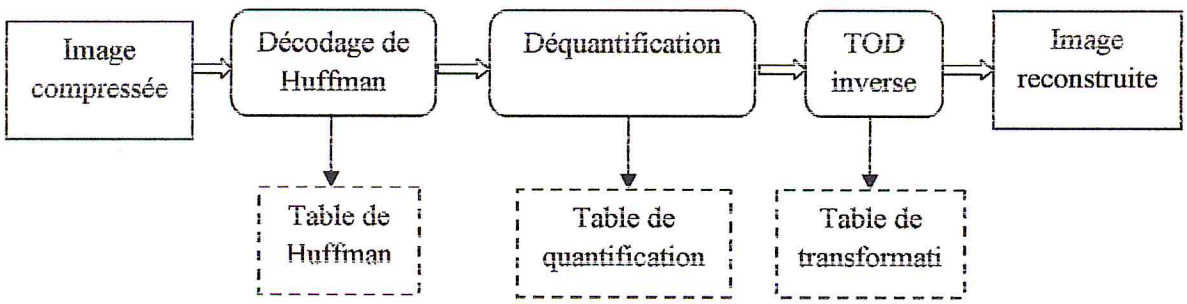

Figure II.2 Procédure de décodage de la compression WSQ.

II.3 Les étapes de la compression WSQ

La compression WSQ se déroule essentiellement en trois étapes (figure II.1), précédées par une étape de prétraitement, dite *la normalisation*.

II.3.1 La normalisation

L'image de l'empreinte digitale doit être capturée avec une précision de 8bits/pixel. Avant que le processus de codage n'applique la TOD, les valeurs des pixels de l'image, soit $I\ (m,\ n)$, doivent être normalisées conformément à l'équation suivante :

$$I'(m,n) = \frac{[I(m,n) - M]}{R} \qquad \begin{matrix} 0 \le m \le Y-1 \\ 0 \le n \le X-1 \end{matrix} \qquad \text{(II.1)}$$

où : Y et X sont respectivement l'hauteur et la largeur de l'image I. M est la moyenne des valeurs des pixels et R représente l'échelle. Ces deux paramètres sont calculés par le processus de codage et transmis au processus de décodage, qui doit appliquer la transformation inverse pour restaurer les valeurs des pixels de l'image à leur échelle originale.

Selon les spécifications du FBI [20], M et R sont calculés par les deux équations suivantes :

$$M = \frac{\sum_{m=0}^{Y-1}\sum_{n=0}^{X-1} I(m,n)}{XY} \qquad \text{(II.2)}$$

Et

$$R = \frac{1}{128} max(I_{max} - M, M - I_{min}) \qquad \text{(II.3)}$$

Où : I_{max} et I_{min} sont respectivement la valeur maximale et la valeur minimale des pixels de l'image I.

II.3.2 La transformation en ondelette discrète (TOD)

La transformation en ondelette discrète (TOD) consiste à diviser le signal en basses et hautes fréquences en utilisant des filtres passe-bas h et passe-haut g (figure II.3). Les sorties des filtres seront sous-échantillonnées par deux pour donner résultat à:

- L représentant les coefficients d'approximation via le filtre h.
- H représentant les coefficients de détails via le filtre g.

Dans le cas des images (i.e. espace à deux dimensions), l'application de la TOD génère quatre sous-bandes (figure II.4): (i) la première correspondant aux basses fréquences (approximation de l'image), (ii) la deuxième aux hautes fréquences colonnes (détails horizontaux), (iii) la troisième aux hautes fréquences lignes (détails verticaux) et (iv) la quatrième aux hautes fréquences lignes et colonnes (détails diagonaux). Elle passe par les opérations suivantes:

- Filtrer l'image d'entrée selon les lignes par le filtre *h*, suivi par le même filtre suivant les colonnes, ceci génère l'approximation (l'espace 00) ;
- Filtrer l'image d'entrée selon les lignes par le filtre *h*, suivi par le filtre *g* suivant les colonnes, ceci génère les détails horizontaux (l'espace 10) ;
- Filtrer l'image d'entrée selon les lignes par le filtre *g*, suivi par le filtre *h* suivant les colonnes, ceci génère les détails verticaux (l'espace 01) ;
- Filtrer l'image d'entrée selon les lignes par le filtre *g*, suivi par le même filtre suivant les colonnes, ceci génère les détails diagonaux (l'espace 11).

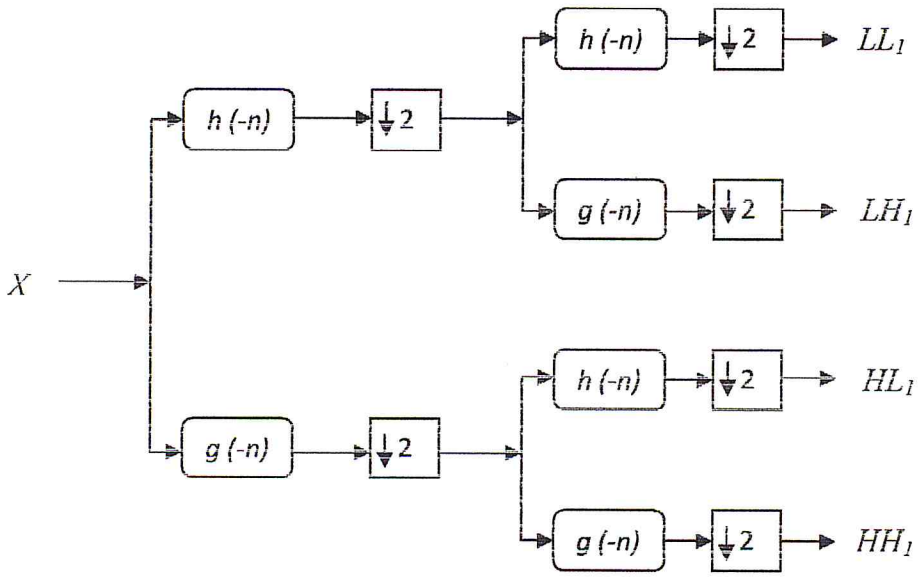

Figure II.3 Décomposition en ondelette 2^{éme} niveau (2-D).

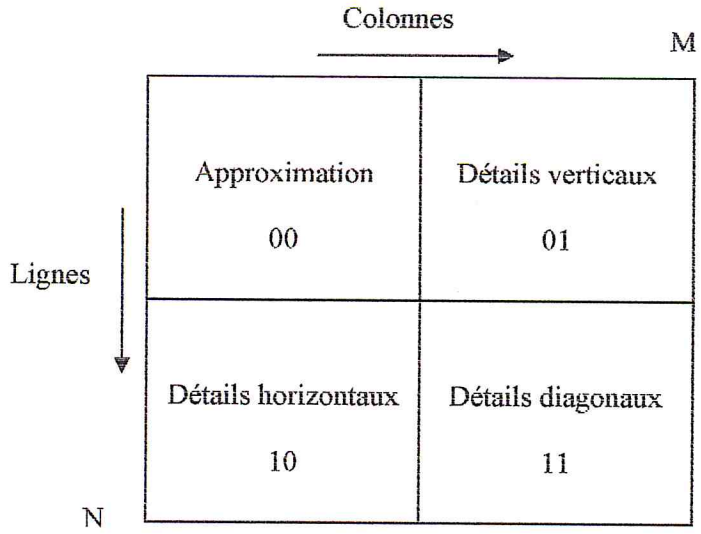

Figure II.4 Les sous-bandes issues de la TOD à un seul niveau.

Pour appliquer la TOD à un niveau supérieur, les mêmes opérations décrites ci-dessus sont appliquées à l'approximation c.à.d. à l'espace 00. Les types de coefficients obtenus sont représentés dans la figure II.5.

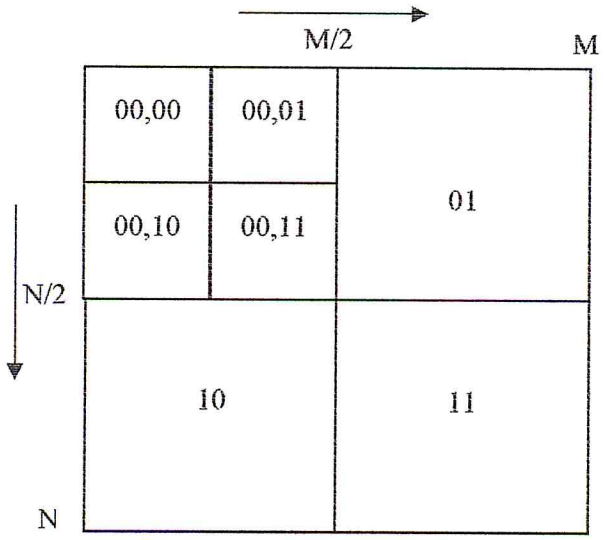

Figure II.5 Type des coefficients de la TOD à deux niveaux.

La table suivante représente les coefficients des filtres (9,7) respectivement passe-bas (h_0) et passe-haut (h_1) utilisés dans l'étape de la transformation en ondelette discrète.

Le filtre	La valeur exacte	La valeur approximative
$h_0(0)$	$-5\sqrt{2}x_1(48\|x_2\|^2 - 16\Re x_2 + 3)/32$	0.85269867900940
$h_0(\pm1)$	$-5\sqrt{2}x_1(8\|x_2\|^2 - \Re x_2)/8$	0.37740285561265
$h_0(\pm2)$	$-5\sqrt{2}x_1(4\|x_2\|^2 - 4\Re x_2 - 1)/16$	−0.11062440441842
$h_0(\pm3)$	$-5\sqrt{2}x_1(\Re x_2)/8$	−0.02384946501938
$h_0(4\pm)$	$-5\sqrt{2}x_1/64$	0.037828455506995
$h_1(-1)$	$\sqrt{2}(6x_1 - 1)/16x_1$	0.78848561640566
$h_1(-2,0)$	$-\sqrt{2}(16x_1 - 1)/64x_1$	−0.41809227322221
$h_1(-3,1)$	$\sqrt{2}(2x_1 + 1)/32x_1$	−0.040689417609558
$h_1(-4,2)$	$-\sqrt{2}/64x_1$	0.064538882628938

Tableau II.1 les filtres utilisés lors de la transformation.

Où :

$$x_1 = A + B - 1/6 \tag{II.4}$$

$$x_2 = -(A + B)/2 - 1/6 + i\sqrt{3}\,(A - B)/2 \tag{II.5}$$

$$A = \left((-14\sqrt{15} + 63)/1080\sqrt{15}\right)^{1/3} \tag{II.6}$$

$$B = \left((-14\sqrt{15} - 63)/1080\sqrt{15}\right)^{1/3} \tag{II.7}$$

Le standard WSQ utilise un modèle de décomposition en 64 sous-bandes (figure II.6) [20]. Les numéros et les types des sous-bandes sont donnés dans tableau II.2.

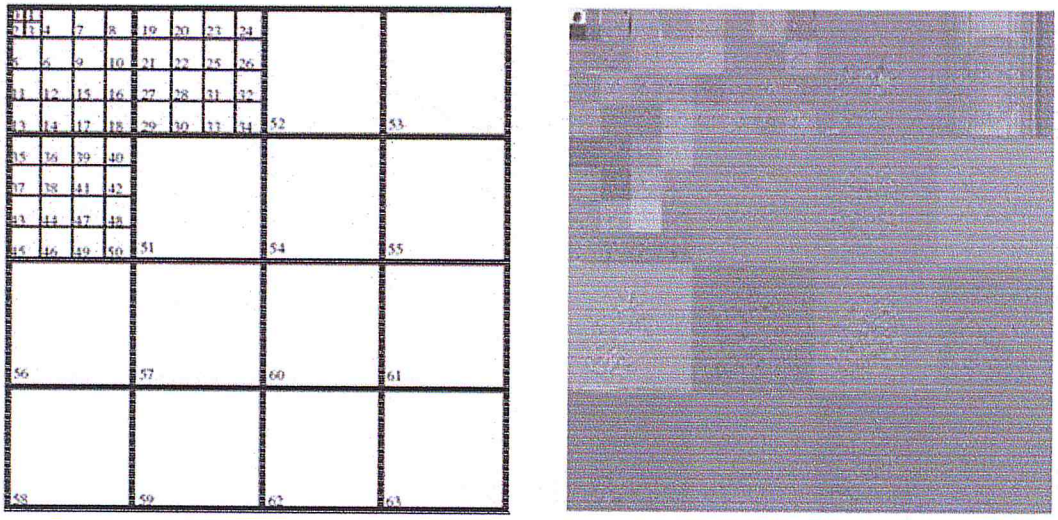

Figure II.6 Décomposition d'une image d'empreinte digitale en 64 sous-bandes.

0	00,00,00,00,00	22	00,10,10,11	44	00,01,00,11
1	00,00,00,00,10	23	00,10,00,10	45	00,01,00,00
2	00,00,00,00,01	24	00,10,00,00	46	00,01,00,10
3	00,00,00,00,11	25	00,10,00,11	47	00,01,10,11
4	00,00,00,10	26	00,10,00,01	48	00,01,10,01
5	00,00,00,01	27	00,10,11,01	49	00,01,10,10
6	00,00,00,11	28	00,10,11,11	50	00,01,10,00
7	00,00,10,10	29	00,10,11,00	51	00,11
8	00,00,10,00	30	00,10,11,10	52	10,10
9	00,00,10,11	31	00,10,01,11	53	10,00
10	00,00,10,01	32	00,10,01,01	54	10,11
11	00,00,01,01	33	00,10,01,00	55	10,01
12	00,00,01,11	34	00,10,01,00	56	01,01
13	00,00,01,00	35	00,01,01,00	57	01,11
14	00,00,01,10	36	00,01,01,10	58	01,00
15	00,00,11,11	37	00,01,01,01	59	01,10
16	00,00,11,01	38	00,01,01,11	60	11,11
17	00,00,11,10	39	00,01,11,10	61	11,01
18	00,00,11,00	40	00,01,11,00	62	11,10
19	00,10,10,00	41	00,01,11,11	63	11,00
20	00,10,10,10	42	00,01,11,01		
21	00,10,10,01	43	00,01,00,01		

Tableau II.2 Numéros et types des sous-bandes du standard WSQ.

II.3.3 La quantification:

Après que l'image est décomposée en sous-bandes, les coefficients obtenus sont uniformément quantifiés. Le type de la quantification utilisée est la quantification uniforme avec zone morte, qui est un type spécial de quantification, où l'intervalle autour de zéro est plus large [20]. La *zone morte* ou *dead-zone* qualifie donc cet intervalle autour de zéro, qui permet à l'ensemble des valeurs de source considérées comme *petites*, d'être quantifiées à une seule même valeur (généralement zéro). Les quantificateurs avec zone morte occupent une place centrale dans la compression par ondelettes car ils permettent d'éliminer les coefficients non significatifs de faible amplitude au profit des coefficients

significatifs qui sont alors quantifiés plus finement. L'intérêt de la zone morte réside dans le fait qu'elle permet de quantifier plus finement les coefficients significatifs de la source.

La largeur de la zone morte (Z) et le pas de quantification (Q) pour chaque sous-bande sont transmis dans la table de quantification. Un pas de quantification nul pour une sous-bande k (i.e. $Q_k = 0$) indique que tous les coefficients de cette sous-bande sont nuls et par conséquent cette sous-bande ne sera pas transmise. L'équation (II.8) suivante est appliquée au coefficient $a_k(m,n)$ de la sous-bande k pour obtenir le coefficient quantifié $p_k(m,n)$.

$$p_k(m,n) = \begin{cases} \left\lfloor \dfrac{a_k(m,n) - Z_k/2}{Q_k} \right\rfloor + 1, & a_k(m,n) > Z_k/2 \\ 0, & -Z_k \leq a_k(m,n) \leq Z_k/2 \\ \left\lceil \dfrac{a_k(m,n) + Z_k/2}{Q_k} \right\rceil - 1, & a_k(m,n) < -Z_k/2 \end{cases} \quad \text{(II.8)}$$

Le processus de décodage déquantifie les coefficients par l'équation suivante :

$$\hat{a}_k(m,n) = \begin{cases} (p_k(m,n) - C)Q_k + Z_k/2, & p_k(m,n) > 0 \\ 0, & p_k(m,n) = 0 \\ (p_k(m,n) + C)Q_k - Z_k/2, & p_k(m,n) < 0 \end{cases} \quad \text{(II.9)}$$

Où : C est les centres des intervalles de quantification. Selon le standard WSQ, la valeur de ce paramètre doit être : 0.44 [19].

II.3.3.1 La valeur estimée de la variance de chaque sous-bande

Le calcule de la largeur de la zone morte montre nécessite la valeur estimée σ_k^2 de la variance de chaque sous bande. La variance d'une sous-bande est estimée à partir d'une sous région de cette sous-bande. Soit a_k une sous-région de la sous-bande k, où sa largeur X_k' et sa hauteur Y_k' sont déterminer comme suit :

$$X_k' = \left\lfloor \frac{3X_k}{4} \right\rfloor \qquad Y_k' = \left\lfloor \frac{7Y_k}{16} \right\rfloor \quad \text{(II.10)}$$

Où : X_k et Y_k sont respectivement la largeur et l'hauteur de la sous-bande k.

La variance σ_k^2 de la sous-bande k est estimée à partir de la sous-région a_k en utilisant l'équation (II.11) suivante:

$$\sigma_k^2 = \frac{1}{X_k' Y_k' - 1} \sum_{n=x_{0,k}}^{x_{1,k}} \sum_{m=y_{0,k}}^{y_{1,k}} (a_k(m,n) - u_k)^2 \qquad \text{(II.11)}$$

où : u_k représente la moyenne de la sous-région a_k. Et :

$$x_{0,k} = \left\lfloor \frac{X_k}{8} \right\rfloor, \qquad x_{1,k} = x_{0,k} + X_k' - 1 \qquad \text{(II.12)}$$

$$y_{0,k} = \left\lfloor \frac{9Y_k}{32} \right\rfloor, \qquad y_{1,k} = y_{0,k} + Y_k' - 1 \qquad \text{(II.13)}$$

II.3.3.2 Calcul le pas de quantification

Le pas de quantification Q_k d'une sous-bande k est calculé en utilisant la formule suivante :

$$Q_k = \begin{cases} 1/q & k = 0-3 \\ 10(A_k \log(\sigma_k^2))/q & k = 4-59 \; et \;\; \sigma_k^2 \geq 1.01 \\ 0 & k = 60-63 \; et \;\; \sigma_k^2 \leq 1.01 \end{cases} \qquad \text{(II.14)}$$

où : σ_k^2 est la variance de la sous-bande k estimée comme expliqué précédemment.

A_k est un constant est sa valeur est donnée comme suit :

$$A_k = \begin{cases} 1.32 & si \;\; k = 52,56 \\ 1.08 & si \;\; k = 53,58 \\ 1.42 & si \;\; k = 54,57 \\ 1.08 & si \;\; k = 55,59 \\ 1.00 & sinon \end{cases} \qquad \text{(II.15)}$$

q est le facteur de proportionnalité, il permet de contrôler les largeurs des pas de quantification et le taux global de compression.

La formule de calcul du paramètre q est donnée comme suit :

$$q = \gamma^{-1} \, 2^{\frac{r}{S}-1} \left[\prod_{k=K} \left(\frac{\sigma_k}{Q_k'} \right)^{1/m_k} \right]^{-1/S} \qquad \text{(II.16)}$$

Où :

- γ est une constante et sa valeur est fixée par le standard à 2.5 [21].

- r représente le taux de compression cible, il est à déterminer par l'application. Par exemple, le standard du FBI utilise une valeur de 0.75 bits/pixels pour r [21].
- m_k est le facteur de sous-échantillonnage, qui est défini pour être le rapport de la taille de l'image à la taille de la sous-bande, par exemples : m_0=1024, m_4=256, m_{52}=16.
- σ_k représente l'écart type de la sous-bande k.
- S dénote la fraction des coefficients de la TOD, elle est calculée par l'équation suivante :

$$S = \sum_{k=K} \frac{1}{m_k} \tag{II.17}$$

$Q_k^{'}$ est un facteur, calculé par la formule suivante :

$$Q_k^{'} = \begin{cases} 1 & k = 0-3 \\ 10(A_k \log(\sigma_k^2)) & k = 4-59 \; et \;\; \sigma_k^2 \geq 1.01 \\ 0 & k = 60-63 \; et \;\; \sigma_k^2 \leq 1.01 \end{cases} \tag{II.18}$$

II.3.3.3 Calcul la largeur de la zone morte

La largeur de la zone morte Z_k d'une sous-bande k est calculée en fonction du pas de quantification Q_k par la formule suivante :

$$Z_k = 1.2\, Q_k \tag{II.19}$$

Le standard WSQ permet à le codeur d'écarter quelques sous-bandes et de transmettre des largeur de pas de quantification nulles ($Q_k = 0$) pour signifier qu'aucun coefficient n'est transmis pour la sous-bande k. par exemple, ceci est toujours fait pour les sous-bandes 60, 61, 62 et 63 et peut être aussi bien fait pour d'autres sous-bandes si le codeur détermine que ces sous-bandes contiennent peu d'information qu'elles ne devraient pas être transmises (i.e. $\sigma^2 < 1.01$). Une fois que les pas de quantification ont été déterminés, la procédure de quantification sera appliquée à chaque sous-bande.

II.3.4 Le codage de Huffman

Le codage de Huffman est un algorithme de codage qui fut mis au point en 1952 par David Albert Huffman. Il appartient au type de statistique qui grâce à une méthode d'arbre permet de coder les octets revenant le plus fréquemment avec une séquence de bits beaucoup plus courte que d'ordinaire. Cet algorithme offre des taux de compression démontrés les meilleurs possibles pour un codage par symbole [20].

Concernant le standard WSQ, les sous-bandes sont regroupées en bloc avant qu'elles soient codées. Pour assurer la transmission progressive de l'image, elle est divisée en trois blocs avec la première coupure entre les sous-bandes 18 et 19, et la deuxième entre 51 et 52 sous-bandes. Toutes les sous-bandes dans un bloc doivent utiliser la même table de Huffman. Dans une sous-bande, les coefficients sont classés de gauche à droite, de haut en bas. Dans un bloc, les sous-bandes doivent être inscrites consécutivement dans l'ordre croissant.

Le codage génère deux codeurs, un codeur de Huffman dit BITS1 construit pour le bloc 1 (les sous-bandes de 0 à 18) et un deuxième codeur de Huffman dit BITS2 construit pour le bloc 2 (les sous-bandes de 19 à 51) et bloc 3 (les sous-bandes de 52 à 59) [20].

Avant qu'un codeur de Huffman ne commence à coder un bloc, les coefficients quantifiés de ce bloc sont remplacés par des codes ou mots spéciaux en utilisant un dictionnaire spécifique (tableau II.3). Dans ce cas, on a trois cas de figure :

- Pour les coefficients, entre -73 et 74 : ces coefficients sont remplacés directement par de mots du (tableau II.3). Par exemple, si le coefficient est -73 alors il sera remplacé par le mot 107, et s'il est 0 il sera remplacé par le mot 1 et pas par le mot 180 (qui n'est pas utilisable).
- Pour les coefficients hors l'intervalle [-73, 74] : on doit ajouter un mot (code) avant le coefficient selon sa valeur. Par exemple, pour coefficient 90, on ajoute le code 101 avant lui (c.à.d. 101 90) ; -90 devient 102 90.
- Pour les séquences de coefficients nuls : pour les séquences de coefficients nuls comportant moins de 100 coefficients, toute la séquence sera remplacée par le nombre de coefficients de la séquence. Par exemple, la séquence 0 0 0 0 0 devient un seule mot qui est 5. Pour les séquences ayant plus de 100

coefficients, on ajoute d'abord un mot (soit 105 pour une séquence de moins de 256 coefficients ou 105 pour des séquences ayant plus de 256 coefficients nuls) puis le nombre de coefficients nuls.

Position	Valeur
1	Séquence de bits 0 de longueur 1 (un seul bit 0)
2	Séquence de bits 0 de longueur 2 (deux bits 0)
3	Séquence de bits 0 de longueur 3 (trois bits 0)
100	Séquence de bits 0 de longueur 100 (cents bits 0)
101	Échappe pour un coefficient positif de longueur de 8 bits
102	Échappe pour un coefficient négatif de longueur de 8 bits
103	Échappe pour un coefficient positif de longueur de 16 bits
104	Échappe pour un coefficient négatif de longueur de 16 bits
105	Échappe pour une séquence de bits 0 de longueur de 8 bits
106	Échappe pour une séquence de bits 0 de longueur de 16 bits
107	Coefficient de valeur -73
108	Coefficient de valeur -72
109	Coefficient de valeur -71
↓	
180	N'est pas utilise. Utilisez position 1 seulement.
↓	
253	Coefficient de valeur 73
254	Coefficient de valeur 74

Tableau II.3 Dictionnaire du codage de Huffman.

II.4 L'organisation du fichier (.wsq)

L'image compressée est représentée par une structure uniforme et un ensemble de paramètres. Les différentes parties de la donnée de l'image compressée sont identifiées par des codes spéciaux de deux octets appelés *marqueurs* (markers). Certains marqueurs sont associés à des séquences particulières de paramètres comme les tables de spécifications et les entêtes. Autres marqueurs sont utilisés sans paramètres comme pour marquer le début et la fin de l'image. Quand un

marqueur est associé avec une séquence particulière de paramètres, le marqueur et ses paramètres composent un *segment marqueur* (marker segment).

- **Segments marqueur (Marker segments) :** Chaque segment marqueur commence par une valeur d'identification de 2 octets, appelée *marqueur*, suivi par une séquence spécifique de *paramètres*. Le marqueur sert à identifier les types de données stockées dans le segment.
- **Marqueurs (Markers) :** Les marqueurs sont des codes bien spécifiques, utilisés pour identifier les différentes parties constituant un fichier WSQ. Tous les marqueurs sont représentés par 2 octets : (i) le premier octet est toujours 'FFh' et indique le début possible d'un segment marqueur ; (ii) le deuxième octet peut avoir n'import quelle valeur, sauf '00h' et 'FFh', et identifie le type de données ou paramètres du segment marqueur (Tableau II.4). Cette structure spéciale des codes affectés aux marqueurs permet au décodeur de parcourir le fichier WSQ et localiser ses différentes parties sans décoder toute la donnée compressée.

Marqueur	Désignation
FFA0h	SOI : début de l'image (Start Of Image)
FFA1h	EOI : fin de l'image (End Of Image)
FFA2h	SOF: début du frame (Start of Frame)
FFA3h	SOB: début du block (Start of Block)
FFA4h	DTT : définit la table de transformation (Define Transform Table)
FFA5h	DQT: définit la table de Huffman (Define Huffman Table)
FFA6h	DHT : définit la (les) table(s) de Huffman (Define Huffman table(s))
X'FFA8'	COM: commentaire (Comment)

Tableau II.4 Désignation des marqueurs.

Note : - les marqueurs SOI et EOI ne sont pas suivis par aucun paramètre ou donnée (c.à.d. segment sans paramètre).

- les segments marqueurs identifiés par SOF et SOB sont appelés entêtes (headers): entête de frame et entête du bloque, respectivement.

Les octets de valeur 'FFh' sont aussi utilisés comme octets de remplissage, qui peuvent précéder n'importe quelle segment marqueur. Chaque octet de valeur 'FFh', qui un octet de la donnée compressée, est toujours suivi par un octet de valeur '00h' pour préciser que l'octet 'FFh' n'est pas le début d'un marqueur ou octet de remplissage (Tableau II.5).

Valeur	Signification
FFA1h	FFh est le premier octet du marqueur EOI.
FFFFFFFFA0h	Trois octets de remplissage de valeur 'FFh' précédent un marqueur SOI.
FF00FF00FF00h	Trois octets de données interprétés comme FFFFFFh.

Tableau II.5 Exemples des cas d'utilisation d'un octet de valeur 'FFh'.

- **Paramètres:** Les paramètres sont des entiers avec des valeurs spécifiques à : (i) la procédure d'encodage, (ii) les caractéristiques de l'image à compresser, et (iii) autres caractéristiques sélectionnées par l'application. Ils encodent des informations critiques, sans lesquels la procédure de décodage ne peut pas reconstruire l'image, à l'exception de quelques paramètres qui sont optionnels.

  Les paramètres sont définis par des codes (entiers non-signés) de longueurs variables, soient : 4 bits, 1-octet ou 2-octets. Les paramètres de longueur 4-bits sont toujours des pairs, et chaque pair est codé par un seul octet. Le premier paramètre de 4-bits du pair doit occuper les 4 bits les plus signifiants du pair. Pour les paramètres de longueurs 2-octets (16 bits), l'octet le plus signifiant doit être en premier

- **Segments des données codées :** Un segment des données codées est la sortie d'une procédure du codage de Huffman. Il contient les sous-bandes quantifiées, qui sont déjà codées par le codage de Huffman en une seule séquence de données. Cette dernière comporte un nombre entier d'octets et pour assurer ça :
  - si nécessaire, des bits 1 sont ajoutes à la fin de la donnée compressée afin de compléter le dernier octet du segment.

- un marqueur ne doit pas figurer dans un segment de données codées, et pour cela tout octet de valeur 'FFh' est systématiquement suivi par un octet de valeur '00h' (i.e. 'FF00h' remplace l'octet 'FFh').

Il est à noter que les sous-bandes contenues dans un segment n'ont pas de séparateurs, c'est au décodeur de déterminer la taille de chaque sous-bande.

II.4.1 La structure Générale

Les différentes parties qui d'un fichier WSQ sont représentées dans Figure II.7. Le premier niveau spécifie qu'un fichier WSQ doit: (i) commencer par le marqueur SOI, (ii) suivi par un frame d'une image, (iii) terminer par le segment marqueur EOI. Le deuxième niveau détermine qu'un frame doit : (i) commencer par une entête du frame, (ii) suivi par un certain nombre de blocs de données, plus précisément de trois à huit blocs. Le troisième niveau spécifie qu'un bloc doit : (i) commencer par une entête du bloc, (ii) suivi par un ou plusieurs segments de données codées par le codage de Huffman.

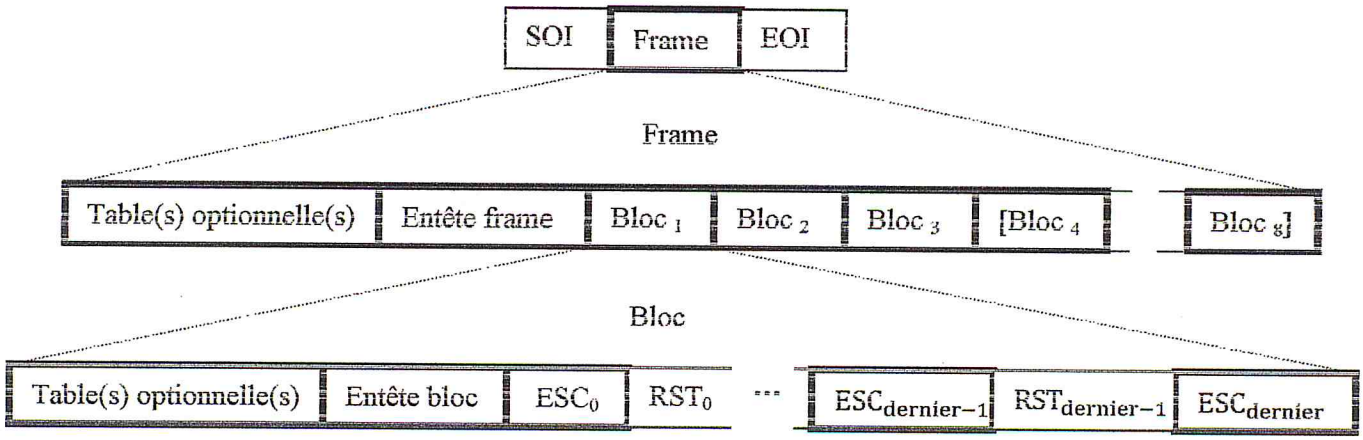

Figure II.7 La structure générale d'un fichier WSQ.

Juste avant l'entête du frame ou chaque entête d'un bloc, des segments marqueurs optionnels peuvent être ajoutés si nécessaire. Ces segments marqueurs sont : table des spécifications de la transformation, table des spécifications de la quantification, table des spécifications du Codage de Huffman ou commentaire (Figure II.8).

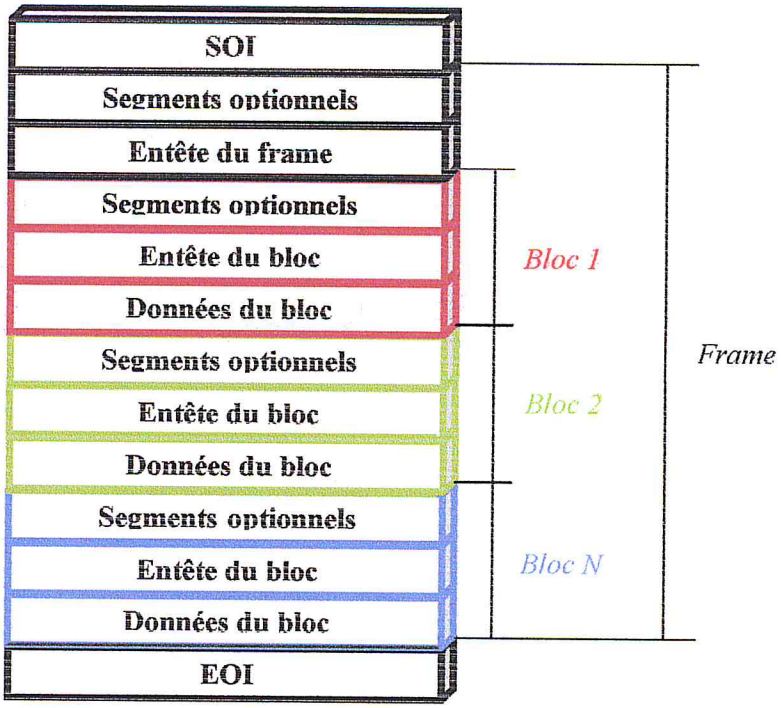

Figure II.8 Format général d'un fichier WSQ.

Deux variations du format de ficher WSQ ont aussi été définies. La première variation est utilisée pour enregistrer les données de l'image compressée seulement. Si un fichier contient les données de l'image sans les tables nécessaires, alors c'est le *'format abrégé de l'image compressé'*. La deuxième variation enregistre seulement les tables des spécifications utilisées pour interpréter l'image compressée. Si un fichier comporte quelques ou toutes les tables nécessaires (transformation, quantification, codage de Huffman) mais pas les données de l'image compressée, alors c'est le *'format abrégé pour tables des spécifications'*.

II.4.2 Syntaxe du l'entête frame

Figure II.9 spécifie l'entête de frame qui doit être présente au début d'un frame. Cet entête spécifie les caractéristiques de l'image source, ainsi que la version de codeur utilisé.

Entête du frame

SOF	Lf	A	B	Y	X	Em	M	Er	R	Ev	Sf

Figure II.9 Syntaxe de l'entête du frame.

Le marqueur et les paramètres, représentés dans Figure II.10, sont définis comme suit :

- **SOF**: (16 bits) Start of Frame; marque le début de l'entête du frame et sa valeur est toujours 'FFA2h'.
- **Lf**: (16 bits) la longueur (en octets) de l'entête du frame. La taille de SOF n'est pas incluse, donc typiquement la valeur de Lf est égale à 17.
- **A**: (8 bits) la valeur du calibrage de la couleur noire.
- **B**: (8 bits) la valeur du calibrage de la couleur blanche; A et B représente l'échelle de calibrage utilisé pour afficher l'image.
- **Y**: (16 bits) le nombre de lignes de l'image source.
- **X**: (16 bits) le nombre d'éléments (8bits/pixel) par ligne dans l'image source.
- **Em**: (8 bits) l'exposant de M; La virgule de M est décalée à gauche Em positions (ex : Si M = 12,345, Em = 3).
- **M**: (16 bits) paramètre de normalisation. Il représente la moyenne de l'image.
- **Er**: (8 bits) l'exposant de R; la virgule de R est décalée à gauche Er positions (ex : Si R = 1,2345, Er = 4);
- **R**: (16 bits) paramètre de la normalisation. Il représente l'échelle de l'image.
- **Ev**: (8 bits) identifie la version de l'algorithme de codeur WSQ (paramétrage) utilisé pour compresser l'image.
- **Sf**: (16 bits) identifie le logiciel (software) utilisé pour compresser l'image.

II.4.3 Syntaxe du l'entête d'un bloc

La figure II.10 spécifie l'entête du bloc qui doit être présente au début de chaque bloc. Cet entête spécifie la table de Huffman qui a été utilisée pour coder toutes les sous-bandes dans le bloc.

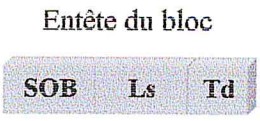

Figure II.10 Syntaxe de l'entête d'un bloc.

Le marqueur et les paramètres présentés dans (Figure II.10) sont définis comme suit:

- **SOB**: (16 bits) Start of Bloc ; marque le début de l'entête du bloc et sa valeur est toujours égale à 'FFA3h'.
- **Ls**: (16 bits) la longueur (en octet) du l'entête du bloc. La taille de SOB n'est pas incluse, donc typiquement la valeur de Ls est égale à 3.
- **Td**: (8 bits) spécifie la table de Huffman utilisé, parmi les huit tables possibles, pour encoder le bloc.

II.4.4 Table de définition de la transformation

La table de définition de la transformation contient un entête qui comporte le marqueur DTT, le paramètre de longueur de la table Lt, et les longueurs de deux filtres (h et g). Chaque vecteur contient un ou plusieurs éléments de longueurs de 6 octets. Ces éléments sont les coefficients des filtres de l'ondelette et sont codés sur 6 octets de précision en utilisant trois paramètres séparés : paramètre de signe (1 octet), paramètre de l'exposant à base 10 (1 octet) et un entier (4 octets) pour représenter la mantisse par exemple : la valeur -0.0089 (-89e-04), est représenté par le triplet suivant : signe(Sn) = 1, exposant (Ex) = 4, mantisse (H) = 89. La figure II.11 spécifie le format d'une table de définition de la transformation.

Table de définition de la transformation

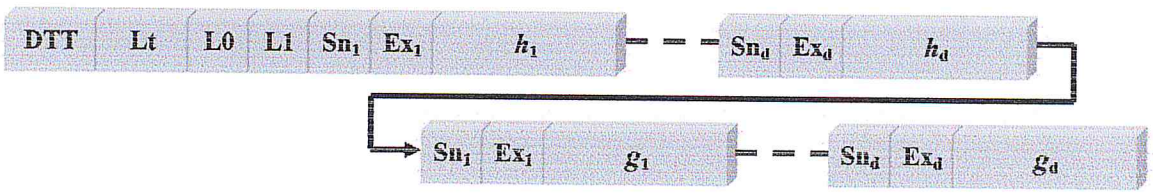

Figure II.11 Syntaxe de la table de définition de la transformation.

- **DTT**: (16 bits) Define Transform Table; marque le début de la table de définition de la transformation et sa valeur est toujours égale à 'FFA4h'.
- **Lt**: (16 bits) la longueur (en octet) de la table de définition de la transformation ; La taille de DTT n'est pas incluse.
- **L0**: (8 bits) le nombre de coefficients du filtre passe-bas (longueur de h); cette valeur est utilisée pour déterminer le nombre de coefficients du filtre h par la méthode suivante : Si la valeur de L0 est paire, alors le nombre de coefficients est égale à L0/2. Sinon, il est égale à (L0+1)/2.

- **L1**: (8 bits) le nombre de coefficients du filtre passe-haut (longueur de g) ; cette valeur est utilisée pour déterminer le nombre de coefficients du filtre g par la méthode suivante : Si la valeur de L1 est paire, alors le nombre de coefficients est égale à L1/2. Sinon, il est égale à (L1+1)/2.
- **Sn_k**: (8 bits) le signe du coefficient k. Si $Sn_k = 0$, le coefficient k est positif, sinon il est négatif.
- **Ex_k**: (8 bits) l'exposant du coefficient k; la virgule du coefficient k est décalée Ex_k positions vers la gauche.
- **h_k**: (32 bits) représente la mantisse du coefficient k du filtre passe-bas h.
- **g_k**: (32 bits) représente la mantisse du coefficient k du filtre passe-haut g.

II.4.5 Table de définition de la quantification

La table de définition de la quantification comporte l'ensemble des valeurs de quantification utilisées pour quantifier les coefficients de la TOD. Figure II.12 spécifie le format d'une table de définition de la quantification. Il est à signaler que les éléments de la table de quantification doivent être spécifiés dans le même ordre des sous-bandes, présentés dans Figure II.5.

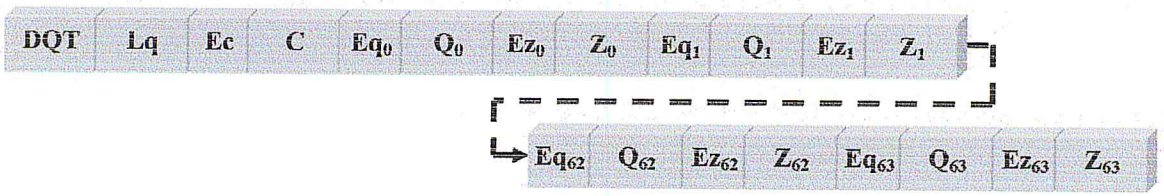

Figure II.12 Syntaxe de la table de définition de la quantification

Le marqueur et les paramètres représentés dans (Figure II.12) sont définis comme suit :

- **DQT**: (16 bits) Define Quantization Table; marque le début de la table de définition de la quantification et sa valeur est toujours égale à 'FFA5h'.
- **Lq**: (16 bits) la longueur (en octet) de la table de définition de la quantification; La taille de DTT n'est pas incluse, donc typiquement, sa valeur est égale à 389.
- **Ec**: (8 bits) l'exposant du paramètre C; la virgule du paramètre C est décalée Ec positions vers la gauche.
- **C**: (16 bits) définit le centre du l'intervalle de quantification C.

- **Eq_k**: (8 bits) l'exposant du l'élément Q_k; la virgule de l'élément Q_k est décalée Eq_k positions vers la gauche.
- **Q_k**: (16 bits) l'élément *k* de la table de quantification ; spécifie la largeur du pas de quantification de la sous-bande *k*.
- **Ez_k**: (8 bits) l'exposant du l'élément Z_k; la virgule de l'élément Z_k est décalée Ez_k positions vers la gauche.
- **Z_k**: (16 bits) l'élément *k* de la table de quantification; spécifie la largeur de la zone morte de la sous-bande *k*.

II.4.6 Table de définition du codage de Huffman

La table de définition du codage de Huffman est un ensemble de codes de longueur variable, utilisés par le codeur et le décodeur. La procédure de construction de ces codes a été prise de l'Annexe de spécification JPEG 'ISO JPEG DIS 10918-1'. Figure II.13 spécifie le format d'une table de définition du codage de Huffman.

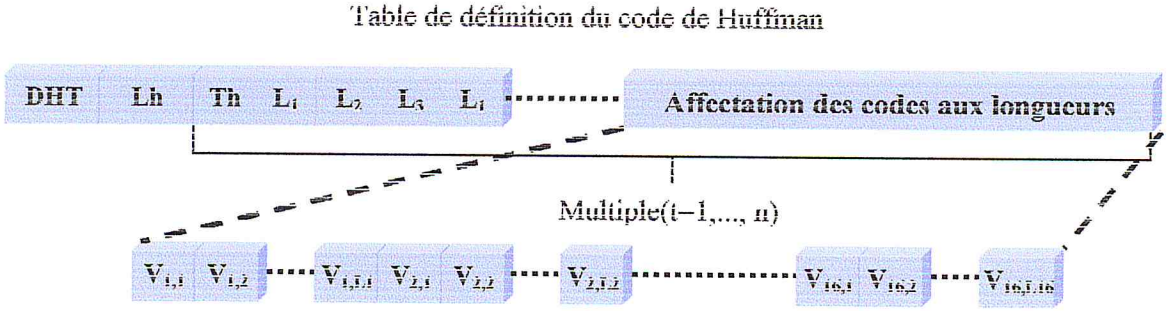

Figure II.13 Syntaxe de la table de définition du codage de Huffman.

Le marqueur et les paramètres représentés dans Figure II.13 sont définis comme suit :

- **DHT**: (16 bits) Define Huffman Table; marque le début de la table de définition du codage de Huffman et sa valeur est toujours égale à 'FFA6h'.
- **Lh**: (16 bits) la longueur (en octet) de la table du codage de Huffman; La taille de DHT n'est pas inclusc. La valeur de Lh varie d'une table à une autre car les codes utilisés pour chaque segment codes sont différents.
- **Th**: (8 bits) l'identifiant de la table du codage.

- L_i: (8 bits) le nombre de codes de taille i; spécifie le nombre de codes pour chacune des 16 longueurs possibles. Les éléments L_i sont les éléments de la liste BITS.
- V_{ij}: (8 bits) la valeur associée à chaque code de Huffman; Les éléments V_{ij} sont les éléments de la liste HUFFVAL.

II.4.7 Le segment de commentaire

Un segment de commentaire comporte un texte en clair qui peut être affiché par une application. Figure II.14 spécifie le format du segment marqueur d'un commentaire.

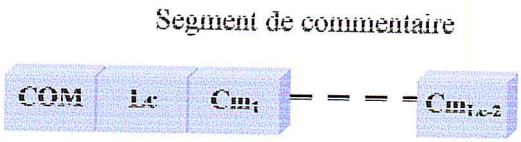

Figure II.14 Syntaxe du segment de commentaire.

- **COM :** (16 bits) Comment; marque le début d'un commentaire et sa valeur est toujours égale à 'FFA8h'.
- **Lc :** (16 bits) la longueur (en octet) du segment de commentaire. La taille de COM n'est pas incluse et sa valeur varie de 2 jusqu'à 65535.
- **Cm_i :** (8 bits) un octet du commentaire ; l'interprétation est à la charge de l'application.

II.5 Conclusion

Dans ce chapitre, nous avons posé en revue les spécifications du standard WSQ, et nous avons aborderons les plus importants fondements théoriques et pratiques. En particulier, nous avons présentons le fonctionnement de la transformation en ondelette, la quantification, et le codage entropique. De plus, on a présenté la structure générale du fichier .wsq, en détaillant ses différents segments.

Dans le chapitre suivant, nous allons présenter le principe de notre méthode de tatouage proposée, Il est constitué essentiellement de deux étapes principales : l'insertion de la marque et sa détection,

CHAPITRE III :
LA PROTECTION DES IMAGES BIOMETRIQUES PAR TATOUAGE NUMERIQUE

III.1. Introduction

Dans ce chapitre, nous aborderons les différents concepts utilisés pour la réalisation du système de sécurisation des données biométriques (visage et empreinte digitale) basé sur le tatouage numérique. L'approche développée comporte deux procédures à savoir. La procédure d'insertion et la procédure d'extraction de la marque.

Pour éviter la destruction de la marque par la compression, la procédure d'insertion est introduite au cours la compression WSQ, tandis que la procédure d'extraction est appliquée lors de la décompression.

III.2. Etat de l'art

Dans la littérature, peu de travaux sont publiés dans ce domaine. C'est un sujet d'actualité et date de quelques années. Dans cette partie, nous avons reporté les techniques de tatouage les plus utilisés dans la transformée en ondelette discrète (TOD). De plus, nous sommes intéressés aux travaux qui effectuent l'insertion de la marque après le processus de quantification. À la fin, nous allons citer les travaux qui réalisent l'insertion au cours du codage entropique.

III.2.1. Les techniques de tatouage numérique basées sur la transformation en ondelette discrète (TOD)

Ratha et al. [22] ont inséré des marques sur les images d'empreintes digitales compressées avec le standard de compression WSQ. Leur technique consiste à sélectionner des coefficients de la transformée en ondelette de l'image d'empreinte digitale.

Une clé k est générée en utilisant une fonction aléatoire, l'insertion est substitutive, elle est effectuée en modifiant le bit le moins signifiant (Least Significant Bit, LSB) des coefficients choisis. Dans cette méthode, les auteurs n'ont pas évalué les performances de robustesse et d'imperceptibilité [18]. Les principales étapes du processus d'insertion sont comme suit :

- **Le choix de l'ensemble des coefficients candidats:** compte tenu des coefficients quantifiés, le rôle de cette étape est de sélectionner tous les coefficients candidats où le changement dans le LSB est tolérable. Cette étape commence par l'exclusion de tous les coefficients appartenant aux

sous-bandes de basses fréquences. Ensuite, les coefficients ayant une large grandeur sont choisis.

- **Sélection des coefficients porteurs :** on entend par coefficients porteurs, les coefficients où la marque est insérée. Ces coefficients sont choisis de l'ensemble des coefficients candidats d'une façon aléatoire, en utilisant un générateur des nombres aléatoires. A noter, que le même générateur est utilisé dans le processus d'insertion, et est utilisé dans le processus d'extraction.
- **Insertion des bits de la marque :** le message à insérer est traduit en une séquence de bits, où chaque bit est inséré dans un coefficient porteur en modifiant la valeur de son bit le moins signifiant pour qu'il soit identique à la valeur du bit à insérer. Par exemple, si le bit le moins signifiant est 0 et le bit à insérer est 1, alors le bit le moins signifiant devient 1, et vice versa.

Le processus de décodage comporte trois étapes. Les deux premières sont similaires que celle du processus d'insertion. La troisième étape consiste à extraire la séquence de bits de la marque insérée.

Noore et al. [22] ont appliqué un algorithme de tatouage qui consiste à cacher deux marques : une image de visage et les données démographiques correspondant à chaque individu (le nom, le prénom, date de naissance,...). Ces dernières sont insérées dans les régions texturées sélectionnées appartenant aux sous bandes de l'image d'empreinte ayant subit une TOD

En premier lieu, la transformée en ondelette à deux niveaux est appliquée à l'image d'empreinte digitale. L'image du visage en niveau de gris est insérée dans les trois sous-bandes de haute fréquence du niveau deux (2) (c-à-d. LH_2, HL_2 et HH_2) (Figure III.1).

Tandis que les informations démographiques sont insérées dans les trois sous-bandes de haute fréquence du niveau un (1) (c-à-d. LH_1, HL_1 et HH_1). Les auteurs ont vérifié que l'insertion n'affecte pas l'exactitude de l'identification par contre aucun test de robustesse n'a été élaboré.

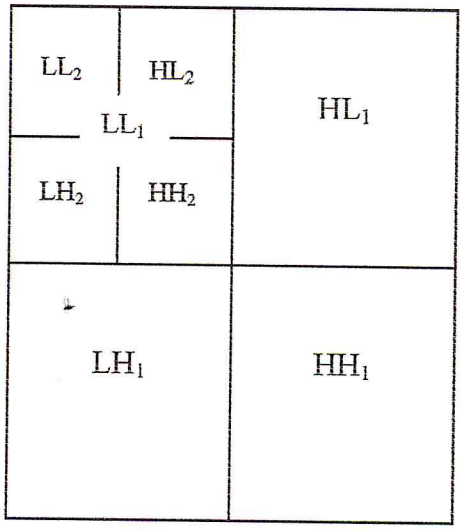

Figure III.1 La transformation en ondelette à deux niveaux.

III.2.2. La technique de tatouage numérique basée sur la quantification

Ces méthodes sont basées sur la Modulation d'Index de Quantification (QIM), elles consistent à utiliser la quantification pour insérer un message dans le signal hôte, et cela en utilisant un quantificateur choisi en fonction de la valeur du bit de message à insérer parmi un ensemble de quantificateurs préétablis. En d'autres termes, le rôle du quantificateur est de remplacer la valeur d'un coefficient par une valeur plus proche appartenant à un ensemble de valeurs discontinues prédéfinies [23].

Et puisque l'ensemble de données à insérer est composé de deux valeurs seulement (c-à-d 0 et 1), l'ensemble des quantificateurs est composé de deux quantificateurs $q_0(.)$ et $q_1(.)$, décalé par un pas Δ l'un par rapport à d'autre.

De plus, ces méthodes sont basées sur l'approche d'insertion **Pair-Impaire** [21]. En d'autres termes, les multiples impairs de Δ représentent une valeur de bit, par exemple 1, et les multiples pairs représentent l'autre, par exemple 0. On considère le problème le plus simple d'insertion d'un bit dans un échantillon, Soient $b \in \{0, 1\}$ le bit à insérer, et $x \in \Re$ l'échantillon original et $y \in \Re$ l'échantillon marqué. Le quantificateur $q_1(x)$ (représenté par croix dans la figure III.2) représente des multiples impairs de Δ, qui sont utilisés pour insérer un bit de données 1, tandis que le quantificateur q_0 (représenté par cercle dans la figure III.2) correspond à un multiple pair de Δ, ce qui représente un bit de données 0 [21].

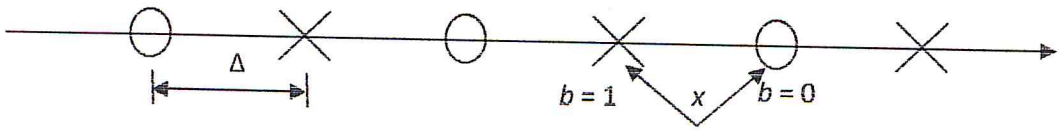

Figure III.2 Insertion d'un bit dans un échantillon en utilisant l'approche de la modélisation d'index de quantification.

Les deux quantificateurs q_0 et q_1 sont formulés comme suit:

$$q_0(x) = \begin{cases} x; & \text{si } mod\left(\left\lfloor \frac{x}{\Delta} \right\rfloor, 2\right) = 0 \\ x+1; & \text{si } mod\left(\left\lfloor \frac{x}{\Delta} \right\rfloor, 2\right) = 1 \end{cases} \quad \text{(III.1)}$$

Et

$$q_1(x) = \begin{cases} x+1; & \text{si } mod\left(\left\lfloor \frac{x}{\Delta} \right\rfloor, 2\right) = 0 \\ x; & \text{si } mod\left(\left\lfloor \frac{x}{\Delta} \right\rfloor, 2\right) = 1 \end{cases} \quad \text{(III.2)}$$

Où : x représente le coefficient qui va porter un bit de la marque.

Δ représente dans ce cas le pas de quantification de la compression WSQ (Q_k) de la sous-bande k où le coefficient x appartient.

L'insertion est réalisée par la quantification de l'échantillon x en utilisant l'un des deux quantificateurs sélectionnés en fonction de la valeur du bit à insérer b [21]. Ainsi, l'échantillon tatoué y est défini comme suis:

$$y = \begin{cases} q_0(x); & si\ b = 0 \\ q_1(x); & si\ b = 1 \end{cases} \quad \text{(III.3)}$$

Le processus d'extraction est basée sur la distance minimale (c'est à dire qu'il trouve le point de quantification le plus proche à y). En d'autres termes, le décodage est effectué par la quantification de l'échantillon tatoué au point le plus proche reconstruit des deux quantificateurs. Un point pair de reconstruction, indique qu'un bit 0 a été inséré. De même, un point impair de reconstruction signifie qu'un bit 1 a été inséré [23]. Par conséquent, le bit extrait $\hat{b}$ est donné par:

III.3.1 Le principe de la méthode proposée

L'approche développée est appliquée à la sécurisation des données biométriques visages et empreintes compressés pas le standard WSQ, module incorporé dans le system d'identification automatique des empreintes digitales (AFIS) exploité par la gendarmerie nationale.

Le synoptique du système de tatouage proposé comprend six (06) modules à savoir (figure III.3) : le module d'acquisition des images biométriques, le module de compression WSQ, le module d'insertion de la marque en utilisant une clé, le module de stockage, le module d'extraction en utilisant la même clé, et le module de décompression.

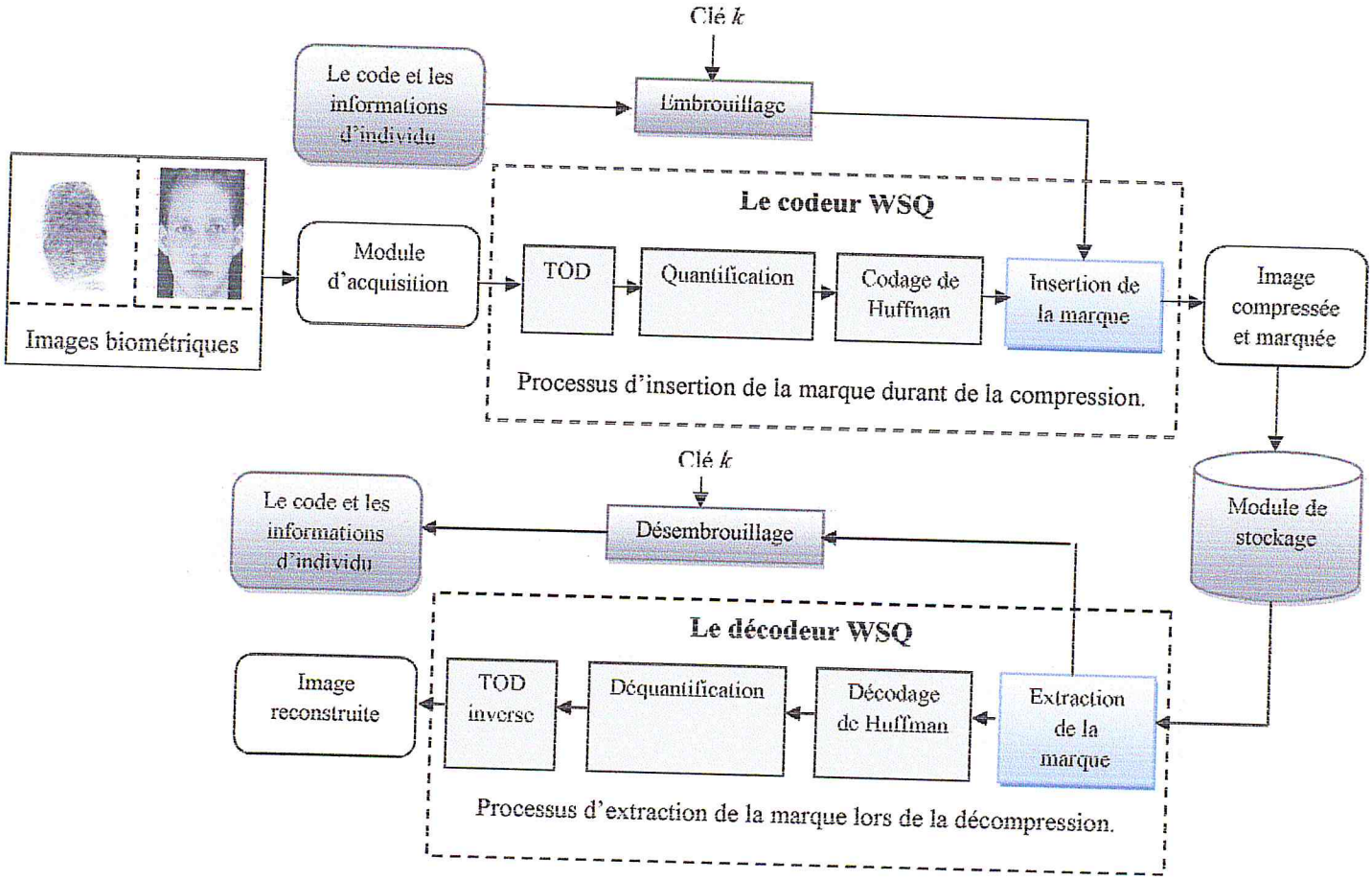

Figure III.3 Schéma général du système proposé.

Le but de la méthode proposée est de remédier au problème exposé par la gendarmerie nationale qui concerne la sécurisation des noms des fichiers d'images de visages et d'empreinte digitales constituant le seul lien entre ces derniers et les personnes correspondantes. En autres termes, la correspondance entre le fichier

biométrique et la personne concernée. Donc les solutions proposées consistent à considérer les noms des fichiers et les informations correspondants comme marques et les images biométriques comme signal porteur (Figure III.4).

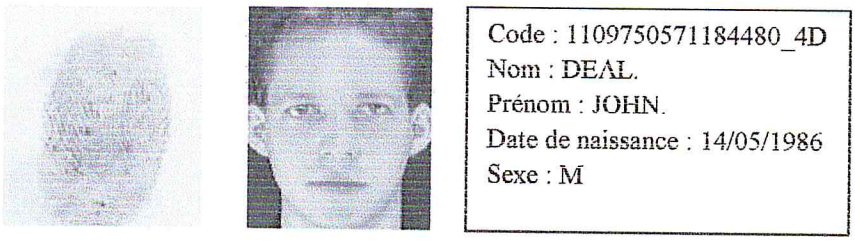

Figure III.4 (a) image d'empreinte digitale, (b) image de visage, (c) la marque insérée.

Si le nom de fichier est modifié, soit par des fausses manipulations faites par l'opérateur ou par des attaques de virus ou programme malicieux (Figure III.5), le nom de fichier est extrait de l'image biométrique et est utilisé pour identifier la personne correspondante.

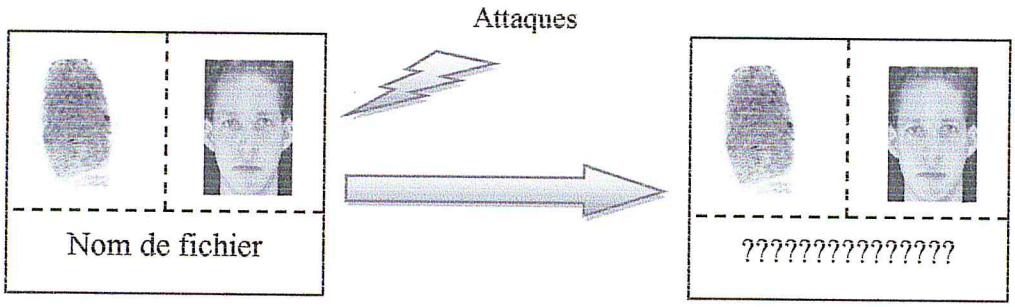

Figure III.5 Modification des noms des fichiers.

Comme les images biométriques tel que l'image d'empreinte digitale sont compressées selon le standard WSQ. Et comme la compression est considérée comme une attaque involontaire, qui détruit la marque insérée, ceci nous conduit à insérer la marque au cours de la compression.

En analysant les différents processus dans la compression WSQ, nous avons noté que le processus de quantification modifie d'une façon irréversible les coefficients de la TOD. En plus les algorithmes d'insertion qui ne sont pas applicables dans le flux compressé exigent la décompression totale ou mieux une

décompression partielle pour extraire la marque, et ceci représente une tâche lourde qu'il vaut mieux éviter [22].

A cet effet, nous proposons deux solutions d'insertion de la marque dans le module de codage entropique. Ces dernières se composent des codes à longueurs variables (VLCs) qui représentent les différentes sous-bandes de l'image, en excluant les régions de la zone morte. Les méthodes ainsi proposées considèrent que les manipulations probables sur les systèmes AFIS n'altèrent pas les images biométriques.

Les techniques de tatouage appartiennent au schéma fragile. La marque qui est constituée du nom de fichier est substituée aux mots de code binaires VLC, la solution adoptée est caractérisée comme suit:

- La marque insérée est une séquence de bits binaires.
- L'image d'origine n'est pas nécessaire pour l'extraction de la marque (tatouage aveugle).
- La détérioration ou la perte totale de la marque en cas des fraudes ou manipulations possibles sur les images biométriques marquées (tatouage fragile).
- La clé d'insertion est nécessaire au décodage pour l'extraction du message (tatouage symétrique).

III.4 Le processus d'insertion de la marque

III.4.1 Génération de la séquence des bits de la marque

Comme nous avons déjà cité, le but est d'insérer le code d'individu dans l'image biométrique compressée. Le code d'individu dans le système AFIS de la gendarmerie nationale comporte 19 caractères qui sont structurés comme suit [21]:

$$x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}x_{12}x_{13}x_{14}x_{15}x_{16}yzt$$

Où : $x_i \in \{0,1,2,3,4,5,6,7,8,9\}$, $i = 1,2,\dots,16$.

$y \in \{\_\}$.

$z \in \{P, 4, 2\}$.

$t \in \{G, D, 1,2,3,4,5,6,7,8,9, 10\}$.

On propose que chaque caractère converti en une séquence de quatre (04) bits binaires, les valeurs sont données dans le tableau III.1 pour le système AFIS de la gendarmerie nationale. A la fin de conversion, on obtient la séquence de bits à insérer β tel que :

$$\beta = \{\beta_\mu : \beta_\mu \in \{0,1\},\ 1 \leq \mu \leq n\} \qquad \text{(III.5)}$$

Caractère	Séquence des bits
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
_	1011
P	1100
G	1101
D	1110

Tableau III.1 Les valeurs des séquences de bits pour le système AFIS.

III.4.2 Embrouillage de la marque

Afin d'augmenter la sécurité d'insertion, la marque est embrouillée avant d'être insérée. L'embrouillage est une opération destinée à transformer un signal numérique en un signal numérique aléatoire, de même signification et de même débit binaire. Le principe de l'approche est donné comme suit :

Soit k la taille de la marque et k' la taille de la clé, le nombre d'itérations n est calculé d'après l'équation suivante :

$$n = \begin{cases} k & si \quad k > k' \\ k' & sinon \end{cases} \qquad \text{(III.6)}$$

Tant que 1 < n **faire**

Si (clé [I *mod k'*] == 1) **alors** Changer (marque [I *mod k*])

Fait.

Cet algorithme est parfaitement schématisé ci-dessous (figure III.6)

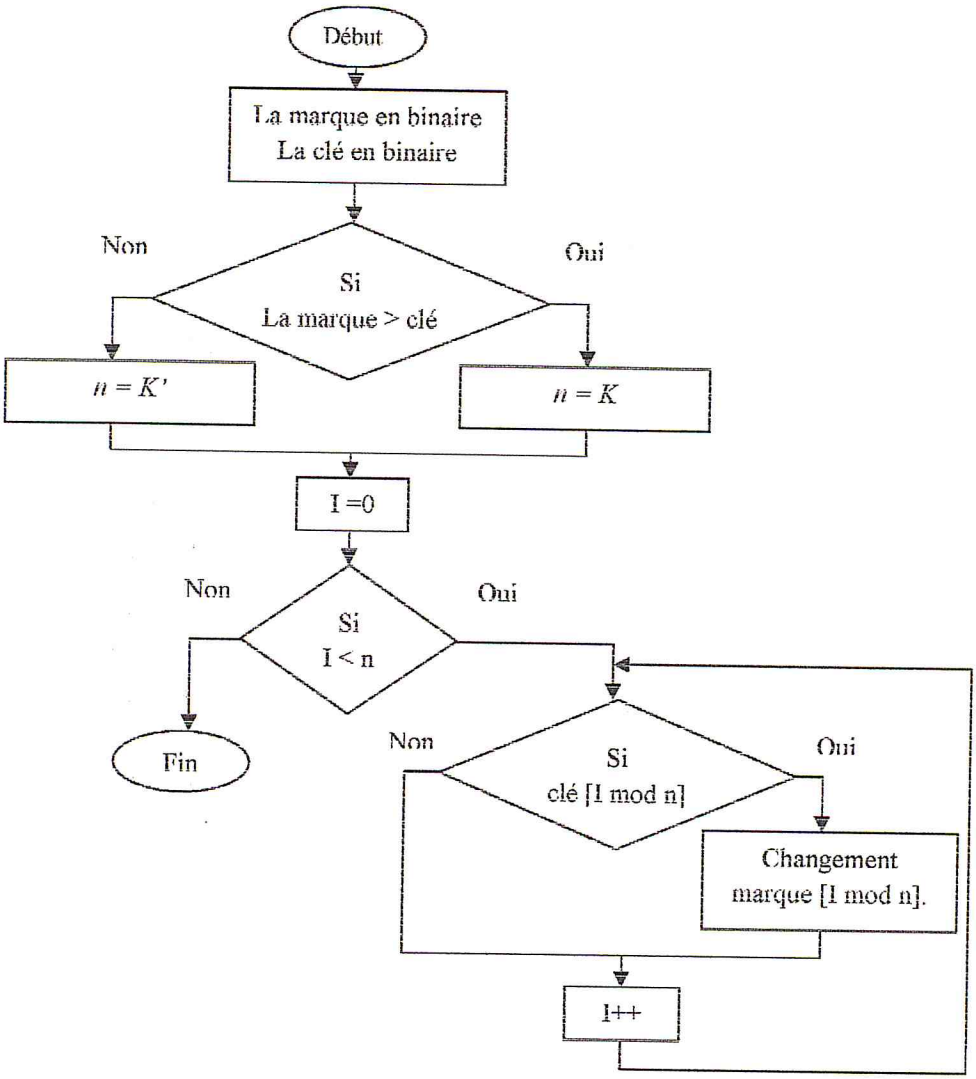

Figure III.6 Organigramme d'embrouillage de la marque.

III.4.3 Description de la première solution proposée

Le codeur code chaque symbole dans la table de Huffman par un mot de code VLC, tel que les différents mots de code n'ont pas nécessairement la même longueur. Un VLC est constitué de deux champs RUN/LEVEL. Le champ binaire RUN est non signé de longueur fixe (8 bits), le champ LEVEL est signé de longueur variable (maximum 16 bits).

On note l_i la longueur du mot de code c_i, et l_j la longueur du mot de code c_j le successeur du code c_i tel que $l_i < l_j$. La distribution des longueurs du code est donc $\{l_1, l_2, \ldots, l_N\}$.

Le principe consiste à générer à partir des mots de codes VLC un ensemble des codes non utilisés. Nous avons proposé deux méthodes pour marquer les images biométriques:

Le processus d'insertion de la marque intervient juste après la table de Huffman du processus de compression WSQ. Plus précisément, on insère la séquence de bits de la marque dans la table VLC. Les étapes du processus d'insertion de la première méthode proposée sont représentées dans l'organigramme de la figure III.6 ci-dessous.

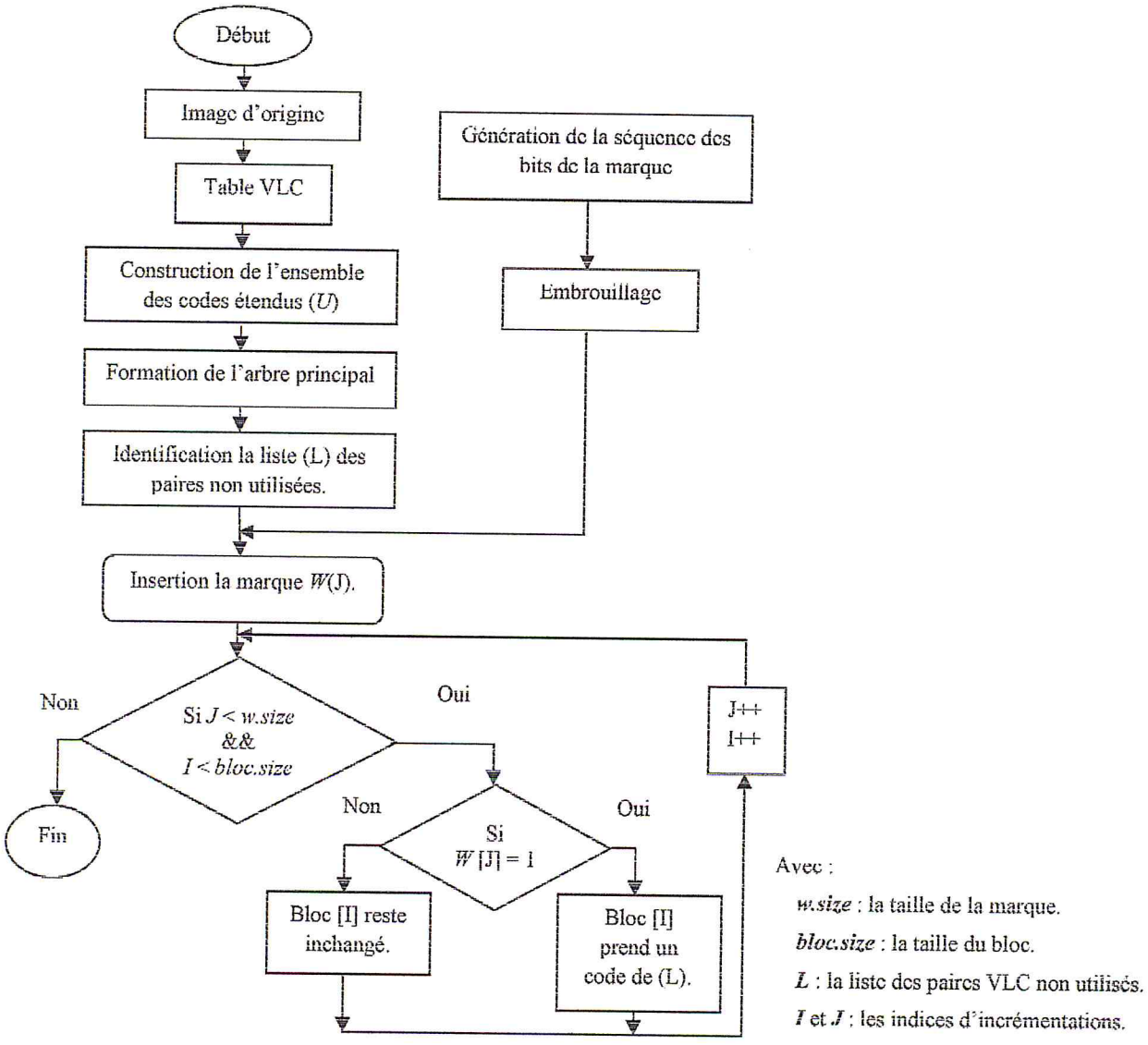

Figure III.7 Organigramme d'insertion de la marque (première méthode).

III.4.3.1 Construction de l'ensemble des codes étendus (U)

Soit la table VLC qui se compose de N codes représentés par $V = \{v_1, v_2, \dots, v_N\}$ où v_i est le i^{ieme} code de longueur l_i.

Un élément de l'ensemble des codes étendus (U) appelé « code space » est construit en appariant des codes VLC délivrés par les tables de Huffman comme suit :

$$U = \{u_{mn}\}, \qquad m, n \in 1, \dots, N, \tag{III.7}$$

Avec

$$u_{m,n} = \{V_m, V_n\} \tag{III.8}$$

Où : U se compose de N^2 mots de code dont la longueur varie entre $2l_1$ à $2l_N$.

III.4.3.2 Formation de l'arbre principal « codetree »

Les « codetrees » pour les codes à longueurs variable (VLC) sont des arbres binaires où les paires VLC occupent les feuilles des nœuds. La racine de l'arbre V est au niveau 0 et consiste de l_N niveaux. Chaque niveau l contient jusqu'à 2^l nœuds, et le nombre des nœuds maximum dans l'arbre est égal à 2^{l_N}. Dans l'arbre, ils existent des nœuds libres (voir la figure III.7), on utilise ces nœuds pour le tatouage.

Dans la figure III.7, un exemple d'arbre des paires VLC V= {00, 010, 0110} est présenté. Les nœuds en noirs représentent des éléments de V et les nœuds indiqués par cercles pointillés ne sont pas disponibles. Le côté droit de l'arbre est non utilisé et contient les nœuds libres.

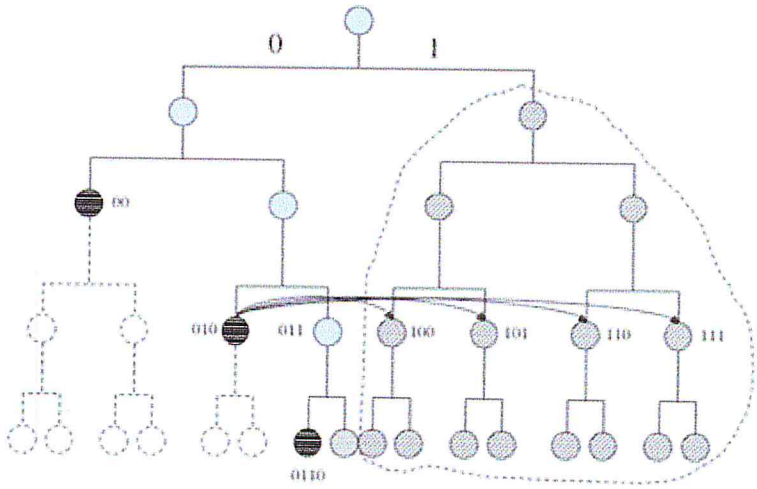

Figure III.8 Exemple d'arbre des paires VLC.

III.4.3.3 Identification la liste des paires non utilisés

Après la construction de l'arbre principal (U) des paires VLC, la table VLC est subdivisée en blocs B_i de telle façon que le nombre des blocs est égale à la longueur de la marque W. Puis, pour chaque bloc B_i, un arbre des paires VLC (U_{Bi}) et construit. Par la suite, l'opération XOR et appliquée sur l'ensemble principal U et les ensembles U_{Bi} afin de déterminer la liste des paires non utilisés.

III.4.3.4 Insertion de la marque

La liste des paires non utilisées garantit la disponibilité des paires qui ne figurent pas dans n'importe quel bloc. Le codeur visite tous les blocs sélectionnés selon une clé. Pour insérer un bit « 1 » le bloc est forcé de contenir une paire de la liste des paires non utilisé. Pour insérer un bit « 0 » le bloc reste inchangé.

III.4.3.5 Extraction de la marque

Afin de récupérer les données insérées, à la réception le décodeur doit savoir quelles sont les paires VLC utilisés pour le tatouage. Pour cela il utilise l'ensemble principal (U) des paires VLC, et le pointeur d'indexation envoyé à l'entête du flux compressé (.WSQ).

Le décodeur utilise ce pointeur pour identifier les paires non utilisés. Lors de la visite des blocs B_i, le décodeur détermine, si le bloc contient une paire de la liste sélectionné par la clé. Si la réponse est positive, le bloc porte un bit « 1 », si non il porte un bit « 0 ».

L'inconvénient de cette solution est le transfert des positions d'insertion comme commentaire dans l'entête du flux compressé. Ce qui conduit à l'augmentation de sa taille, donc l'augmentation de la bande passante de transmission. Afin de résoudre cet inconvénient, nous avons proposé une deuxième solution.

III.4.4 Description de la deuxième solution proposée

La deuxième solution comporte trois étapes. Elles sont représentées dans l'organigramme de la figure III.8 : les deux premiers étapes qui sont la construction de l'ensemble des codes étendus (U) et la formation de l'arbre principal sont les mêmes que les deux premiers étapes décrites dans la première solution. Alors que la troisième étape qui constitué l'insertion de la marque est différente que celle décrite dans la première solution, son principe est détaillée ci-dessous :

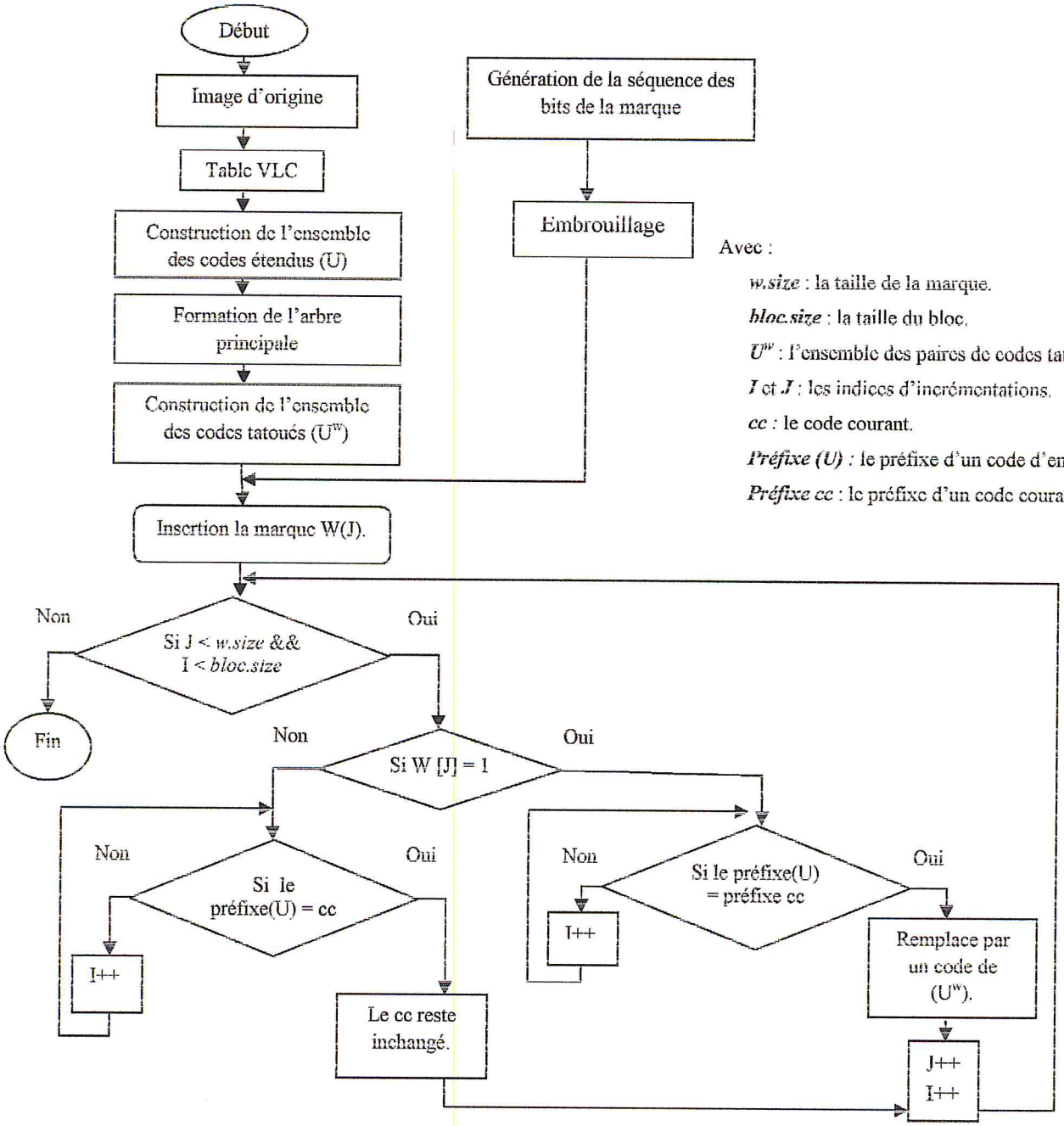

Figure III.9 Organigramme d'insertion de la marque (deuxième solution).

III.4.4.1 Construction de l'ensemble des paires de code tatouées (U^w)

L'ensemble U^w contient les paires de code VLC utilisés pour le tatouage. Cet ensemble est généré à partir de l'ensemble U, sa génération est comme suit :
On prend chaque paire u_{ij} de l'ensemble U et on fait un décalage du $k^{ième}$ bit de gauche à droite pour obtenir une nouvelle paire u_{ij}^k. Dans le cas où la partie droite n'est pas suffisante, on fait le décalage inverse de droite à gauche. Cette opération

déplace le nœud u_{ij} à un autre nœud u_{ij}^{k} dans le même niveau en satisfaisant les conditions suivantes :

- $u_{ij}^{k} \notin u_{mn}$ *et* $u_{mn} \notin u_{ij}^{k}$, ces deux conditions assurent qu'aucune paire de la marque ne viole la condition de préfixe (c'est-à-dire aucun code u_{ij}^{k} n'est un préfixe d'un code u_{mn}).
- $u_{ij}^{k} \notin u_{mn}^{p}$ *et* $u_{mn}^{p} \notin u_{ij}^{k}$, ces conditions sont nécessaires pour éviter les collusions. Une collusion se produit quand une paire VLC marquée devient un préfixe d'une autre paire VLC marquée ou non marquée.
- Il n'existe pas deux mots de code VLC dont le processus de décalage donne la même paire de code.
- La longueur du nouveau VLC (en bits) doit être inférieure ou égale à l'originale. Ceci afin d'éviter l'augmentation de la taille du flux compressé. Pour être plus précis, la valeur de la table de Huffman *RUN* du nouveau VLC reste la même et le changement se fait au niveau des bits du code *LEVEL*. Cette dernière doit être minimale.

Pour implémenter les paires marquées de VLC dans l'arbre, il est essentiel de savoir :

(i) les nœuds de VLC au même niveau qui sont disponibles pour l'insertion de la marque.

(ii) les nœuds qui sont sur des trajectoires de collusion et doivent être évités.

La figure III.8 illustre un exemple d'insertion V= {00, 010, 0110}, l'ensemble des paires de codes étendu U correspond contient {0000, 00010, 000110, 01000, 010010, 0100110, 011000, 0110010, 01100110}.

Puisque aucune paire de code ne commence par « 1 », la moitié droite de l'arbre est vide. Pour déterminer les mots de code tatoués de l'ensemble U^{w}, on commence par décaler le 1er bit le moins significatif (LSB) du mot de code « 0000 ». Ce dernier prend la valeur « 0001 ». Malheureusement, le code généré provoque une collusion avec les paires de code {00010, 000110}. Ceci provoque la violation des conditions de préfixe (1) et (2), (voir la figure III.9).

Pour éviter la collusion, nous procédons au décalage du 2^{iéme} bit du LSB. Donc le mot de code « 0000 » devient « 0010 ». Ce nouveau code vérifie les quatre conditions citées (voir la figure III.10).

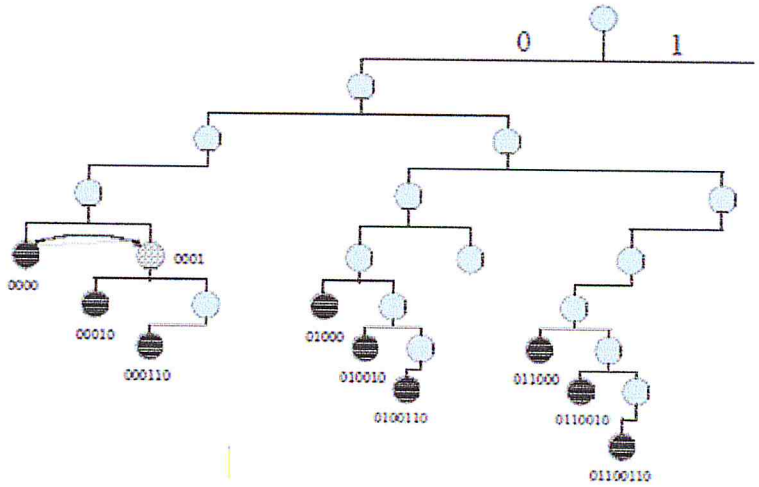

Figure III.10 La collusion provoquée par le tatouage de « 0000 » à « 0001 ».

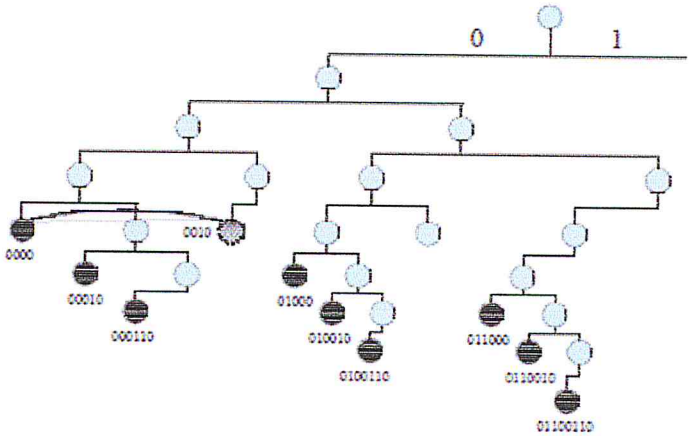

Figure III.11 Le tatouage de « 0000 » à « 0010 » sans collusion.

Note : - Afin d'assurer l'invisibilité de la marque, nous avons effectué un décalage au maximum de deux (2) bits du LSB.

III.4.4.2 Insertion de la marque

L'insertion est effectuée selon la valeur du bit de la marque : si le bit à insérer est un « 1 », le codeur cherche dans l'ensemble U un mot de code dont le préfixe est le code courant, et il le remplace ensuite par une paire de code appartenant à l'ensemble U''.

Si le bit à insérer est un « 0 », le codeur cherche le mot de code dans le préfixe est le code courant. S'il existe, le code courant reste inchangé. Sinon il passe au codc suivant.

III.5 Processus d'extraction de la marque

Le but de ce processus est d'extraire bit par bit la séquence de marque insérée et de reconstruire par la suite le code de l'individu. Dans la pratique, si l'image marquée a été modifiée par une des attaques, les informations cachées ne peuvent pas être récupérées intégralement. Inversement au processus d'insertion, ce processus intervient avant le processus de décodage de Huffman du décodeur WSQ. Il comporte les étapes suivantes :

III.5.1 Extraction des bits de la marque

Afin de récupérer les données insérées. Le décodeur a déjà l'ensemble V des paires de codes VLC. Il utilise cet ensemble pour construire d'une manière similaire comme le processus d'insertion les deux ensembles U et U^w. Il examine ensuite, si le préfixe du code courant $préfixe\_C_c(i)$ appartient à l'ensemble V et tout le code courant $C_c(i)$ appartient à l'ensemble U^w alors le bit extrait est un « 1 ». Si le préfixe de code courant $préfixe_{C_c}(i)$ appartient à l'ensemble V et le code courant $C_c(i)$ n'appartient pas à l'ensemble U^w, alors le bit extrait est un « 0 ».

Le processus d'extraction se résume comme suit :

$$w(i) = \begin{cases} 1 & si\ préfixe\_C_c(i) \in V\ et\ C_c(i) \in U^w \\ 0 & sinon \end{cases} \qquad \text{(III.9)}$$

Où : w(i) représente le bit de la marque.

$C_c(i)$ représente le code courant.

U^w l'ensemble des paires de code tatoués.

Une fois la marque est extraite, elle est désembrouillée par le processus inverse d'embrouillage.

III.5.2 Reconstruction du code d'individu

Le code de l'individu est reconstruit à partir de la séquence binaire extraite et désembrouillée. Ceci en utilisant les correspondances des valeurs inscrites dans la génération de la marque (tableau III.1) pour le système AFIS de la Gendarmerie Nationale.

III.6 Conclusion

Dans ce chapitre, nous avons décrit en détail les solutions proposées pour faire face à la problématique posée. Basée sur les techniques du tatouage numérique, ces solutions proposent d'insérer des marques dans les images biométriques compressées par le standard WSQ. Et pour diminuer l'effet de la compression sur les marques insérées, nous avons proposé d'intégrer le module d'insertion après le module de codage entropique. Le module d'extraction ne nécessite pas l'image d'origine lors de la détection (un système de tatouage aveugle).

Dans le prochain chapitre, nous allons présenter les différentes phases de la mise en œuvre de l'application développés avec des tests et résultats qui évalueront les performances de cette méthode.

CHAPITRE IV :
RESULTATS ET TESTS

IV.1 Introduction

L'objectif de ce projet est la sécurité des données biométriques par la technique du tatouage numérique, et cela lors de sa transmission d'un module à un autre ou pour la vérification de l'intégrité des images sauvegardées dans le module de stockage. Et pour atteindre cet objectif, nous avons implémenté deux modules. Le premier module permet l'insertion de la marque, qui est le code et les informations d'individu. Il a été intégré au cours de la compression WSQ, plus précisément après le module de codage de Huffman. Tandis que le deuxième module, qui a été intégré avant le processus de décompression et avant le module de décodage de Huffman, permet l'extraction de la marque insérée (c.à.d. le code d'individu).

Dans ce chapitre nous allons exposer les résultats auxquels nous sommes parvenus en appliquant la méthode de tatouage définie dans le chapitre précédant. Nous allons évaluer ici les performances de la méthode de tatouage en termes d'invisibilité et d'intégrité. Dans la dernière section, nous exposerons l'interface de notre application et ses différentes fonctionnalités qu'elle offre à travers quelques prises d'écrans illustratives.

IV.2 Analyse expérimentale des résultats

Dans cette section, nous avons effectué des tests sur notre système de protection des images biométriques, nous allons présenter quelques résultats obtenus lors d'une série de tests sur l'ensemble des images biométriques des tailles différentes codées par 8 bits par pixel.

L'évaluation des performances est principalement portée sur deux points principaux :

1- L'imperceptibilité de la marque, évaluée en calculant le PSNR (Peak Signal to Noise Ratio) [21], et le SSIM (Structural SIMilarity) [19].

2- La performance d'extraction, évaluée en calculant le BER (Bit Error Rate) [21].

IV.2.1 Analyse de l'imperceptibilité

IV.2.1.1 Le PSNR

Pour mesurer la similitude entre l'image d'origine et celle tatouée, le PSNR a été utilisé. Il est défini par la formule suivante :

$$PSNR = 20\log_{10}\left(\frac{N * M * 255^2}{\sum_{i=1}^{N}\sum_{j=1}^{M}[I(i,j) - I'(i,j)]^2}\right) \quad \text{(IV.1)}$$

Où : I et I' représentent l'image d'origine et l'image compressée tatouée de même taille $N*M$ respectivement.

IV.2.1.2 Le SSIM

L'idée de SSIM est de mesurer la similarité de structure entre les deux images, plutôt qu'une différence pixel à pixel comme le fait par exemple le PSNR. L'hypothèse sous-jacente est que l'œil humain est plus sensible aux changements dans la structure de l'image. La métrique SSIM est défini par la formule suivante :

$$SSIM(I,I') = \frac{(2\mu_I\mu_{I'} + C_1)(2\mathrm{cov}_{II'} + C_2)}{(\mu_I^2 + \mu_{I'}^2 + C_1)(\sigma_I^2 + \sigma_{I'}^2 + C_2)} \quad \text{(IV.2)}$$

Avec :

μ_I la moyenne de I.

$\mu_{I'}$ la moyenne de I'.

σ_I^2 la variance de I.

$\sigma_{I'}^2$ la variance de I'.

$\mathrm{cov}_{II'}$ la covariance de I et I'.

$C_1=(K_1L)^2$, $C_2=(K_2L)^2$ deux variables destinées à stabiliser la division quand le dénominateur est très faible.

L *est* la dynamique des valeurs des pixels, elle est égale à 255 pour des images codées sur 8 bits.

$K_1=0.01$ et $K_2=0.03$ par défaut.

Nous avons effectué des tests sur l'mage de référence *Homme* de taille 256x256 pixels et codé sur 8 bits par pixels.

Le tableau IV.1 représente les résultats de la compression effectuée sur l'image *Homme* en se basant sur le standard de compression WSQ. Les résultats obtenus sont exprimés en fonction du PSNR et le SSIM.

Image d'origine	Image compressée	PSNR (dB)	SSIM
		44,126	0,7374282

Tableau IV.1 Table de résultat obtenu après la compression.

En faisant référence au schéma de tatouage implémenté, le choix de la position d'insertion est très important. Afin de trouver les bonnes positions d'insertion, nous effectuons des tests sur l'mage de référence *Homme* de taille 256x256 pixels et codé sur 8 bits par pixels. La marque à insérer est un code de taille 76 bits.

Le Tableau IV.2 représente les résultats de tatouage effectué sur les trois (03) blocs du codage entropique. Le bloc 1 contient les sous-bandes de 0 à 18, le deuxième contient les sous-bandes de 19 à 51 et le bloc 3 contient les sous-bandes de 52 à 59.

Les résultats obtenus sont exprimés en fonction du PSNR et le SSIM.

Image d'origine	Image tatouée	PSNR (dB)	SSIM
 (256*256)	 Bloc 1	38,33028	0,4932749
	 Bloc 2	41,05027	0,6603472
	 Bloc 3	43,3064	0,7253349

Tableau IV.2 Table des résultats obtenus selon le choix d'insertion.

A partir du tableau ci-dessus, nous remarquons que les résultats obtenus dans le bloc 3 sont meilleurs résultats objectif et subjectif au point de vue qualité visuelle de l'image (imperceptibilité de la marque) par apport les deux autres blocs (bloc 1 et bloc 2). La dégradation visuelle est vraiment perceptible sur les images tatouées du bloc 1 et bloc 2.

Nous avons effectué des tests sur deux (2) types d'images biométriques. Le premier type, les images de visage de taille 320*400 pixels sont codées sur 8 bits par pixels. Le deuxième type, les images d'empreintes digitales de taille 248*292 pixels sont codées sur 8 bits par pixels. La marque à insérer est un code et les informations nécessaires de l'individu.

Les tableaux IV.3 et IV.4 représentent les résultats du tatouage effectué sur les images biométriques (visage et empreinte digitale) après la compression WSQ.

Les résultats obtenus sont exprimés en fonction du PSNR, le SSIM, la capacité d'insertion maximale, et les tailles des images après la compression et après d'insertion.

Image d'origine	Taille en Ko		PSNR (dB)	SSIM	Capacité (bits)
	Image compressée	Image compressée tatouée			
 (320*400)	7,42	7,43	44,380	0,838	880
 (320*400)	7,48	7,49	45,534	0,838	944
 (320*400)	7,39	7,40	45,728	0,837	912

Tableau IV.3 Table des résultats obtenus après l'insertion dans les images de visage.

Image d'origine	Taille en Ko		PSNR (dB)	SSIM	Capacité (bits)
	Image compressée	Image compressée tatouée			
 (248*292)	5,44	5,49	45,185	0,817	768
 (248*292)	5,30	5,36	37,711	0,892	752

Tableau IV.4 Table des résultats obtenus après l'insertion dans les images d'empreintes digitales.

A partir des tableaux ci-dessus, et au point de vue imperceptibilité des marques, on peut dire que les valeurs du PSNR et SSIM sont acceptables si on fait une comparaissons avec l'article paru dans [23]. Ainsi, la capacité d'insertion est maximale, elle varie selon la taille des images biométriques de telle façon à obtenir un bon compromis entre l'imperceptibilité de la marque et son intégrité.

IV.2.2 Analyse des performances

Pour mesurer la performance du décodeur, nous avons évalué l'exactitude de la marque extraite avec le Bit Error Rate (BER), qui représente le taux d'erreurs de la marque extraite. Il est calculé en divisant le nombre des bits erronés de la marque extraite par le nombre total des bits de la marque insérée. La métrique BER est défini par la formule suivante :

$$BER = 100 - (\text{nombre bits erroné} / \text{nombre des Bits}) \qquad \text{(IV.3)}$$

Pour analyser les performances du processus d'extraction, nous avons effectué des tests sur les images biométriques des différentes tailles.

Le tableau IV.5 représente les résultats obtenus, ces résultats sont exprimés en fonction du BER, la marque insérée et la marque extraite.

Image d'origine	La marque insérée	La marque extraite	BER %
 (320*400)	1109750571184480_4D Smith Julia F 10/01/1979	1109750571184480_4D Smith Julia F 10/01/1979	0
 (248*292)	1321750575434480_4D Brown John M 31/10/1982	1321750575434480_4D Brown John M 31/10/1982	0

Tableau IV.5 Table des résultats obtenus du BER.

A partir du tableau ci-dessus, nous remarquons que les valeurs du BER obtenus sont nulles (BER= 0 %) pour tous les tests effectués sur les images. Donc, les informations insérées sont à 100% extraites. Ce qui est l'objectif de la solution proposée.

IV.3 Présentation de l'application

Nous allons présenter dans cette section l'environnement de programmation utilisé, ensuite nous exposerons l'interface de l'application ainsi que les fonctionnalités qui permettent d'accomplir notre application afin de facilité son utilisation et cela par le prises d'écrans.

IV.3.1 Environnement de programmation

Pour implémenter notre application, nous avons utilisé le langage de programmation Visuel C++ 2008 qui est un environnement de développement intégré Windows conçu par Microsoft. Il fait partie de la suite des logiciels Visuel Studio. En effet, Visuel C++ 2008 intègre différents outils pour développer et debugger un programme en C++ s'exécutant sous le système Windows. Visuel C++ 2008 permet aussi de réaliser, de façon très simple, les interfaces des applications et de relier facilement le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événements système, etc.).

Nous avons choisi le Visuel C++ pour le développement de notre application, car il offre plus de possibilités pour le traitement d'image. Au plus, une exécution rapide des programmes.

IV.3.2 Interface de l'application

Notre application comporte une fenêtre « menu principal» (figure IV.1) et des sous fenêtres « forms ». Le menu principal de l'application nommé « Sécurisation des données biométriques par tatouage numérique » permet le chargement de l'image (visage ou empreinte), puis le choix du traitement à effectuer sur cette image : compression/décompression ou insertion/extraction et métriques.

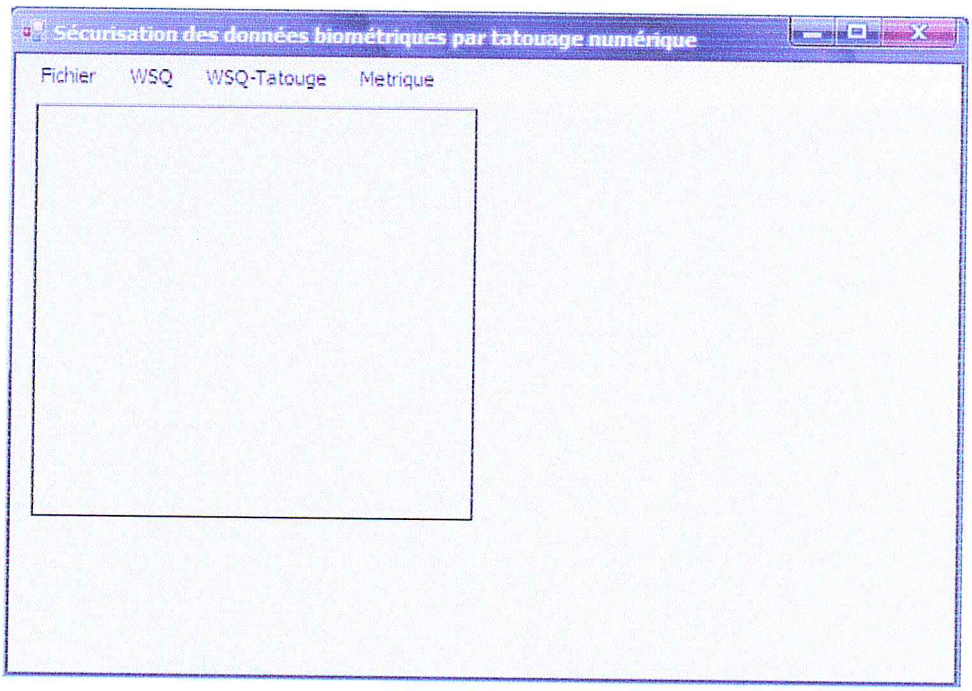

Figure IV.1 Menu principale.

Ainsi, on peut aussi calculer le PSNR (figure IV.2) et le SSIM (figure IV.3) pour analyser l'imperceptibilité de la marque.

Figure IV.2 calcul le PSNR.

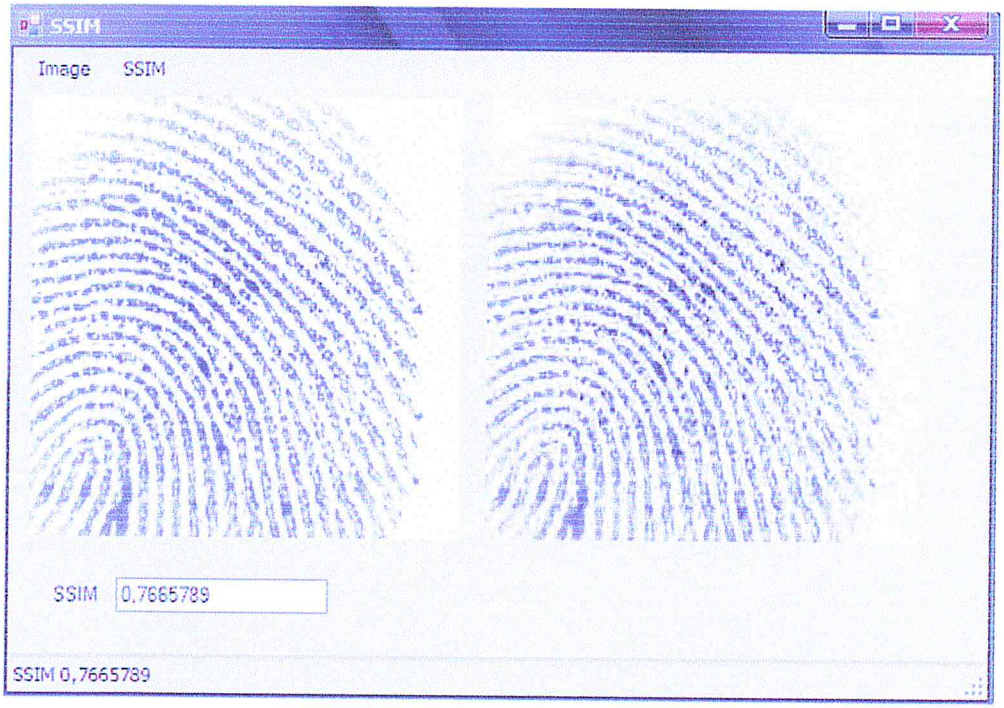

Figure IV.3 calcul le SSIM.

Deux (02) choix des images biométriques à utiliser (visage ou empreinte digitale). Pour les deux images, les mêmes méthodes correspondantes sont associées à la barre de menu, se qui permet de spécifier au préalable (figure IV.4).

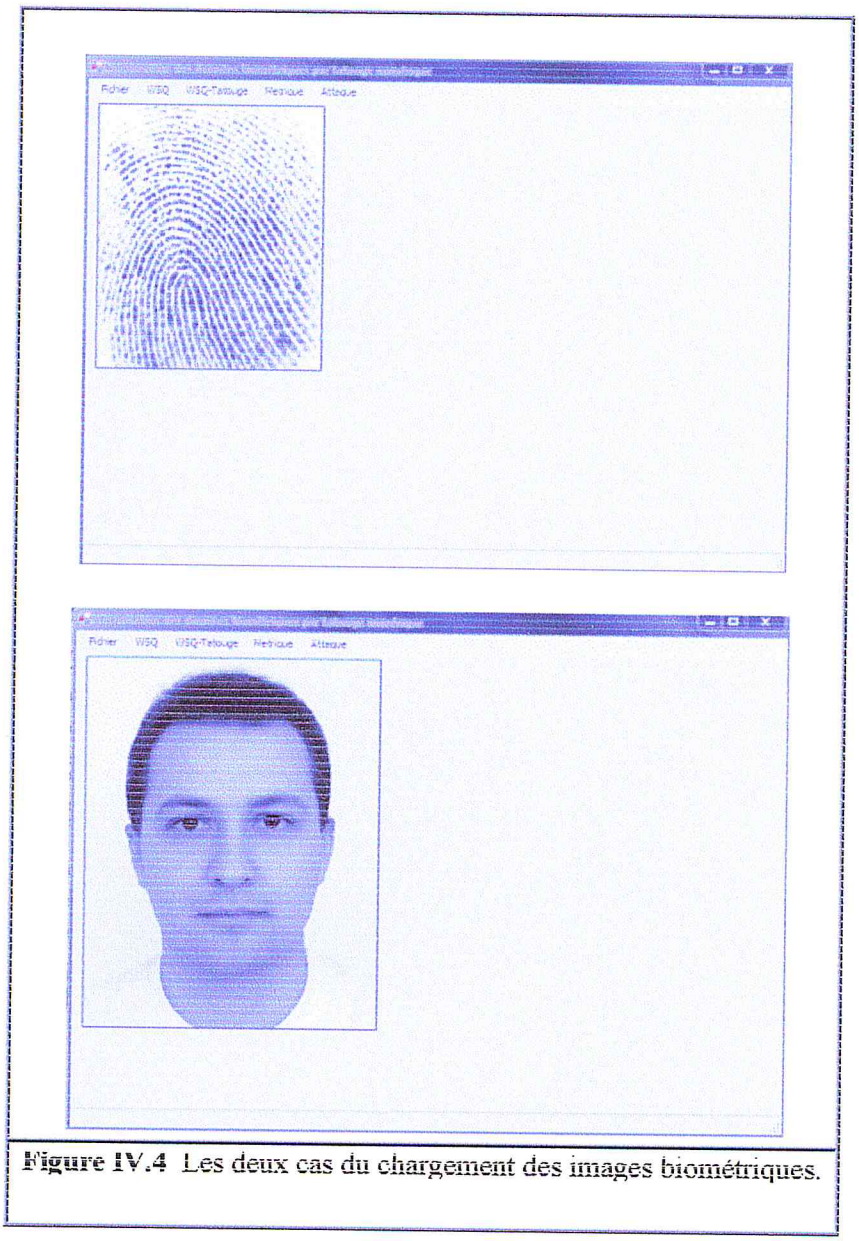

Figure IV.4 Les deux cas du chargement des images biométriques.

Pour l'insertion, l'utilisateur clique sur le bouton « Ouvrir » du sous-menu « fichier » pour sélectionner l'image à charger. Après, l'utilisateur choisit d'effectuer l'insertion en utilisant la méthode de compression WSQ, une fenêtre correspondante apparue (figure IV.5) contenant l'image d'origine, la marque à insérer, et la clé d'embrouillage. Enfin, en cliquant sur le bouton « insertion », l'image est compressée et marquée. Tous ces résultats peuvent être enregistrés dans un fichier de format (.wsq).

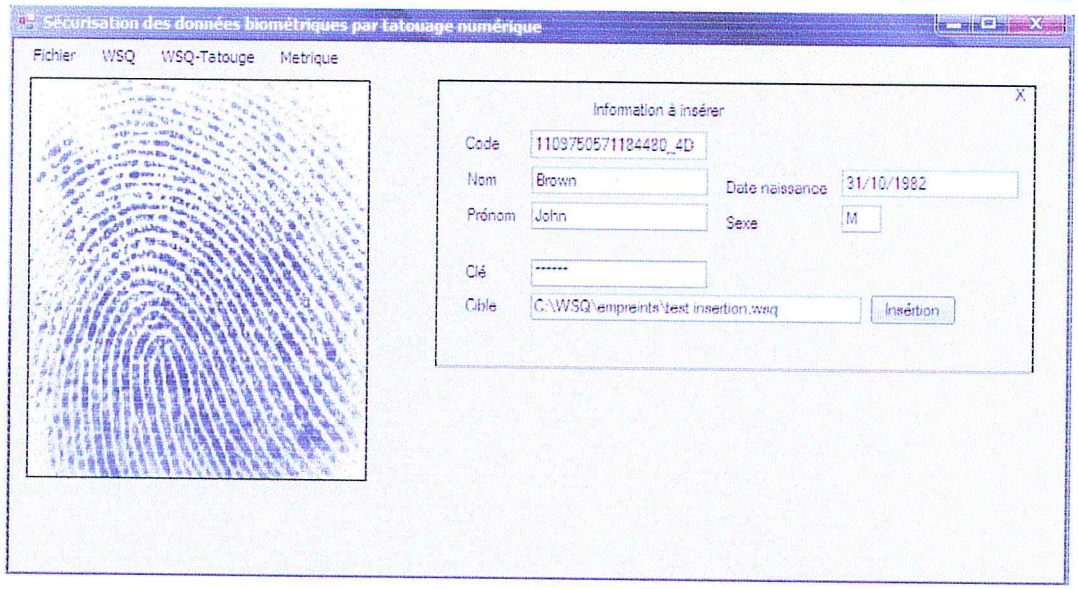

Figure IV.5 Insertion de la marque.

Pour l'extraction, l'utilisateur clique sur le bouton « Ouvrir » du sous-menu « fichier » pour charger le fichier marqué de format (.wsq). Après, si l'utilisateur choisit d'effectuer l'extraction, une fenêtre correspondante est apparue (figure IV.6) contenant le fichier marqué au préalable et la zone de texte de la clé. Pour extraire la marque, l'utilisateur saisie la clé d'insertion et clique sur le bouton « extraction ». Enfin, une fenêtre correspondante est apparue (figure IV.7) contenant la marque extraite et l'image reconstruit.

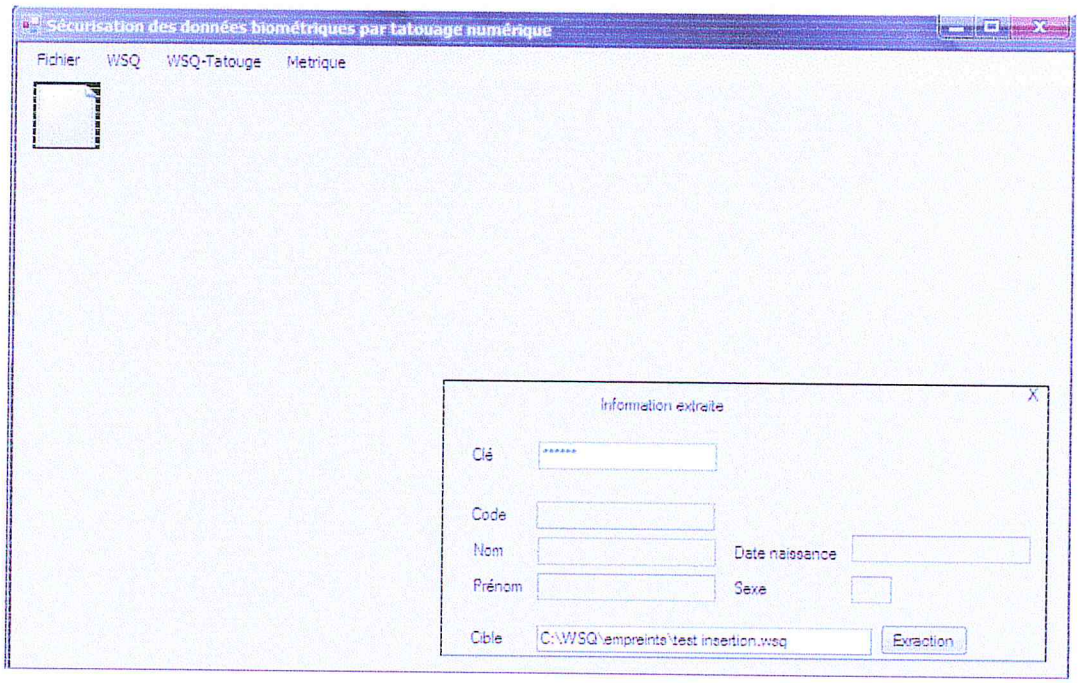

Figure IV.6 Extraction de la marque.

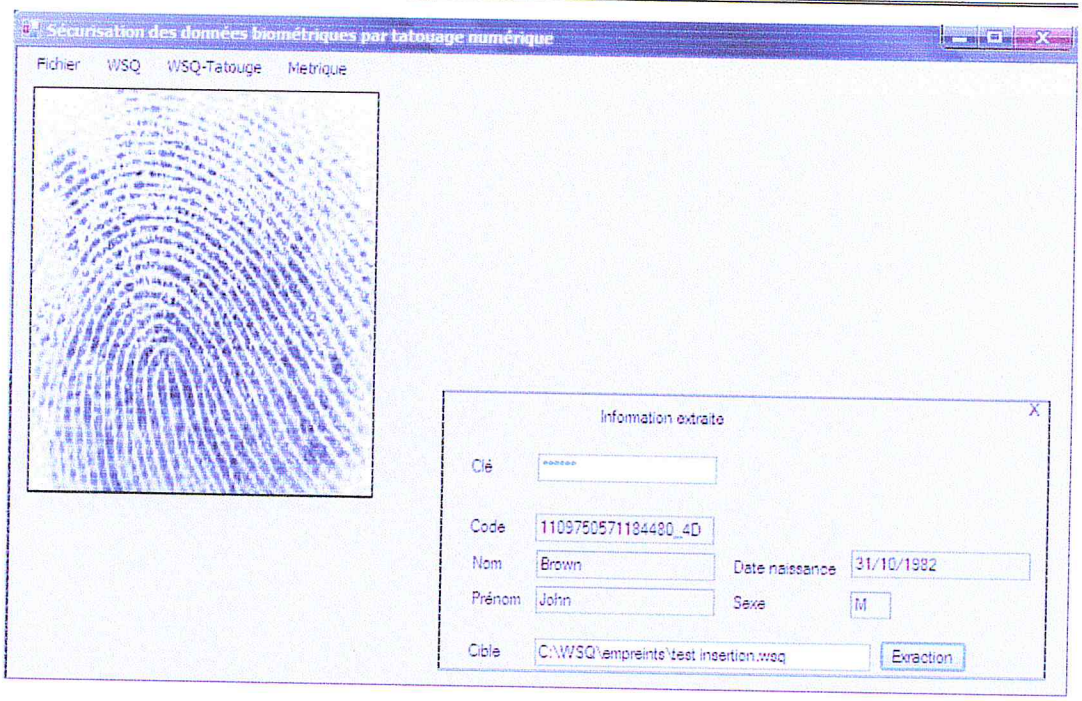

Figure IV.7 Résultat d'extraction de la marque.

Ainsi, on peut aussi calculer le BER pour analyser les performances d'extraction de la marque, ce dernière est affichée dans une fenêtre (figure IV.8).

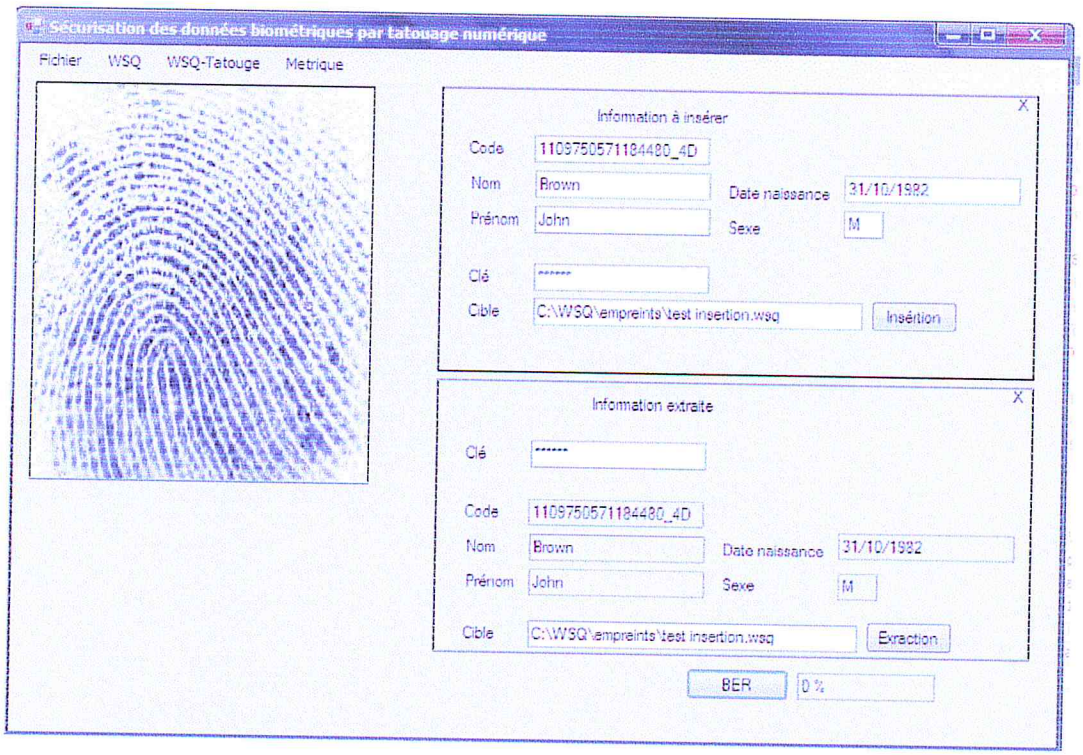

Figure IV.8 Calcule le BER.

IV.4 Conclusion

Dans ce chapitre nous avons exposé nos différents tests et expérimentations effectués ainsi que les résultats obtenus Les tests ont été effectués sur des images biométriques réelles. L'approche de sécurisation de ces images au cours de leur compression par le standard WSQ a prouvé son efficacité en termes d'imperceptibilité et de l'intégrité de la marque insérée.

Aussi dans ce chapitre, nous avons décrit l'interface de l'application avec toutes les fonctionnalités afin de permettre à un opérateur d'utiliser l'application aisément.

Conclusion générale

Ce projet vise la protection des codes des images biométriques contre toute tentative de modification. Car, ce sont ces codes des images qui font les liens entre les individus et leurs images biométriques. Pour y parvenir, nous avons proposé une méthode basée sur le tatouage numérique, pour insérer les codes dans les images elles mêmes, comme ça, les codes sont plus sécurisés. De plus, cette méthode a été adaptée pour éviter l'influence de la compression WSQ sur les informations insérées.

Le but attendu de ce travail est de développer les différents modules d'un système de tatouage numérique visant la protection de l'intégrité des images biométriques. En faite, les modules réalisés seront intégrés dans le projet « la sécurisation des systèmes biométriques multimodal », initié au niveau de la gendarmerie nationale.

Pour ce faire, nous avons mené une étude théorique sur les avantages apportés par chaque domaine d'insertion, après quoi nous avons opté pour un système de tatouage numérique aveugle ne nécessitant pas la donnée d'origine lors de l'extraction de la marque. Cette marque est de type fragile et composée de plusieurs bits insérés dans le domaine compressé, précisément dans les codes space de la table VLC en utilisant la règle substitutif comme règle d'insertion.

Pour l'insertion de la marque, nous avons choisi qu'elle soit effectuée dans l'étape de codage entropique du standard de compression WSQ. En utilisant une technique basée sur l'approche de code de longueur variable (Variable Length Code VLC) que nous avons proposé.

Quelques tests sont été réalisés afin d'évaluer les performances des techniques implémentés en terme de d'imperceptibilité de la marque (analyser le niveau de dégradation de la qualité visuelle de l'image après l'insertion de la marque) et la performance de l'extraction de la marque (analyser les erreurs d'extraction).

Les résultats obtenus ont permis, d'une part de valider notre choix de faire l'insertion au cours de la compression, et d'autre part de montrer que les performances visées telles que l'imperceptibilité et la sécurité de la marque, sont bien atteintes.

Toutefois, ce système peut être amélioré en envisageant les perspectives suivantes :

- Insérer les minuties d'empreintes digitales comme marque dans l'image de visage.
- Schéma bimodale, c'est-à-dire insérer la marque et l'empreinte dans la TOD du WSQ.

Bibliographie

[1] TABOUCHE Abdelkader et BOUMAZA Adel, "Sécurisation des images d'empreintes digitales par le tatouage numérique," Mémoire d'ingénieur d'état en Génie Informatique. EMP 2010.

[2] A. Parisis, P. Carré and A. Trémeau, " Tatouage d'images couleur," Thèse de Doctorat, Laboratoire SIC, Université de Poitiers, 2004.

[3] Cox et Morgan Kaufmann, "Digital watermarking and steganography," deuxieme edition, 2007.

[4] F.A.P Peticolas, R.J. Anderson et M.G. Kuhn, "Information Hiding- A Survey, " Proceeding of the IEEE, special issue on Protection of Multimedia Content, vol. 87, pp. 1062- 1078, July 1999.

[5] Vincent Martin, " Contribution des filtres LPTV et des techniques d'interpolation au tatouage numérique," thèse de doctorat, École doctorale : Informatique et Télécommunications, 28 novembre 2006.

[6] Yann bodo, " Elaboration d'une technique d'accès conditionnel par tatouage et embrouillage vidéo basée sur la perturbation des vecteurs de mouvement," thèse de doctorat, Ecole National Supérieur des Télécommunications, 09 septembre 2004.

[7] G.C. Langelaar et al. "Watermarking Digital Image and Video Data," *IEEE Signal Processing Magazine*, vol. 17, pp. 20-46, Sept. 2000.

[8] E.Khelifi, "Image Compression and Watermarking in the Walevet Transform Domain," Phd, Queen's University of Belfast, UK, Mai 2007.

[9] J. Murillo-Fuentes, "Independent component analysis in the blind watermarking of digital images," Neurocomputing, vol 70, pp 2881–2890, 2007.

[10] N. Ratha, J. H. Conell et R.M. Bolle, "Enhancing Security and Privacy in Biometrics-Based Authentication Systems," IBM Systems Journal, vol. 40, pp. 614-634, 2001.

[11] P. Bas, " Méthodes de tatouages d'images fondées sur le contenu," Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2000.

[12] F. Raynal, " Etudes d'outils pour la dissimulation d'information: approches fractales, protocoles d'évaluation et protocoles cryptographiques," thèse de doctorat, Université Paris XI, mars 2002.

[13] V. Duc Minh, N. Thi Hoang et L. Maiti, " Tatouage des images dans un domaine fréquentiel," Dans Lecture Notes in Computer Science, Information Hiding, vol. 11, pp 6-14, janvier 2006.

[14] Nafornita Corina, "Filigranage dans le domaine des ondelettes," Mémoire de diplôme pour obtenir le degré de M.Sc, L'université "Politehnica" Timisoara, Faculté d'Electronique et Télécommunications, 2004.

[15] BELMOULOUD Hichem et BENKACI Sofiane " Tatouage d'images médicales en vue d'une transmission de données," Mémoire d'ingénieur d'état en Electronique, Faculté d'Electronique et d'Informatique, Département Télécommunications, USTHB, Algérie, 2005.

[16] Khalil Zebbiche, «Data hidding for securing fingerprint data access," thèse Phd, Queen's University of Belfast, juillet 2008.

[17] Guekhakhma Said, "Authentification du contenu H.264/AVC le tatouage numérique et la signature numérique," Mémoire pour l'obtention D'un diplôme d'ingénieur d'état en informatique, Option : Système d'informatique (SI), juillet 2009.

[18] Z. Ni, Y. Shi, N. Ansari, and W. Su, "The Security Portal for Information System Security Professionals," Proc. ISCAS 2003, vol. 2, pp. 912–915.

[19] Ming Sun Fu and Oscar C. Au, " Data hiding by smart pair toggling for halftone images," Proc. of IEEE Int. Conf On Acoustics, Speech and Signal Processing, vol. 4, pp. 2318–2321, June 2000.

[20] (CJISD) FBI, " WSQ Gray-Scale Fingerprint Image Compression Spécification," Version 3.1, Octobre 2010.

[21] CHERKI Mohamed et ZIOUCHE Merwan "Application du tatouage numérique aux images compressées par le standard WSQ" Mémoire d'ingénieur d'état en Génie Informatique. EMP 2011.

[22] B. Mobasseri, R. Berger, "A Foundation for Watermarking in Compressed Domain," IEEE Signal Processing Letters, pp. 399-402, May 2005.

[23] B. Liu, F. Liu, B. Lu, and al, "Real-Time Steganography in Compressed Video," Information Engineering Institute, The Information Engineering University, Zhengzhou Henan Province, 450002, China.

[24] Michael P. Marcinak, Bijan G. Mobasseri, "Metadata Embedding in Compressed UAV Video," Intelligent Ship Symposium, Philadelphia, May 2003.