République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la recherche scientifique

Université Mentouri de Constantine
Faculté des sciences de l'ingénieur

Département de l'informatique

Thèse de doctorat en sciences en Informatique

Thème :

Contribution à l'authentification souple d'images digitales par des techniques de marquage numérique

Application aux images médicales

Présentée par :
Mme Chikhi Samia née Boucherkha

Dirigée par :
Pr. Benmohammed Mohamed

Devant le jury composé de :

Prof. M. Boufaida	Université de Constantine	Président
Prof. M. Benmohammed	Université de Constantine	Rapporteur
Prof. M.T. Laskri	Université de Annaba	Examinateur
Prof. N. Djedi	Université de Biskra	Examinateur
Prof. M. Benslama	Université de Constantine	Examinateur

Octobre 2008

Résumé

La croissance exponentielle du trafic des images digitales sur les réseaux soulève un nombre conséquent de problèmes de droits d'auteur, mais également d'authenticité des images échangées. Le marquage numérique est une nouvelle technologie avancée pour résoudre ce type de problèmes. Contrairement aux techniques classiquement employées en cryptographie pour assurer la fonction d'authentification, les nouvelles méthodes de marquage proposées privilégient une intégrité en termes de contenu sémantique à une intégrité numérique stricte. Une limite à l'utilisation de l'authentification par marquage est la distorsion infligée à l'image par le processus d'insertion qui peut ne pas être acceptable dans certaines applications sensibles comme le domaine médical. Les schémas qualifiés de réversibles sont les plus appropriés dans ce cas. Une autre fonctionnalité très souhaitable est la capacité de localisation des manipulations.

Ce travail est consacré à l'authentification souple des images numériques, avec comme objectif d'application le contrôle d'intégrité dans des applications de télémédecine.

Dans ce but, nous définissons un modèle de signature numérique fondé sur les attributs texturaux de l'image, qui est capable de détecter les altérations portant sur la texture et d'ignorer les transformations géométriques bénignes, tout en maintenant une bonne qualité visuelle de l'image. A travers une étude sur les méthodes d'analyse d'image par texture, nous avons pu constater la robustesse de certaines d'entre elles aux transformations géométriques. Parmi ces méthodes, la cooccurrence se démarque par de très bons résultats mais présente l'inconvénient de temps de calcul assez importants. Nous utilisons les statistiques de premier et second ordre pour établir un ensemble de descripteurs sur lesquels nous appliquons différents traitements pour les adapter à nos besoins spécifiques en « semi-fragilité » et réduire les temps de calculs. Pour assurer la réversibilité de notre modèle d'authentification, nous utilisons un algorithme stéganographique reposant sur l'expansion de la différence des pixels adjacents. Pour augmenter sa capacité tout en évitant le problème d'overflow caractérisant la méthode originale, nous proposons une nouvelle manière de codage des coefficients d'expansion. L'insertion/extraction des données est réalisée bloc par bloc pour permettre la localisation des régions altérées. Nous procédons à l'évaluation du pouvoir discriminant de l'approche et de sa capacité de localisation d'attaques en effectuant le contrôle d'intégrité sur un ensemble d'images test dont des images médicales.

Nous terminons en proposant deux solutions de sécurité globale pour les images médicales, combinant compression, chiffrement et marquage pour assurer l'authentification de l'image en même temps que la confidentialité des données patient qui l'accompagnent., en les insérant à leur tour dans l'image. Pour mettre en évidence l'utilité de telles solutions nous exposons un scénario d'utilisation à travers une application de partage d'images médicales sur le Net, entre une communauté limitée d'utilisateurs.

Mots-clés : Authentification, Intégrité, Fonctions de hachage, Marquage numérique, Texture, Statistiques de second ordre, Traitement d'images, Imagerie médicale.

Remerciements

En premier lieu je tiens à remercier mon directeur de thèse, le Professeur M. Benmohamed pour la confiance qu'il m'a accordée, sa gentillesse sans égal, ainsi que la liberté de mouvement qu'il m'a toujours concédée.

Je remercie chaleureusement les membres de mon jury d'avoir bien voulu consacrer du temps et de l'attention à mon travail, à commencer par

- Monsieur M. Boufaida, Professeur à l'Université de Constantine, pour l'honneur qu'il me fait de présider le jury de cette thèse.

- Ma profonde gratitude envers Monsieur M. T Laskri, Professeur et Recteur à l'Université de Annaba, qui malgré ses taches colossales, particulièrement en cette fin d'année, a accepté de regarder de prés mes travaux.

- Monsieur N. Djedi, Professeur à l'Université de Biskra et Monsieur M. Benslama, Professeur à l'Université de Constantine qui ont gentiment accepté d'expertiser mes travaux.

Bien évidemment je remercie mon mari pour sa confiance et son soutien sans faille ainsi que mes enfants pour leur patience au cours de toutes ces années.

Enfin, je salue l'ensemble de mes étudiants qui ont participé peu ou prou à certaines des expérimentations de ce travail durant leurs projets de fin d'études. Je pense particulièrement à Zakaria, Saoussen, Bahidja, et Latifa.

- Es-tu certain de chercher la clé au bon endroit?
- Non, mais c'est le seul lieu éclairé.

Proverbe arabe

Table des matières

Remerciements…… i
Résumé…..…… ii
Liste des figures……...……...… iii
Liste des tableaux……..……..……...… v
Introduction générale……..……..……..……..……... 1
I. Position du problème……. 1
II. Contributions……... 2
III. Organisation de la thèse ..…….... 4
Chapitre 1 : La protection de documents numériques..…… 6
I. Sécurité de documents numériques…….. 7
I.1 La confidentialité…… 7
I.2 L'intégrité……. 8
I.3 L'authentification……... 8
I.4 La non-répudiation……. 8
II. Les mécanismes de protection…….. 9
II.1 La cryptographie……. 9
II.2 La dissimulation de données…….... 9
III. Comparatif des différentes techniques……... 11
III.1 Cryptographie vs stéganographie…….. 11
III.2 Stéganographie vs marquage numérique…… 11
Chapitre 2 : Outils cryptographiques..……..……... 13
I. Définitions……... 14
II. Le chiffrement……... 15
II.1 Chiffrement par substitution……..…… 15
II.1.1 Substitution simple ou substitution monoalphabétique…….…. 15
II.1.2 Substitution homophonique ……..…... 15
II.1.3 Substitution polyalphabétique ……..…... 16
II.1.4 Substitution par polygrammes ……..…... 16
II.2 Chiffrement par transposition……..… 17
II.2.1 Transposition simple par colonne…… 17
II.2.2 Transposition complexe par colonnes ……. 17
II.2.3 Transposition par carré polybique …… 17
II.3 Systèmes symétriques ou à clef secrète……... 17
II.3.1 les algorithmes de chiffrement en continu……. 17
II.3.2 les algorithmes de chiffrement par blocs……... 17
II.3.3 Chiffrement par blocs avec itération……. 18
II.4 Systèmes asymétriques ou à clef publique…….. 20
III. Authentification et contrôle d'intégrité……... 22
III.1 Les fonctions de hachage……..... 22
III.2 La signature numérique…….. 23

III.3 Les codes d'authentification de message ou MAC..... 24
IV. Accord sur les clefs et authentification mutuelle..... 25
V. Générateurs aléatoires et pseudo-aléatoires..... 26
VI. La cryptanalyse..... 27
VI.1 Objectifs de la cryptanalyse..... 27
VI.2 Les attaques..... 28
VI.3 Fiabilité des systèmes cryptographiques..... 28
VI.3.1 Problème des clefs..... 28
VI.3.2 Principe de Kerckhoffs..... 29
VI.3.3 Sécurité inconditionnelle..... 30

Chapitre 3 : La dissimulation de données..... 32

I. La stéganographie..... 33
I.1 Définition introductive..... 33
I.2 Mode d'opération..... 33
I.3 Les applications de la stéganographie..... 34
I.4 Mise en œuvre..... 35
I.4.1. Message transporté dans un texte..... 35
I.4.2 Message transporté dans une image..... 35
I.4.3 Message transporté dans du son..... 37
I.5 La stéganalyse..... 37
II- Le marquage numérique..... 38
II.1 Définition introductive..... 38
II.2 Mode de fonctionnement..... 40
II.3 Applications du marquage..... 41
II. 4 Caractérisation du schéma de marquage..... 46
II .5 Quelques algorithmes de marquage..... 48
II.5.1 Domaine spatial..... 49
II.5.2 Domaine fréquentiel..... 50
II.5.3 Autres approches..... 50
II.6 Les Attaques sur le marquage numérique..... 51
II.6.1 Les attaques basiques involontaires..... 52
II.6.1.1 Les transformations géométriques..... 52
II.6.1.2 Les transformations fréquentielles..... 54
II.6.2 Les attaques volontaires..... 55
II.6.3 Les attaques de nature cryptologique..... 56

Chapitre 4 : Etat de l'art sur l'authentification des images par marquage numérique..... 58

I. Notions d'intégrité..... 59
II. Exemples classiques de manipulations malveillantes..... 60
III Caractéristiques d'un système d'authentification d'image..... 61
IV Revue des méthodes existantes..... 63
IV.1 Marquage fragile..... 63
IV.1.1. Insertion dans le domaine spatial..... 64
IV.1.2. Insertion dans le domaine transformé..... 68
IV.2 Marquage semi-fragile..... 68
IV.2.1 Exemple de méthode transparente à la compression Jpeg..... 69
IV.2.2 Marquage par région..... 70

IV.2.3 Les ondelettes…… 72
IV.2.4 Marquage de caractéristiques de l'image ou basé sur le contenu…….... 72
IV.3 Marquage réversible…… 74
IV.3.1 Les schémas basés compression……... 75
IV.3.2 Les schémas utilisant l'expansion de la différence…… 78
IV.3.3 Les schémas utilisant le décalage d'histogramme……..…... 81
V. Les attaques contre les systèmes d'authentification ……..... 83

Chapitre 5 : Extraction de signature par analyse de texture…… 87

I. L'analyse d'image par texture…… 89
II. Extraction d'attributs de premier ordre……..…… 91
III. Extraction d'attributs de second ordre……..…… 94
IV. Principe de notre approche……..…… 96
IV.1 Etape de prétraitement……. 96
IV.2 Génération de la signature…….. 98
IV.3 Vérification de la signature et de localisation d'attaques…….. 100
IV.4 Insertion par la méthode de Wu et Tsai…….. 100
IV.5 Insertion par la méthode de Tian…… 101
V. Calcul de distance……..... 102
VI. Critères de qualité et mesure de distorsion d'une image…… 103

Chapitre 6 : Protection des images médicales…….. 108

I. Les différentes modalités d'imagerie médicale……..… 110
II. CC-MARK: Un système de compression-chiffrement-marquage d'images……. 113
II.1 Méthode de compression……... 114
II.2 Méthode de chiffrement……..… 115
II.3 Méthode de marquage……... 116
II.4 Processus combiné…… 116
III. Un système d'authentification à clé secrète basé compression…..……..… 119
IV. MEDIMAGE : Un service web pour le partage sécurisé d'images médicales……. 124
IV.1 Fonctionnalités du système…….. 125
IV.2 Règles d'utilisation…….. 127
IV.3 Méthode de conception…… 128
IV.4 Réalisation du service…….. 132

Conclusion générale…….. 135

Références bibliographiques……. 138

Liste des figures

2.1	Schéma générique de la cryptologie……..	14
2.2	Le mode CBC……	18
2.3	Chiffre de Feistel à 2 rondes……..	19
2.4	Exemple de chiffrement avec RSA……	21
2.5	Fonction de hachage itérative……	23
2.6	Obtention d'une signature numérique……	24
2.7	MAC obtenu à l'aide d'un algorithme de chiffrement symétrique…….	25
3.1	Schéma générique de la stéganographie……	34
3.2	Interaction des contraintes de marquage……	39
3.3	Schéma générique du marquage numérique……..	40
3.4	L'image Lena…….	42
3.5	Exemple de marquage visible……	46
3.6	Exemple de symétrie horizontale……...	52
3.7	Exemple de découpage simple……..	53
3.8	Exemple de mise en page : Rotation (7°), Mise à l'échelle (120%) et découpage…….	53
3.9	Exemple de mosaïque d'image……..	54
3.10	Exemple de bruitage d'une image……..	54
3.11	Exemple de filtrage linéaire……...	55
3.12	Exemple de perte lors de compression JPEG……	55
4.1	Schéma générique d'un système d'authentification……..	62
4.2	Mécanisme d'insertion par la méthode Yeung et Mintzer…….	65
4.3	Mécanisme d'insertion par la méthode de Wong……..	66
4.4	Fonctionnement de la méthode de Wong……...	66
4.5	Mécanisme d'insertion dans la méthode de Kundur……..	68
4.6	Marquage de caractéristiques…….	73
4.7	Schéma bloc du marquage réversible……	74
4.8	Marquage réversible basé compression…….	75
4.9	Illustrations du schéma de Celik et al……	76
4.10	Insertion par le schéma de Tian…….	79
4.11	Schéma bloc de l'algorithme de Wu et Tsai…….	80
4.12	Insertion par l'algorithme de Vleeschouwer…….	82
5.1	Illustration de différentes textures……	89
5.2	Illustration de la moyenne…….	91
5.3	Illustration de la variance……...	92
5.4	Illustration de l'écart type……..	92
5.5	Illustration du skewness…….	93
5.6	Illustration du kurtosis……...	94
5.7	Exemple d'une image à 4 niveaux de gris et ses 4 GLCMs……..	94
5.8	Exemple de normalisation de la matrice……	97
5.9	Processus de génération de la signature…….	99
5.10	Processus de vérification de la signature et de localisation d'attaques..	100

5.11	Ordre de traitement des blocs……	101
5.12	Images originales et images marquées respectivement par Wu et Tsai et Tian……	103
5.13	Histogrammes de Lena et Baboon (a) avant marquage, (b) marquage Wu et Tsai, (c) marquage Tian…...	104
5.14	Variations du PSNR pour des images marquées par (a) Wu et Tsai, (b) Tian…….	104
5.15	(a) Image originale, (b) Image marquée puis attaquée (c) Image vérifiée……	106
5.16	Variations du TFA et TFR pour les deux ordres de statistiques……	106
6.1	Images échantillons originales……..	117
6.2	Résultats sur les images échantillons traitées par (a) l'algorithme séquentiel, (b) l'algorithme imbriqué, (c) images décodées…….	117
6.3	Histogrammes des 3 images avant et après codage……...	118
6.4	Variations du temps en fonction de la taille des images……	119
6.5	Schéma de formation et d'insertion de la marque…….	120
6.6	Schéma d'extraction et de vérification de la marque…….	121
6.7	Impact du marquage sur trois images test en termes de PSNR…….	122
6.8	Exemple d'image IRM (512*512 pixels) avec pas d'insertion 1 pixel.	122
6.9	Exemple d'Image IRM (256*256 pixels) avec pas d'insertion 2 pixels	122
6.10	Exemple d'image Rayon X (121*104 pixels) avec pas d'insertion 1 pixel……	123
6.11	Exemple d'image d'échographie (800*600 pixels) avec pas d'insertion 25 pixels……...	123
6.12	Echantillon d'images a : originale et marquée b : originale, marquée puis attaquée……..	124
6.13	Architecture du système…….	127
6.14	Diagrammes des cas d'utilisation……	129
6.15	Diagramme des classes participantes de "Enregistrement"……...	130
6.16	Diagramme des classes participantes de "Consulter Image"…….	130
6.17	Diagramme de classes participantes de "Supprimer marque"……	131
6.18	Diagramme de séquence de " Enregistrement"……..	131
6.19	Diagramme de séquence de "Consulter image"……..	132
6.20	Consultation de la base……	133
6.21	Vérification d'une image de la base……..	133
6.22	Exemples de formulaires…….	134

Liste des tableaux

4.1	Table des différences……	80
5.1	Echantillons d'attributs de premier ordre normalisés……	104
5.2	Distance de Hamming entre vecteurs originaux et vecteurs attaqués (1^{er} ordre)……..	105
5.3	Echantillons d'attributs de second ordre normalisés……	105
5.4	Distance de Hamming entre vecteurs originaux et vecteurs attaqués (2^{e} ordre)……....	105
6.1	Principales modalités d'imagerie médicale……	112
6.2	Phases de chiffrement d'un bloc de 3 pixels……	115
6.3	Consommation en temps pour les deux algorithmes (en secondes)…….	118
6.4	Mesures de l'entropie (en bits)…….	119
6.5	Format des données patient……..	121
6.6	Vérification de l'intégrité en comparant l'empreinte calculée avec l'empreinte extraite……	124

Introduction Générale

I. Position du problème

La révolution numérique a engendré des moyens plus faciles pour le traitement, le stockage et la transmission des images numériques. Cependant, elle a aussi engendré des moyens de falsification beaucoup plus simples qu'avec le format analogique traditionnel. Avec le développement rapide des techniques de traitement d'images et la disponibilité actuelle de puissants outils de retouche tels que MS-Photoshop, il est devenu possible de réaliser des contrefaçons très élaborées, sans laisser aucune trace. Le risque est encore plus grand dans un environnement ouvert tel que l'Internet grâce auquel on peut télécharger, modifier puis redistribuer les images numériques à volonté.

Dans ces circonstances, il est devenu nécessaire d'élaborer des outils adaptés à ces nouvelles menaces. En particulier, la vérification de l'intégrité et de l'authenticité des images numériques revêt un aspect très important dans toute application mettant en oeuvre la publication ou l'échange des images numériques. Le développement de ces techniques sécuritaires aidera sans doute des domaines comme le e-commerce ou la télémédecine à se débarrasser de leur timidité actuelle.

La méthode traditionnelle utilisée pour l'authentification des données est le chiffrement, qui rend inintelligibles les données transitant sur les réseaux. Un concept plus récent est la signature numérique (DS) qui consiste en un condensé de l'image chiffré avec la clé privée d'un crypto-système à clé publique. Une alternative à la signature numérique est le code d'authentification de message (MAC) où le condensé est chiffré grâce à la cryptographie à clé secrète. En général, le DS/MAC est concaténé à l'image dans un fichier séparé, augmentant ainsi les besoins en largeur de bande. Pour éviter cette surcharge et pour d'autres besoins que nous identifierons plus tard, la technologie du marquage numérique ou watermarking a été récemment avancée. Cette dernière repose sur l'insertion d'une petite quantité d'information à l'intérieur de l'image même, de manière secrète, sans dégrader sa qualité visuelle. Ces quatre technologies peuvent être utilisées en complémentarité et pourront alors composer un solide arsenal pour protéger les images numériques publiées sur le web, si judicieusement agencées, comme nous le montrerons dans une de nos contributions.

La définition cryptographique de l'intégrité repose sur une décision binaire qui garantit que les données reçues sont rigoureusement identiques à celles émises. Cette définition est en principe applicable à tout type de documents numériques, néanmoins, dans la pratique elle s'avère être beaucoup trop stricte et inadaptée pour les documents de type images. En effet, l'interprétation que l'on a d'une image dépend principalement des éléments la constituant plutôt que des valeurs numériques des pixels ou de sa résolution. En d'autres termes, le problème de l'intégrité des images se pose en termes de contenu sémantique plutôt qu'en valeur des pixels la constituant. Il s'agit de pouvoir détecter les modifications de l'image pouvant engendrer une gêne dans sa visualisation et/ou une erreur dans son interprétation. Dans le but d'assurer un service d'intégrité approprié aux images, il est donc primordial de distinguer les manipulations malveillantes consistant à détourner le contenu initial de l'image, des manipulations liées à son utilisation ou son stockage sous une forme numérique. Cette authentification est qualifiée de « souple ou soft », par rapport à celle assurée par les mécanismes cryptographiques classiques, qui elle est qualifiée de « exacte ou hard ». Les techniques de marquage qui privilégient une intégrité en termes de contenu à une intégrité numérique stricte sont dites « basées sur le contenu ».

Une limite évidente à l'utilisation de l'authentification par marquage numérique est la distorsion infligée à l'image hôte par le processus d'insertion. Même si cette distorsion est souvent minime, elle peut ne pas être acceptable dans certaines applications sensibles. Par exemple, le domaine militaire où l'authentification est utilisée dans la surveillance, le guidage automatique et la poursuite d'engins ne souffre d'aucune modification. De même, le domaine médical, où toute distorsion de l'image peut avoir de sévères conséquences, puisque pouvant fausser un diagnostic. Il est donc souhaitable de disposer de schémas d'authentification capables de supprimer toute distorsion de l'image après une vérification positive de la marque. Les schémas offrant cette possibilité sont qualifiés de réversibles (ou inversibles).

Notre but dans cette étude est de présenter une méthodologie d'authentification souple des images numériques sous la contrainte de réversibilité, avec comme objectif d'application l'imagerie médicale. Dans ce but, nous définissons un modèle de signature fondé sur les attributs texturaux de l'image. Nous tenterons de l'appliquer au domaine médical, tout en proposant des solutions de sécurité globales pour les images médicales fondées sur une combinaison de diverses techniques cryptographiques et stéganographiques.

II. Contributions

Une première contribution consiste donc à définir un modèle de signature numérique pour l'authentification fondée sur les attributs texturaux de l'image. Nous utilisons les statistiques de premier et second ordre pour établir un ensemble de descripteurs de texture sur lesquels nous appliquons différents traitements pour les adapter à nos besoins spécifiques en « semi-fragilité ». Pour assurer la réversibilité de notre modèle d'authentification, nous utilisons un

algorithme stéganographique reposant sur l'expansion de la différence. Pour augmenter sa capacité tout en évitant les overflows caractérisant la méthode originale, nous proposons une nouvelle manière de codage des coefficients d'expansion, ce qui constitue en soi une autre contribution. Nous procédons à l'évaluation du pouvoir discriminant des différentes approches d'analyse de texture et de la capacité de localisation d'attaques en effectuant le contrôle d'intégrité sur un ensemble d'images test dont des images médicales.

L'image médicale est rarement transmise seule sur un réseau. Elle doit souvent être mise en perspective avec les autres éléments plus conventionnels du dossier médical : L'historique du malade, ses antécédents, sa pathologie ainsi que certaines informations administratives. Ce mode de circulation d'informations soulève de sérieuses questions de sécurité relatives aux dossiers médicaux, particulièrement face aux exigences des aspects éthique et légal propres au domaine. Les informations médicales doivent être rendues illisibles donc chiffrées avant d'être transférées. De plus, pendant le transfert des données, il ne faut absolument pas qu'une image soit dissociée du nom du patient concerné pour éviter toute confusion d'appartenance à la réception de celle-ci. D'autre part, les images médicales numérisées, posent par leur taille importante, de nombreux problèmes quant à leur transmission ou à leur stockage. Pour gagner aussi bien en vitesse qu'en place, il est presque toujours nécessaire de faire une compression de l'image pour pouvoir l'utiliser dans une application quelconque de télémédecine.

Une autre contribution consiste donc à proposer une solution de sécurité globale pour les images médicales, fournissant l'authentification de l'image en question, tout en assurant la confidentialité des données patient qui l'accompagnent en les insérant sous une forme chiffrée au sein de l'image en même temps que l'information d'authentification (l'empreinte). Ceci permettra en plus de faire transiter l'image seule, sans avoir besoin de l'accompagner d'un fichier textuel. Pour ceci, un algorithme à faible coût est proposé, combinant les techniques de compression, chiffrement et marquage dans un seul et même algorithme réalisant les 3 taches en une seule passe : c'est l'algorithme "CC-MARK". Nous démontrerons qu'il est plus avantageux d'utiliser cet algorithme à travers de nombreux résultats expérimentaux.

Nous discuterons finalement la possibilité de développer un système d'authentification d'images destiné à protéger les images médicales publiées sur le Net. Nous présenterons une architecture possible d'un tel système permettant à une communauté déterminée d'utilisateurs de partager en toute sécurité une collection d'images médicales par le biais d'un service web purement académique. Les fonctionnalités du système seront décrites à travers un scénario simplifié mettant en œuvre une combinaison des techniques précitées, en tant que mécanismes interactifs et complémentaires. Il peut fournir le contrôle nécessaire au propriétaire de l'image, et la garantie aux clients qui désirent s'assurer de l'authenticité des images obtenues via Internet.

Beaucoup des techniques étudiées à travers l'état de l'art, et bien que parfois très élaborées, ne constituent pas des solutions viables dans des systèmes réels, à cause de leur difficulté de mise en oeuvre et leur consommation en temps et espace. Dans toutes les propositions de cette thèse on privilégiera donc les directions les plus simples, tendant à minimiser les temps de calculs.
Enfin, les solutions de sécurité proposées reposent uniquement sur des outils logiciels**.**

III. Organisation de la thèse

Dans cette thèse, nous aborderons les aspects de l'authentification des images numériques et en particulier, celle des images médicales. Plus précisément, cette thèse est composée des chapitres suivants :

– chapitre 1 : Nous y présentons la problématique de la protection des documents numériques. Les différents aspects de sécurité impliqués dans cette protection seront identifiés et les différents outils permettant de les assurer seront énumérés, pour être finalement comparés sur la base de plusieurs paramètres.

– chapitre 2 : Les principes de base de la cryptographie y sont énoncés. Nous détaillerons particulièrement certains concepts utiles pour les méthodes de marquage décrites par la suite, ou qui nous serons utiles dans nos contributions, comme le chiffrement, les fonctions de hachage, la signature numérique, la gestion des clés et le protocole de Diffie-Hellman.

– chapitre 3 : Dans ce chapitre nous exposerons les deux techniques de dissimulation de données, à savoir, la stéganographie et marquage numérique. Les concepts sous-jacents à chaque technique seront soulignés et les limitations de chacune montrées. Nous y montrerons leur mode de fonctionnement, leurs applications respectives ainsi que les attaques possibles sur chacune. Nous y verrons que les attaques de types cryptographiques sont les plus pernicieuses. On s'intéressera plus particulièrement au marquage semi-fragile de préférence au marquage fragile, et au marquage sans perte qui supprime les contraintes de distorsion minimale.

– chapitre 4 : L'objectif de ce chapitre est de dresser un état de l'art concernant les différentes méthodes permettant d'assurer un service d'intégrité adapté aux images par le biais du marquage numérique. Nous introduirons cette notion d'intégrité sémantique particulière aux images, ainsi que les critères à prendre en considération pour construire un système d'authentification performant. Nous y envisagerons une classification des algorithmes d'authentification des images suivant qu'ils assurent un service d'intégrité stricte ou bien une intégrité en termes de contenu, suivant le mode de stockage des données d'authentification ou encore selon la nature des informations qu'ils enfouissent dans l'image à protéger. Plusieurs algorithmes significatifs seront détaillés afin d'introduire progressivement les notions clés associées à ce type de service. Finalement, nous

identifierons les différentes attaques en montrant comment elles peuvent être montées et comment elles peuvent être évitées

– chapitre 5 : Dans ce chapitre nous exposerons notre système d'authentification « souple » basé sur le contenu sémantique des images. Nous commencerons par une revue sur les méthodes d'analyse d'image par texture pour montrer que les approches par les statistiques de premier et de second ordre permettent une bonne discrimination surfacique. Nous montrerons comment extraire des paramètres de texture, en guise de signature basée sur le contenu, et comment adapter ces paramètres afin de pouvoir discriminer les manipulations d'images autorisées de celles qui seront interdites. La localisation des régions altérées est une fonctionnalité obtenue à travers un mécanisme d'insertion/extraction par bloc. Nous terminerons le chapitre sur un ensemble d'expérimentations destinées à valider l'approche

– chapitre 6 : Nous présenterons dans ce chapitre deux solutions globales de protection des images médicales. La première combinant chiffrement, marquage et compression de l'image médicale, la deuxième consistant en un système d'authentification à clé secrète basé compression utilisant un code d'authentification de messages (MAC)**.** Nous expliciterons dans un premier temps les modules de chiffrement, de compression et de marquage qui composent l'algorithme CC-MARK et nous montrerons la manière dont ils seront combinés pour un maximum d'efficacité. Un ensemble de résultats sera présenté sur un échantillon d'images représentatives.
Dans un second temps, nous présenterons un scénario possible d'utilisation de nos algorithmes qui se manifeste sous la forme d'un système destiné à l'échange d'images médicales entre une communauté bien déterminée d'utilisateurs sur le Net. Les fonctionnalités principales du système seront décrites et les interactions entre les différents acteurs seront montrées. De même, on y établira le protocole englobant les règles sous-jacentes à son utilisation.

– Pour finir, une conclusion générale sera donnée pour faire le point sur l'ensemble des travaux effectués. Nous y présenterons également les différentes perspectives d'études et d'améliorations de notre approche.

Chapitre 1

La protection de documents numériques

Introduction

De nos jours, l'information représente un réel enjeu stratégique et économique et de ce fait, qui contrôle l'information détient énormément de pouvoir. Par conséquent, les techniques de protection des media numériques représentent des enjeux économiques, stratégiques et juridiques considérables. Dans un contexte où les échanges d'informations dématérialisées se développent, il est indispensable de pouvoir bénéficier de systèmes sécurisés, afin de protéger les données à caractère personnel ou confidentiel, ou pour assurer la sécurité des transactions financières et commerciales.
En effet, l'utilisation d'un réseau de communication expose les échanges de médias numérisés à certains risques, qui nécessitent l'existence de mesures de sécurité adéquates. Il est donc nécessaire d'avoir accès à des outils techniques, permettant une protection efficace de ces dernières contre les manipulations arbitraires. Cette nécessité a conduit de nombreux chercheurs à se pencher sur le problème de la sécurisation des données numériques face au piratage et à la contrefaçon, afin notamment de faciliter le développement économique des techniques de communication audiovisuelle en réseaux.

La cryptographie a très longtemps été le seul moyen efficace pour répondre à ces exigences. Cette technologie est ainsi reconnue comme étant un outil essentiel de la sécurité et de la confiance, dans les communications électroniques. Avec le développement du tout numérique, s'est posée la problématique de protéger les contenus multimédia. Les techniques de cryptographie, initialement développés pour protéger des données le plus souvent à caractère textuel, se trouvent souvent

confrontés à des inadéquations liées au caractère multimédia des documents à protéger. Dans ce contexte, les nouvelles technologies de dissimulation de données apparaissent comme étant une alternative pouvant s'avérer efficace et complémentaire aux approches de type cryptographique. Elles vont être amenées à jouer un rôle croissant en matière de protection contre la fraude informatique, de sécurité des données, de protection de la confidentialité des correspondances, de protection du secret professionnel, et du commerce électronique.
Ce chapitre expose les différents services de sécurité engagés dans la protection de documents numériques ainsi que les mécanismes permettant de les assurer. Pour mieux cerner les spécificités de chacun, un comparatif est finalement présenté.

I. Sécurité de documents numériques

La protection des données numériques concerne principalement les quatre aspects suivants:

- La confidentialité;
- L'intégrité;
- L'authentification;
- La non répudiation;

Ces services sont assurés par divers mécanismes de sécurité plus ou moins complexes, que nous exposerons en détail, plus loin dans ce document. Comme nous le verrons, ces mécanismes sont traditionnellement de nature cryptographique, mais la dissimulation de données, offre aussi une alternative intéressante, particulièrement pour les données de type images auxquelles nous nous intéressons particulièrement.

I.1 La confidentialité

Il s'agit de garantir le secret du document numérique transmis ou archivé. Ce service de sécurité consiste à s'assurer que seules les personnes autorisées peuvent prendre connaissance des données échangées. Le mécanisme traditionnel qui permet d'obtenir ce service est le chiffrement des données concernées à l'aide d'un algorithme cryptographique, mais aujourd'hui, d'autres moyens sont aussi utilisés, tels que la stéganographie ou le marquage numérique. Tout, du courrier électronique aux commandes d'administration d'un ordinateur à distance, peut être ainsi protégé sous une forme chiffrée. Trop souvent la cryptologie est limitée dans les esprits à cette fonction de protection de la confidentialité. Sans doute des raisons historiques ne sont pas étrangères à cette confusion. En effet, pendant des siècles, c'est à peu près le seul usage qui en a été fait.
On parle aussi de confidentialité du trafic lorsqu'on désire empêcher l'analyse du trafic en cachant les adresses source et destination, la taille des paquets, la fréquence des échanges, ...

I.2 L'intégrité

Il s'agit de garantir qu'un message ou un document électronique n'a pas été altéré accidentellement ou frauduleusement pendant son transfert sur le canal de communication. Il est particulièrement important que, dans toute négociation ou accord contractuel, on puisse vérifier qu'aucune modification du document électronique n'a été faite.

Pour assurer l'intégrité, on peut utiliser le chiffrement sous sa forme symétrique ou asymétrique, ou la signature numérique ou encore les codes d'authentification de messages. L'intégrité est très liée à l'authentification de l'origine des données, et les deux services sont souvent fournis conjointement.

On distingue deux types d'intégrité :

— L'intégrité en mode non connecté permet de détecter des modifications sur un datagramme individuel, mais pas sur l'ordre des datagrammes.

— L'intégrité en mode connecté permet en plus de détecter la perte de paquets ou leur réordonnancement.

I.3 L'authentification

On distingue deux types d'authentification :

— Authentification d'un tiers : C'est l'action qui consiste à prouver son identité.
Ce service est généralement rendu par l'utilisation d'un "échange d'authentification" qui implique un certain dialogue entre les tiers communicants. Ce dialogue est appelé *protocole d'authentification*.

— Authentification de l'origine des données : Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré.

Dans ce cas, l'authentification désigne souvent la combinaison de deux services : authentification et intégrité en mode non connecté. Ces deux services n'ont en effet pas de sens séparément et sont souvent fournis conjointement. C'est à cet aspect de l'authentification que l'on s'intéresse particulièrement dans cette thèse.

I.4 La non-répudiation

Il s'agit de se protéger contre la contestation d'envoi ou de réception d'un message ou d'un document électronique lors d'une transaction. En d'autres termes, il s'agit de garantir que les partenaires d'une transaction ne puissent nier avoir envoyé ou reçu le document en question. Les signatures numériques basées sur le concept de clefs publiques constituent, comme nous le verrons plus loin, des témoignages de participation dans les échanges électroniques.

II. Les mécanismes de protection

II.1 La cryptographie

La cryptographie est le premier dispositif garantissant la sécurité des documents électroniques.
Elle permet de stocker des informations sensibles ou de les transmettre à travers des réseaux non sûrs (comme Internet) de telle sorte qu'elles ne puissent être lues par personne à l'exception du destinataire convenu.

Les données qui peuvent être lues et comprises sans mesures spéciales sont appelées *texte clair*, et le procédé qui consiste à dissimuler le texte clair de façon à cacher sa substance est appelée *chiffrement.* Chiffrer du texte clair produit un texte illisible appelé texte chiffré ou cryptogramme. Le chiffrement garantit que l'information est cachée à quiconque elle n'est pas destinée, même ceux qui peuvent lire des données chiffrées. Le processus de retour du texte chiffré à son texte clair originel est appelé *déchiffrement* [Sti03].
Plusieurs autres mécanismes cryptographiques sont disponibles pour assurer les différents services cités plus tôt, mais le plus utilisé est, sans conteste, le chiffrement.

Un *algorithme cryptographique*, ou *chiffre*, est une fonction mathématique utilisée dans le processus de chiffrement et de déchiffrement et qui fonctionne en combinaison avec une *clef* qui peut être un mot, un nombre, ou une phrase pour chiffrer le texte clair [Sti03]. Le même texte clair se chiffre en un texte chiffré différent si l'on utilise des clefs différentes. La sécurité des données chiffrées est entièrement dépendante de deux choses: la force de l'algorithme cryptographique et le secret de la clef.
Un algorithme cryptographique, plus toutes les clefs possibles et tous les protocoles qui le font fonctionner constituent un *cryptosystème*.

En matière de protection d'images numériques, on parle plutôt de *brouillage* que de chiffrement. Les techniques de *brouillage* sont des méthodes ad-hocs qui différent un peu de celles plus générales que nous allons présenter dans le chapitre suivant et qui sont, le plus souvent, basées sur la transposition de lignes et de colonnes.

II.2 La dissimulation de données

La dissimulation de données (en anglais *data hiding*) désigne l'insertion dans un support numérique d'une certaine *quantité d'information* binaire secrète de manière *imperceptible* et plus ou moins *robuste***,** suivant l'application visée [DR99]. Le terme « dissimulation » ne signifie pas ici que l'information est visible mais codée, il s'agirait alors de cryptographie. Il signifie plutôt que la présence de l'information à protéger (appelée message utile) n'est pas perceptible parce que enfouie dans une autre information (appelée message de couverture). Dans le cas de la protection des informations numériques, le message utile permet d'*identifier le propriétaire* du message de couverture ou *son origine* ou encore de garantir son *intégrité*.
Les techniques de dissimulation de données se servent des spécificités des supports multimédia pour, d'une part offrir des services de sécurité similaires à

ceux offerts par la cryptograhie, mais surtout ajouter des briques élémentaires à l'édifice.

La dissimulation de données englobe deux techniques très proches l'une de l'autre, mais qui n'ont pas les mêmes objectifs, ni les mêmes contraintes. Selon le contexte, on distingue :
- la stéganographie où il doit être impossible de distinguer si le message de couverture contient un message utile ou non. La contrainte la plus importante est alors l'imperceptibilité.
- Le marquage numérique où le message utile est lié à l'identité de l'ayant droit du document de couverture, et doit donc rester présent même si celui-ci subit des modifications. La contrainte principale est alors la robustesse.
Bien que le procédé de dissimulation de données soit applicable à tout type de données (texte, image ou son), on ne s'intéressera dans ce travail qu'au seul cas des images numériques.

- La quantité d'information cachée, appelée aussi *capacité* du canal caché, est exprimée en bits. La capacité nécessaire à un système de dissimulation de données dépend du cadre applicatif. Elle varie de 1 bit à plusieurs centaines, pour les applications courantes.

- Par *imperceptibilité*, on entend à la fois :
  – l'*imperceptibilité perceptuelle* : par exemple, l'oeil humain ne doit pas pouvoir faire la *différence* entre une image dans laquelle on a inséré de l'information et la même image non marquée,
  – l'*imperceptibilité numérique* : idéalement, une personne non autorisée ne doit pas pouvoir s'apercevoir de la *présence* du marquage.
  En marquage numérique, on se focalise principalement sur l'imperceptibilité perceptuelle. En stéganographie, on porte une attention particulière à l'imperceptibilité numérique.

- Par *robustesse*, on entend que l'information insérée doit pouvoir être récupérée même si l'image hôte a subi des modifications. Cette caractéristique est typique au marquage numérique. L'ampleur des modifications (qu'on nommera attaques) auxquelles un marquage doit survivre sera discutée dans le prochain chapitre, elle varie en fonction de chaque cadre applicatif.

- Par *sécurité*, on entend qu'une personne non autorisée ne doit pas pouvoir *détecter* ou *relire* l'information insérée, suivant le cadre applicatif considéré.

Ces quatre contraintes sont souvent antinomiques, et nécessitent un compromis. C'est le cadre applicatif qui va déterminer l'importance relative de ces contraintes entre elles. Intuitivement, on sent bien que plus on cachera d'information, moins elle sera robuste, et inversement. En général, c'est la contrainte d'imperceptibilité qui est la plus forte; on ne veut pas que l'insertion dégrade le support au point qu'il devienne inutilisable.
On cherche généralement à caractériser une autre grandeur : *la plus petite quantité de compromis* en insertion. En fonction de l'application visée, on cherchera à

optimiser préférentiellement telle ou telle contrainte pesant sur un schéma d'insertion. Il n'est pas toujours possible de respecter précisément les contraintes que l'on se fixe.
Notons encore la différence entre *détection* et *décodage* de l'information insérée :
– La détection cherche à répondre à la question : Le support contient-il un message secret ?
– Le décodage cherche à répondre à la question : Quel est le message que l'on a inséré dans le support ?
Nous verrons au chapitre III que le problème de la « simple » détection trouve des applications parfois très importantes.

III. Comparatif des différentes techniques

III.1 Cryptographie vs stéganographie

— La stéganographie aborde la protection des données par une approche différente de celle de la cryptographie. Comme dans le cas de la cryptographie, la technique permet d'échanger des messages avec un correspondant sans que des personnes non autorisées ne puissent en prendre connaissance. Mais alors qu'avec la cryptographie, la sécurité repose sur le fait que le message transmis est incompréhensible, en matière de stéganographie, la sécurité repose sur la remise en question même de l'existence du message secret. De plus, le fait même de chiffrer les données montre que ces dernières ont de la valeur.

— La cryptographie protége les données numériques durant leur transmission entre un émetteur et un récepteur. Après réception et déchiffrement, les données sont identiques aux données d'origine mais elles ne sont plus protégées. Donc le document peut être facilement recopié et redistribué. Avec la stéganographie le message secret reste protégé même après réception.

— Une autre différence très importante entre la cryptographie et la stéganographie se situe au niveau des attaques qui peuvent avoir lieu contre ces techniques. En cryptographie, l'ennemi va tenter de déchiffrer le message, alors qu'en stéganographie l'ennemi va tenter de découvrir le médium de couverture.

III.2 Stéganographie vs marquage numérique

— Dans le cas de la stéganographie, on cherche à cacher une quantité très importante de données qui peut aller jusqu'à dissimuler une image dans une autre image.

— Dans le cas du marquage numérique, on cherche juste à marquer une image en dissimulant une quantité limitée d'information qui à pour but par exemple de démontrer l'intégrité du document ou encore de protéger les droits d'auteurs. Souvent, on se limite à la dissimulation d'un seul bit: image marquée/non marquée.

— Dans la stéganographie, l'existence du message caché doit rester secrète, alors que pour le marquage numérique, seul le message doit rester caché mais son existence, tant qu'on ne peut pas le supprimer ou le modifier, peut être connue. En fait, on peut considérer que le marquage numérique est une sous discipline de la stéganographie.

— En matière d'attaques, en stéganographie, le pirate va chercher à lire les données dissimulées dans le document, tandis que dans le cas d'un document marqué, l'attaquant va chercher à « laver » le document de toute signature possible (ou alors il peut essayer d'usurper l'identité de l'auteur en remplaçant la marque).

Enfin, la cryptographie, la stéganographie, et le marquage numérique sont trois disciplines très proches les unes des autres puisque toutes les trois consistent à protéger une information à caractère sensible.

Conclusion

Ce chapitre, très court, pose la problématique de la sécurisation des échanges de documents numériques tout en montrant les moyens d'y parvenir. Trois techniques différentes ont été succinctement présentées, en attendant d'être dûment détaillées dans les deux chapitres suivants. Le point commun qui nous intéresse entre ces trois techniques, est leur aptitude à authentifier les images numériques, fonctionnalité très utile dans tout système d'échange d'images, et qui constitue le thème central de notre étude.

La méthode la plus utilisée jusqu'à présent a été la cryptographie, qui est utilisée pour protéger les données numériques durant leur transmission entre un émetteur et un récepteur. Après réception et déchiffrement, les données sont identiques aux données d'origine mais elles ne sont plus protégées. Donc le document peut être facilement recopié et redistribué. Pour remédier à cela, les techniques de stéganographie et du marquage numérique ont été avancées. Ces nouvelles techniques reposent sur une approche de dissimulation d'informations à travers des données hôtes. La plus intéressante est sans doute le marquage numérique qui permet de protéger le contenu d'un document tout en le laissant accessible.

Même si l'objectif principal de cette thèse est orienté vers la dissimulation de données, il nous parait utile de présenter d'abord les différents outils cryptographiques et leurs fonctionnalités au niveau du chapitre suivant, pour mieux cerner les possibilités et les limites en matière de sécurité des autres techniques. Cette partie sert donc d'introduction aux deux chapitres suivants.

Chapitre 2

Outils cryptographiques

Introduction

Les techniques de cryptographie représentent des enjeux économiques, stratégiques et juridiques considérables. Procédé d'origine militaire, la cryptographie reste considérée comme un enjeu de sécurité intérieure et extérieure, par un certain nombre de gouvernements, malgré le développement des utilisations civiles et commerciales de ces techniques.

Dans un contexte où les échanges d'informations se développent, il est indispensable de pouvoir bénéficier de systèmes sécurisés, afin de protéger les données à caractère personnel ou confidentiel, ou pour assurer la sécurité des transactions financières et commerciales. En effet, l'utilisation d'un réseau de communication expose les échanges à certains risques, qui nécessitent l'existence de mesures de sécurité adéquates. Il est donc nécessaire d'avoir accès à des outils techniques, permettant une protection efficace de la confidentialité des données, et des communications contre les intrusions arbitraires. Le chiffrement des données est très souvent le seul moyen efficace pour répondre à ces exigences. Les technologies cryptographiques sont ainsi reconnues comme étant des outils essentiels de la sécurité et de la confiance, dans les communications électroniques. Elles vont être amenées à jouer un rôle croissant en matière de protection contre la fraude informatique, de sécurité des données, de protection de la confidentialité des correspondances, de protection du secret professionnel, et du commerce électronique.

Dans ce chapitre nous exposons les principes de base de la cryptographie en insistant particulièrement sur certains concepts intervenant dans les méthodes de marquage décrites par la suite, ou qui nous serons utiles dans nos contributions, comme le chiffrement par blocs, les fonctions de hachage, la signature numérique et la gestion des clés.

I. Définitions

La cryptographie est l'étude des méthodes permettant de transcrire des données intelligibles, en des données inintelligibles, par l'application de transformations mathématiques dont l'effet est réversible [Sti03].

Ces transformations, basées le plus souvent sur l'arithmétique modulaire, désignent un processus appelé *chiffrement* (noté E), qui donne un texte chiffré C ou cryptogramme, à partir d'un texte en clair M. On a donc que

$$E(M) = C.$$

Inversement, le *déchiffrement* (noté D), est le processus qui permet de reconstruire le texte en clair à partir du texte chiffré. On a alors que

$$D(C) = D(E(M)) = M$$

En pratique, E et D sont des fonctions paramétrées par des clefs K_e et K_d

$$\begin{cases} E_{Ke}(M) = C \\ D_{Kd}(C) = M \end{cases}$$

Un système cryptographique se base donc sur une paire d'algorithmes (chiffrement et déchiffrement) et une clef k appartenant à l'espace des clefs $\mathcal{K}$. La robustesse des algorithmes est généralement basée sur la difficulté de calcul mathématique. Cette difficulté est souvent liée à la longueur de la clef.
L'espace des clefs définit deux grandes catégories de systèmes cryptographiques:

- les systèmes à clef secrète (ou symétriques) pour lesquels $K_e = K_d = K$
- et les systèmes à clef publique (ou asymétriques) pour lesquels $K_e \neq K_d$

Parallèlement, la *cryptanalyse* est l'étude des procédés cryptographiques ayant pour but de trouver des faiblesses dans le système cryptographique afin de pouvoir *décrypter* des textes chiffrés sans disposer de la clef de déchiffrement.

Le *décryptage* est le processus consistant à retrouver le texte en clair sans connaître la clef de chiffrement [Sti03].

L'ensemble des deux disciplines de cryptographie et de cryptanalyse, forme la discipline de la cryptologie ou « science du secret ».

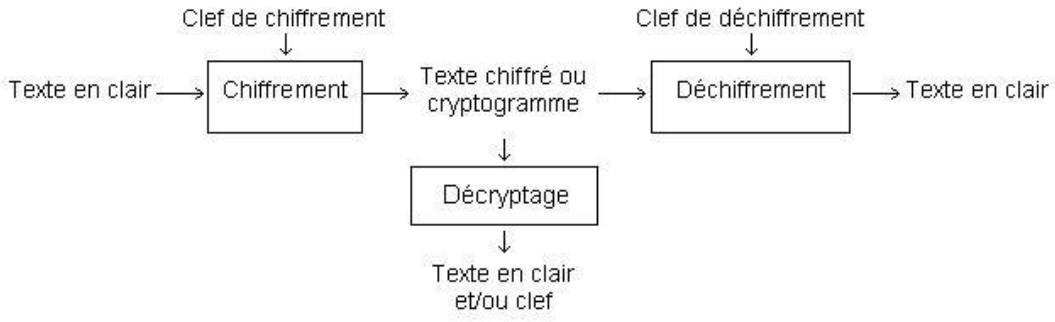


Figure 2.1 Schéma générique de la cryptologie

II. Le chiffrement

Le chiffrement est le mot clé de la cryptographie. Différentes méthodes de chiffrement ont été imaginées, à travers les siècles, pour se protéger de la curiosité et de la malveillance de ses ennemis.
Les transformations de la cryptographie traditionnelle sont basées sur les caractères, en les remplaçant par d'autres caractères ou en transposant les caractères ou en faisant les 2 opérations plusieurs fois [Sti03] .

Les transformations de la cryptographie moderne sont essentiellement basées sur l'arithmétique modulaire et le ou exclusif (xor). Les caractères qui composent le message sont d'abord transformés en une succession de bits, puis une série de transformations est opérée sur ces bits.
Avant l'apparition des ordinateurs, la sécurité du chiffrement reposait sur le secret des opérations réalisées, il suffisait de connaître la façon de coder pour décoder très facilement. Aujourd'hui, les nouveaux algorithmes de chiffrement utilisés sont publics, et leur sécurité repose plutôt sur le concept des clefs comme le stipule le principe de Kerckhoffs, énoncé plus tard, au niveau du paragraphe VI.3.2.

II.1 Chiffrement par substitution

Il consiste à remplacer les caractères (voire les mots) par d'autres symboles. Ceci implique le choix d'un ensemble de symboles qui devront jouer le rôle de substituts.
La complexité des systèmes à substitutions dépend de 3 facteurs :
— la composition spécifique de l'alphabet utilisé pour chiffrer ou pour communiquer,
— Le nombre d'alphabets utilisés dans le cryptogramme,
— La manière spécifique dont ils sont utilisés.
On distingue couramment quatre types de substitutions différentes :

II.1.1 Substitution simple ou substitution monoalphabétique

Chaque caractère du texte en clair est remplacé par un caractère correspondant dans le texte chiffré. Les exemples les plus célèbres sont les algorithmes de César,

Rot13, et le code morse [Sti03]. Les algorithmes à base de substitutions monoalphabétiques sont facilement cassés par une cryptanalyse rudimentaire.

II.1.2 Substitution homophonique :

Même principe que le précédent, sauf qu'à un caractère du texte en clair on fait correspondre plusieurs caractères dans le texte chiffré. Par exemple, " A " peut correspondre à 5, 13, 25 ou 56 ; " B " 7, 19, 31, ou 42 ; etc. Ce procédé est plus sûr, mais peut aussi être craqué par une cryptanalyse plus pointue.

II.1.3 Substitution polyalphabétique :

Le principe ici consiste à remplacer chaque caractère du message en clair par un nouveau caractère pris dans ou un plusieurs alphabets aléatoires associés. Par exemple, on pourra utiliser n substitutions. L'illustration la plus simple qui correspond à ce principe est l'utilisation d'une fonction à base de ou exclusif (XOR) [Sta03]. L'exemple le plus célèbre de ce type est l'algorithme de VIGENERE. Il consiste à coder les mots d'un texte en ajoutant à chacun de ses caractères un caractère d'un autre mot appelé clef. On associe dans un premier temps à chaque caractère le code ASCII correspondant. Le code ASCII de chacun des caractères de la clef est ajouté indéfiniment en vis-à-vis avec les codes des caractères du texte à chiffrer.
Par exemple le texte " rendezvousamidi " avec la clé "bonjour" sera codé de la manière suivante:

Texte original:

r	e	n	d	e	z	v	o	u	s	a	m	i	d	i
18	5	14	4	5	26	22	15	21	19	1	13	10	4	10

Clé:

b	o	n	j	o	u	r
2	15	14	10	15	21	18

Texte chiffré

r+b	e+o	n+n	d+j	e+o	z+u	v+r	o+b	u+o	s+n	a+j	m+o	i+u	d+r	i+b
8+2	5+15	14+14	4+10	5+15	26+21	22+18	15+2	21+15	19+14	1+10	13+15	10+21	4+18	10+2

Pour déchiffrer ce message il suffit d'avoir la clé secrète et faire le processus inverse, à l'aide d'une soustraction.
Le problème posé est que lorsque les messages sont beaucoup plus longs que la clé, il est possible de repérer la longueur de la clé et d'utiliser pour chaque séquence de la longueur de la clé la méthode consistant à calculer la fréquence d'apparition des lettres, permettant de déterminer un à un les caractères de la clé.
La solution consiste à utiliser une clé dont la taille est proche de celle du texte afin de rendre impossible une étude statistique du texte chiffré. Ce type de système de chiffrement est appelé *système à clé jetable* [Sti03].

II.1.4 Substitution par polygrammes

Les caractères du texte en clair sont chiffrés par blocs. Par exemple, " ABA " peut être chiffré par " RTQ " tandis que " ABB " est chiffré par " SLL ". Les exemples les plus célèbres sont les algorithmes de PLAYFAIR et de HILL [Sta03].

II.2 Chiffrement par transposition

Avec le principe de la transposition toutes les lettres du message sont présentes, mais dans un ordre différent. Ce type de chiffrement utilise le principe mathématique des permutations. Plusieurs types différents de transpositions existent :

II.2.1 Transposition simple par colonnes

Le message à chiffrer est écrit horizontalement dans une matrice prédéfinie, et le texte à chiffrer se déduit en lisant la grille verticalement. Pour déchiffrer le message on réalise le procédé inverse. L'algorithme allemand ADFGFX [Sta03] est fondé sur ce principe.

II.2.2 Transposition complexe par colonnes

Un mot clé secret (avec des caractères tous différents) est utilisé pour dériver une séquence de nombres commençant à 1 et finissant au nombre de lettres composant le mot clé. Cette séquence est obtenue en numérotant les lettres du mot clé en partant de la gauche vers la droite et en donnant l'ordre d'apparition dans l'alphabet. Une fois que la séquence de transposition est obtenue, on chiffre en écrivant d'abord le message par lignes dans un rectangle puis le texte est lu par colonnes, en suivant l'ordre déterminé par la séquence.

II.2.3 Transposition par carré polybique

Un mot clé secret est utilisé pour construire un alphabet dans un tableau. Les coordonnées des lignes et des colonnes correspondant aux lettres du texte à chiffrer sont utilisées pour transcrire le message en chiffres. Avec ce procédé, chaque lettre du texte en clair est représentée par deux chiffres écrits verticalement. Ces deux coordonnées sont ensuite transposées en les recombinant par deux sur la ligne ainsi obtenue.

II.3 Systèmes symétriques ou à clef secrète :

Dans ce type de système, la même clé est partagée par l'émetteur et le récepteur pour chiffrer et déchiffrer l'information. Le problème de cette méthode est de trouver alors le moyen de transmettre de manière sécurisée la clé à son correspondant. Parmi les systèmes symétriques**,** on distingue :

II.3.1 les algorithmes de chiffrement en continu ou de flux (*Stream Cipher*), qui agissent sur le texte en clair un bit à la fois (Ex : RC4).

II.3.2 les algorithmes de chiffrement par blocs, qui opèrent sur le texte en clair par groupes de bits appelés blocs. Ces algorithmes peuvent être utilisés suivant 2 modes :

— **le mode ECB (**Electronic CodeBook**) :**

Découper le message en blocs de taille fixe puis chiffrement bloc par bloc. Ce mode a l'avantage de permettre le chiffrement en parallèle des différents blocs composant le message. Cependant, l'inconvénient est que deux blocs identiques donnent toujours une même valeur, ce qui peut donner des informations sur le message : un attaquant actif pourra manipuler les messages chiffrés en retirant, répétant ou interchangeant des blocs. De plus, si un bit du texte chiffré est modifié pendant le transfert, tout le bloc déchiffré correspondant sera faux.

— **le mode CBC (**Cipher Block Chaining**) :**

- Obtenir une valeur aléatoire appelée Vecteur d'Initialisation VI,
- Appliquer le ou exclusif (xor) avec le premier bloc,
- Chiffrer avec la clef secrète,
- Utiliser la valeur obtenue comme valeur aléatoire pour le bloc suivant.

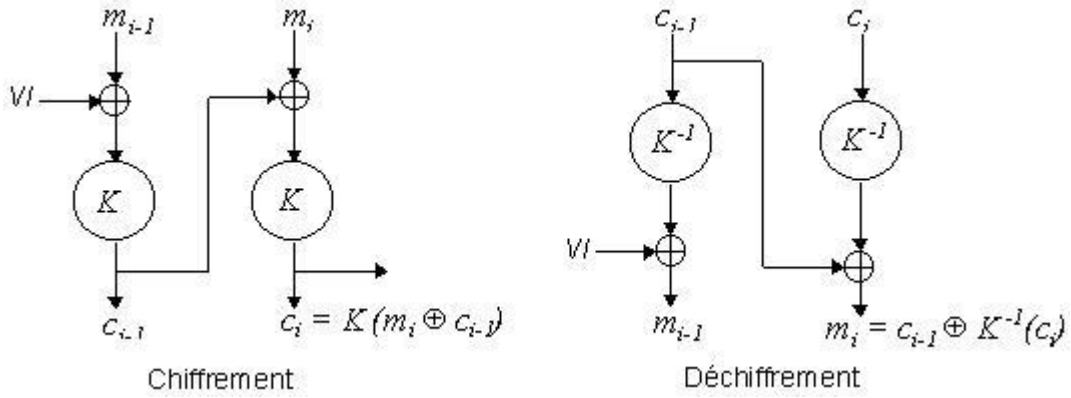


Figure 2. 2 Le mode CBC

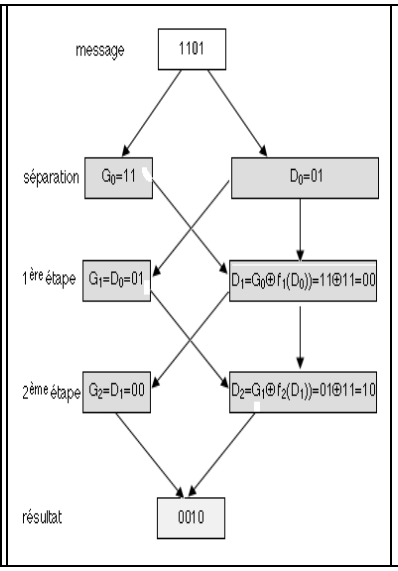
La taille des blocs a un impact sur la sécurité et sur la complexité : les blocs de grandes dimensions sont plus sécuritaires mais sont plus lourds à implémenter. L'avantage de ce mode est que l'utilisation d'un VI différent pour chaque message, assure que deux messages identiques donneront des cryptogrammes totalement différents. Mais l'inconvénient réside dans la nécessité de transmettre la valeur initiale. De plus, il n'est plus possible de paralléliser le chiffrement des différents blocs;

On pourrait craindre que le chaînage de bloc n'entraîne une propagation d'erreur importante. De fait, une erreur d'un bit sur le texte en clair affectera tous les blocs chiffrés suivants. Par contre, si un bit du texte chiffré est modifié au cours du transfert, seul le bloc de texte en clair correspondant et un bit du bloc de texte en clair suivant seront endommagés : le mode CBC est dit auto-réparateur.

II.3.3 Chiffrement par blocs avec itération :

Ce type d'algorithme chiffre les blocs par un processus comportant plusieurs rondes. Dans chaque ronde, la même transformation est appliquée au bloc, en utilisant une sous-clef dérivée de la clef de chiffrement. On parle alors de réseau ou chiffre de Feistel [Sta03].
Dans un réseau de Feistel, un bloc du texte en clair est découpé en deux, la transformation de ronde est appliqué à une des 2 moitiés, et le résultat est combiné avec l'autre moitié par « ou exclusif ». Les 2 moitiés sont alors inversées pour l'application de la ronde suivante.
L'exemple simple suivant illustre ce principe :

entrée	f_1	sortie
00	→	01
01	→	11
10	→	10
11	→	01
entrée	f_2	sortie
00	→	11
01	→	00
10	→	00
11	→	01



message		résultat
0000	→	0100
0001	→	1100
0010	→	1010
0011	→	0111
0100	→	0011
0101	→	1001
0110	→	1111
0111	→	0000
1000	→	1101
1001	→	0101
1010	→	0001
1011	→	1110
1100	→	1000
1101	→	0010
1110	→	0110
1111	→	1011

Figure 2.3 Chiffre de Feistel à 2 rondes

L'algorithme de chiffrement symétrique le plus populaire est le DES (Data Encryption Standart) qui est un chiffre de Feistel à 16 rondes. Le DES utilise des blocs de 64 bits, et une clef formée de 64 bits dont 56 utiles et 8 de parité.

La transformation DES est en fait la composition de 16 transformations élémentaires paramétrées par des sous-clefs (K_1,K_2…K_{16}) extraites de la clef initiale. La transformation élémentaire T_i, paramétrée par K_i est la suivante :
– le bloc d'entrée de 64 bits est découpé en deux blocs de 32 bits L_i et R_i (pour Left et Right);
– on applique ensuite la relation suivante :

$$L_{i+1} = R_i \; R_{i+1} = L_i \oplus f(R_i,K_i)$$

Cette transformation est quasi involutive et ce, quelle que soit la nature de *f*. En effet, il suffit d'inverser le rôle de R et L pour trouver la transformation inverse :

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus f(L_{i+1};K_i)$$

Il est facile de vérifier que l'algorithme de déchiffrement est identique à celui du chiffrement, à condition d'utiliser la séquence des sous-clefs dans l'ordre inverse (K_{16}, K_{15} ... K_2, K_1).
Pendant longtemps le DES demeura le standard en matière de chiffrement (depuis 1977), mais avec la puissance de calcul des nouveaux ordinateurs, la clé de 56 bits est devenue trop petite. Depuis 2000, le nouveau standard est A.E.S. (Advanced Encryption Standart) qui peut être utilisé avec 128, 192 ou 256 bits. D'autres systèmes de chiffrement symétriques d'utilisation courante sont :

- IDEA (International Data Encryption Algorithm) avec des blocs de 64 bits et une clef de 128 bits,
- Triple DES avec des blocs de 64 bits et une clef de 112 bits,
- **RC5 (Rivest's Code n° 5).**
- Blowfish

II.4 Systèmes asymétriques ou à clef publique

Le concept de cryptographie à clef publique fut inventé par Whitfield Diffie et Martin Hellman en 1976, dans le but de résoudre le problème de la distribution des clefs posé par la cryptographie à clef secrète [Sti03]. De nombreux algorithmes ont été proposés depuis, tous basés sur des problèmes mathématiques sophistiqués difficiles à résoudre.

Ici, les clefs de chiffrement et de déchiffrement sont distinctes et ne peuvent se déduire l'une de l'autre. On peut donc rendre l'une des deux publique tandis que l'autre reste privée. Si la clef publique sert au chiffrement, tout le monde peut chiffrer un message, que seul le propriétaire de la clef privée pourra déchiffrer. Certains algorithmes permettent d'utiliser la clef privée pour chiffrer. Dans ce cas, n'importe qui pourra déchiffrer, mais seul le possesseur de la clef privée peut chiffrer. Cela permet donc la signature numérique de messages. Certains algorithmes asymétriques ne sont adaptés qu'au chiffrement, tandis que d'autres ne permettent que la signature numérique.

L'algorithme asymétrique le plus populaire est sans conteste RSA [Web13]. Inventé par trois chercheurs du MIT en 1978, Rivest, Shamir et Adleman, RSA permet le chiffrement et la signature numérique. Il est aujourd'hui encore très largement utilisé. Cet algorithme repose sur la difficulté de factoriser des grands nombres entiers. Il est facile de multiplier deux nombres premiers, par exemple 127 et 997, et de trouver 126 619. Mais il est plus difficile de factoriser, c'est-à-dire de retrouver 127 et 997 à partir de 126 619.

La génération des paires de clefs se fait de la manière suivante :
- Chaque utilisateur va choisir deux grands nombres premiers, p et q, et calcule n = pxq. n est rendu public, p et q doivent rester secrets et sont donc détruits une fois les clefs générées.
- On choisit ensuite aléatoirement une clef publique e telle que e et (p-1)**.**(q-1) soient premiers entre eux.
- La clef privée d est obtenue grâce à l'algorithme d'Euclide :

$$e.d \equiv 1 \bmod (p-1)(q-1).$$

La fonction de chiffrement est alors, de façon simplifiée, $c = m^e \bmod n$, m étant le message en clair et c le cryptogramme. Si m est plus grand que n, il est divisé en morceaux de valeur inférieure à n et chaque morceau est chiffré séparément suivant cette formule. Du fait de la relation entre e et d, la fonction de déchiffrement correspondante est $m = c^d \bmod n$.
L'exemple simple suivant montre le procédé de chiffrement /déchiffrement :

- Soit p = 3 et q = 11, on a donc n = pxq = 33;
- Ainsi, z = (p-1)x(q-1) = 2 x10 = 20 ;
- On choisit aléatoirement e = 3, qui n'a pas de facteur commun avec 20;
- On cherche $d = e^{-1} \bmod (20)$, soit d = 7.

Si le texte à chiffrer est : " ASSEZ " en codant chaque lettre avec son numéro alphabétique **:**

	Texte en clair (p)			texte chiffré (C)		Après déchiffrement
Caractère	valeur	P^3	P^3 (mod33)	C^7	C^7 (mod 33)	caractère
A	01	1	1	1	1	A
S	19	6859	28	13492928512	28	S
S	19	6859	28	13492928512	28	S
E	05	125	26	8031810176	5	E
Z	26	17576	20	1280000000	26	Z
			Calculs de l'émetteur		calculs du récepteur	

Figure 2.4 Exemple de chiffrement avec RSA

Pour un cryptanalyste, retrouver la clef privée à partir de la clef publique nécessite de connaître (p-1)(q-1) = p.q-p-q+1 = n+1-p-q, donc de connaître p et q. Pour cela, il doit factoriser le grand nombre n. Donc n doit être suffisamment grand pour que cela ne soit pas possible dans un temps raisonnable par rapport au niveau de sécurité requis. Actuellement, la longueur du module n varie généralement de 512 à 2048 bits suivant les utilisations. Compte tenu de l'augmentation des vitesses de calcul des ordinateurs et des avancées mathématiques en matière de factorisation des grands nombres, la longueur minimale des clefs doit augmenter au cours du temps.

D'autres exemples d'algorithmes asymétriques sont Elgamal [Sti03] et Rabin [Sti03].

Tous les algorithmes asymétriques actuels présentent l'inconvénient d'être bien plus lents que les algorithmes à clef secrète ; de ce fait, ils sont souvent utilisés non pour chiffrer directement des données, mais pour chiffrer une *« clef de session »* secrète, particulièrement dans des applications impliquant l'échange d'une grande quantité d'informations.

- échange d'une clef de session sur le canal par un algorithme de chiffrement asymétrique,
- Utilisation de cette clef pour la communication de données à l'aide d'un algorithme de chiffrement symétrique;

De tels systèmes sont dits *hybrides*, dont un exemple est PGP[Web14], utilisé notamment pour le courrier électronique.
Créé en 1992 par Philip Zimmermann, PGP (Pretty Good Privacy) est un système de chiffrement à clé publique qui combine les avantages d'un IDEA performant et d'un RSA robuste : IDEA est 1000 fois plus rapide que RSA et il est pratiquement impossible de percer la clé secrète RSA.

Voici ce qui se passera, lorsqu'on utilise PGP pour chiffrer un message e-mail par exemple :

- PGP génère d'abord une clé aléatoire de session pour le message;
- Il utilise l'algorithme IDEA pour chiffrer le message avec la clé de session;
- Il emploie ensuite l'algorithme RSA pour chiffrer la clé de session avec la clé publique du destinataire,
- Il prépare enfin le tout, la clé de session chiffrée et le message chiffré, pour l'envoi par e-mail.

De plus, les cryptogrammes sont compressés avec le système zip de façon à prendre moins de place.

Le déchiffrement est le processus inverse, et se déroule donc comme suit :

- PGP utilise IDEA pour déchiffrer la clé secrète sur disque du destinataire avec comme clé le mot de passe fourni au clavier par le destinataire,
- Il emploie RSA pour déchiffrer la clé de session avec la clé secrète du destinataire,
- Il utilise de nouveau IDEA pour déchiffrer enfin le message avec la clé de session.

III. Authentification et contrôle d'intégrité

III.1 Les fonctions de hachage

Une fonction de hachage ou fonction de condensation est une fonction qui convertit un message de longueur quelconque en une chaîne de taille inférieure et fixe, appelée *empreinte* ou *condensé* (*ou digest* en anglais) du message initial.
Une fonction de hachage *à sens unique* est une fonction de hachage avec laquelle il est aisé de calculer l'empreinte d'un message donné, mais il est difficile d'engendrer des messages qui ont une empreinte donnée, et donc de déduire le message initial à partir de l'empreinte. On demande généralement en plus à une telle fonction d'être *sans collision*, c'est-à-dire qu'il soit impossible de trouver deux messages ayant la même empreinte. En fait, on utilise souvent le terme

fonction de hachage pour désigner une fonction de hachage à sens unique sans collision.
La plupart des fonctions de hachage sont construites par itération d'une fonction de compression : le message *M* est décomposé en n blocs $m_1,...,m_n$, puis une fonction de compression *f* est appliquée à chaque bloc et au résultat de la compression du bloc précédent ; l'empreinte notée *h(M)* est le résultat de la dernière compression.

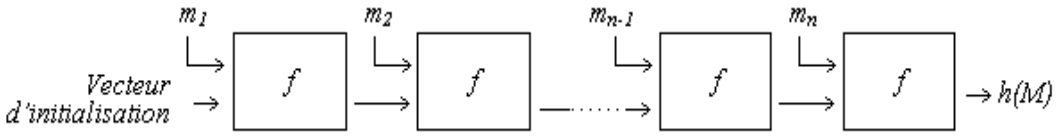


Figure 2.5 Fonction de hachage itérative

Des exemples de fonctions de hachage couramment utilisées sont MD5 [Web1], SHA1 [Web2] et RIPE-MD[Web14] :

- MD5 (Message Digest 5) **:** Développé par Rivest en 1991, MD5 produit une empreinte de 128 bits à partir d'un texte d'entrée de taille arbitraire manipulé par blocs de 512 bits.

- SHA (Secure Hash Algorithm) **:** c'est la norme du gouvernement Américain pour le hachage. SHA-1 est une amélioration de SHA qui produit une empreinte de 160 bits à partir d'un message de longueur maximale de 2^{64} bits. Tout comme MD5, SHA-1 travaille sur des blocs de 512 bits.

- RIPE-MD : Développée dans le cadre du projet RIPE (RACE Integrity Primitives Evaluation) de la communauté Européenne, RIPE-MD fournit une empreinte de 128 bits. RIPE-MD-160 est une version renforcée de RIPE-MD qui fournit une empreinte de 160 bits.

III.2 La signature numérique

La norme ISO 7498-2 [Web 16], définit la signature numérique comme des « données ajoutées à une unité de données, ou transformation cryptographique d'une unité de données, permettant à un destinataire de prouver la source et l'intégrité de l'unité de données et protégeant contre la contrefaçon ».
La mention « protégeant contre la contrefaçon » implique que seul l'expéditeur doit être capable de générer la signature. Une signature numérique fournit donc les services d'*authentification* de l'origine des données, d'*intégrité* des données et de *non-répudiation*.
Sur le plan conceptuel, la façon la plus simple de signer un message consiste à chiffrer celui-ci à l'aide d'une clef privée d'un système à clef publique : seul le possesseur de cette clef est capable de générer la signature, mais toute personne ayant accès à la clef publique correspondante peut la vérifier. Dans la pratique, cette méthode s'avère peu utilisable du fait de sa lenteur, et on préfère calculer d'abord une empreinte du message à signer et à ne chiffrer que cette empreinte. Le calcul d'une empreinte par fonction de hachage étant rapide et la quantité de données à chiffrer étant fortement réduite, cette méthode est bien plus rapide.

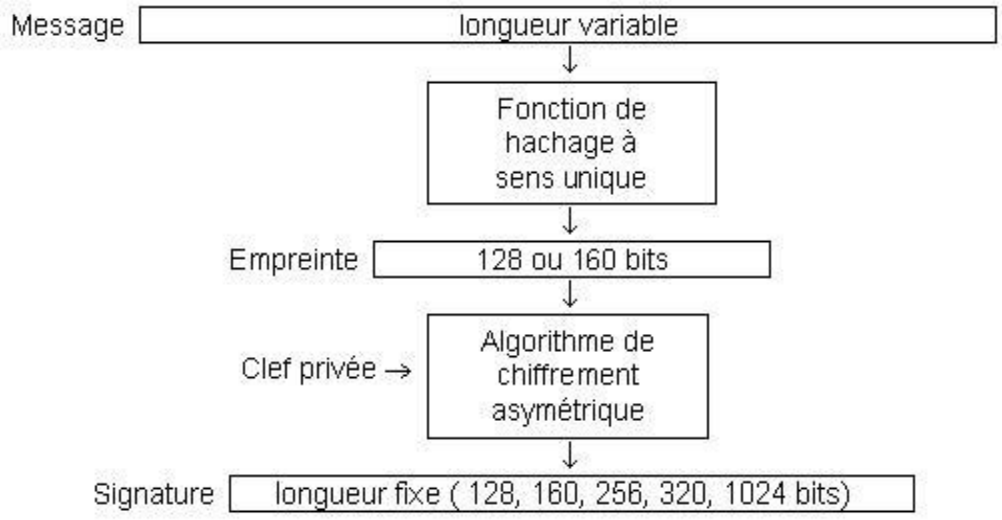


Figure 2.6 Obtention d'une signature numérique

DSS (Digital Signature Standard)**:** C'est la norme officielle de signature numérique du gouvernement Américain. Adoptée en 1994, elle utilise SHA comme fonction de hachage à sens unique et Elgamal pour la génération et la vérification de la signature [Sta03].

RSA : La même procédure RSA utilisée pour le chiffrement à clef publique peut être utilisée pour la signature en inversant e et d, c'est-à-dire en chiffrant avec la clef privée et en déchiffrant avec la clef publique correspondante : $s = m^d \bmod n$ et $m = s^e \bmod n$. Si DSS est la norme officielle aux U.S.A., RSA est une norme de fait, qui est en pratique beaucoup plus utilisé que DSA.

III.3 Les codes d'authentification de message ou MAC

Un code d'authentification de message (Message Authentication Code, MAC) est une empreinte du message dépendant à la fois de l'entrée et d'une clef secrète. On peut construire un MAC à partir d'une fonction de hachage ou d'un algorithme de chiffrement par blocs[Sta03].
Un moyen simple de transformer une fonction de hachage à sens unique en un MAC, consiste à chiffrer l'empreinte avec un algorithme à clef secrète. Une autre façon courante de générer un MAC consiste à appliquer un algorithme de chiffrement symétrique en mode CBC au message; le MAC est alors le dernier bloc du cryptogramme.
Grâce à l'utilisation de la cryptographie à clef secrète, le MAC fournit à la fois les services d'authentification de l'origine des données et d'intégrité des données mais ne fournit pas la non répudiation.

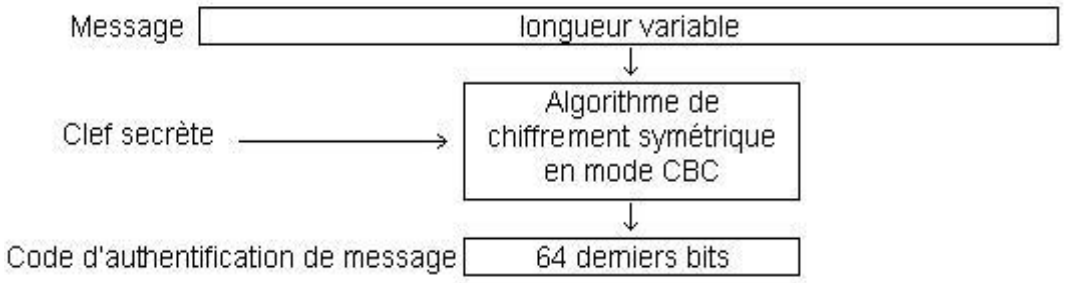


Figure 2.7 MAC obtenu à l'aide d'un algorithme de chiffrement symétrique

Les exemples les plus courants de calcul de MAC sont les suivants :

Keyed-Hash : consiste à appliquer la fonction de hachage non pas simplement aux données à protéger, mais à un ensemble dépendant à la fois des données et d'un secret. Le MAC est alors la valeur de type H(secret, message), H(message, secret) ou H(secret, message, secret). Ces méthodes, présentées en 1992 par Gene Tsudik dans [Tsu92], s'appellent respectivement méthode du préfixe secret, du suffixe secret et de l'enveloppe secrète.

HMAC : présentée dans la RFC 2104 [Web 17], la méthode HMAC peut être utilisée avec n'importe quelle fonction de hachage itérative telle que MD5, SHA-1 ou encore RIPE-MD.
Soit H une telle fonction, K le secret et M le message à protéger. H travaille sur des blocs de longueur b octets (64 en général) et génère une empreinte de longueur l octets (16 pour MD5, 20 pour SHA et RIPE-MD-160). Il est conseillé d'utiliser un secret de taille au moins égale à l octets. On définit deux chaînes, ipad (inner padding data) et opad (outer padding data), de la façon suivante :

ipad = l'octet 0x36 répété b fois,
opad = l'octet 0x5C répété b fois.

Le MAC se calcule alors suivant la formule suivante :

$HMAC_K(M) = H (K \oplus opad, H(K \oplus ipad, M))$.

Une pratique courante avec les fonctions de calcul de MAC est de tronquer la sortie pour ne garder comme MAC qu'un nombre réduit de bits. Avec HMAC, on peut ainsi choisir de ne retenir que les t bits de gauche, où t doit être supérieur à l/2 et 80. On désigne alors sous la forme HMAC-H-t l'utilisation de HMAC avec la fonction H, tronqué à t bits (par exemple, HMAC-SHA1-96).

IV. Accord sur les clefs et authentification mutuelle

Pour établir une communication sécurisée, on procède généralement, en premier lieu, à une authentification à des fins de contrôle d'accès. Puis, un échange de clef permet l'utilisation d'un mécanisme de sécurisation des échanges : l'authentification est ainsi étendue à la suite de la communication.
Souvent on a recours, pour la sécurisation des échanges entre deux parties, à un « tiers de confiance » qui joue le rôle de notaire, en participant à la mise en œuvre de mécanismes de sécurité, notamment en intervenant dans les protocoles d'authentification et d'échange de clef.

Exemple : Kerberos, Sesame [Sta03].

Un protocole d'établissement de clef permet à deux parties de se mettre d'accord sur une même clef secrète en vue d'un échange d'information;

Exemple : Protocole Diffie-Hellman [Sta03] : ce protocole permet à deux tiers X et Y de générer un secret partagé sans avoir aucune information préalable l'un sur l'autre et sans tiers de confiance. Il est basé sur un mécanisme à clef publique. Sa sécurité dépend de la difficulté de calculer des logarithmes discrets sur un corps fini. Les étapes de ce protocole sont les suivantes :

1. X et Y se mettent d'accord sur un grand entier n tel que (n-1)/2 soit premier et sur un entier g primitif par rapport à n. Ces deux entiers sont publics.
2. X choisit de manière aléatoire un grand nombre entier a, qu'il garde secret, et calcule sa valeur publique, $A = g^a \bmod n$. Y fait de même et génère b et $B = g^b \bmod n$.
3. X envoie A à Y ; Y envoie B à X.
4. X calcule $K_{AB} = B^a \bmod n$; Y calcule $K_{BA} = A^b \bmod n$. $K_{AB} = K_{BA} = g^{ab} \bmod n$ est le secret partagé par X et Y.

Une personne qui écoute la communication connaît *g, n,* $A=g^a \bmod n$ et $B=g^b \bmod n$, ce qui ne lui permet pas de calculer $g^{ab} \bmod n$: il lui faudrait pour cela calculer le logarithme de *A* ou *B* pour retrouver a ou b.

V. Générateurs aléatoires et pseudo-aléatoires

La génération de nombres aléatoires servant à la création de clefs secrètes ou privées nécessite l'utilisation d'aléas imprévisibles par les opposants. Malheureusement, il s'avère impossible de produire des suites vraiment aléatoires à l'aide uniquement d'un ordinateur : le générateur sera toujours périodique, donc prévisible. On a donc recours à des générateurs pseudo-aléatoires dits « cryptographiquement » sûrs. Un tel générateur doit présenter les caractéristiques suivantes [Sti03]:

- La période de la suite doit être suffisamment grande pour que les sous-suites finies utilisées avec l'algorithme ne soient pas périodiques.

- Ces sous-suites doivent, sur le plan statistique, sembler aléatoires. Par exemple, elles doivent disposer du même nombre de 0 et de 1, les distributions des segments de 0 et des segments de 1 doivent être les mêmes et on ne doit pas pouvoir les compresser à moins d'avoir le secret (appelé dans ce contexte *germe*) utilisé pour initialiser le générateur.

- Le générateur doit être imprévisible, au sens où il doit être impossible de prédire le prochain aléa à partir des aléas précédents : si on exécute le générateur de suites deux fois, avec exactement les mêmes entrées, on doit obtenir deux suites aléatoires différentes.

De nombreux générateurs ont été développés dans le monde académique comprenant des tests variés sur leur caractère aléatoire. Tous ces générateurs sont périodiques, mais avec des périodes potentielles suffisamment grandes pour être

utilisés dans les applications les plus exigeantes. Mais le problème de corrélations non désirées persiste. Ce sont ces propriétés que le cryptanalyste utilisera pour attaquer le système.

La plupart des générateurs pseudo-aléatoires sont construits en utilisant des registres à décalage (*shift registers*) et, en particulier, les registres à décalage à rétroaction linéaire (*Linear Feedback Shift Registers, LFSR)*. Ces derniers présentent l'inconvénient de générer des suites linéaires, si bien que des grands nombres générés à partir de sous-suites sont fortement corrélés. C'est pourquoi les générateurs pseudo-aléatoires sont généralement construits en combinant, à l'aide d'une fonction non linéaire, plusieurs registres à décalage de tailles différentes. Ce type de générateur est très utilisé par les algorithmes de chiffrement en continu.

On peut aussi avoir recours à des éléments extérieurs comme les déplacements de la souris sur l'écran, la vitesse de frappe sur un clavier, l'entrée d'un micro enregistrant le bruit atmosphérique, ...etc.

Un exemple de générateur pseudo-aléatoire considéré comme étant sûr est Yarrow [Sta03] développé par Bruce Schneier et John Kelsey de Counterpane. Sa caractéristique principale est que ses composantes sont plus ou moins indépendantes, ce qui permet à différents systèmes comportant des contraintes différentes, d'utiliser son modèle général. Ce système est basé sur l'utilisation d'une fonction de hachage, ainsi que des primitives cryptographiques. D'autres exemples sont : [Sta03]

- BBS ou Blum-Blum-Shub : Développé par L. Blum, M. Blum, et M. Shub, basé sur la théorie des résidus quadratiques.
- MUGI Développé par Hitachi**,**
- STRANDOM (STrong pseudo-RANDOM) : Développé par Y. Zheng.

VI. La cryptanalyse

VI.1 Objectifs de la cryptanalyse

Si la cryptographie s'occupe d'élaborer des méthodes de protection de données, la cryptanalyse va tenter, au contraire, de casser ces protections. L'objectif de la cryptanalyse peut être malveillant, c'est-à-dire essayer de prendre connaissance d'informations confidentielles privées, ou au contraire utile, en mettant à l'épreuve les algorithmes et les protocoles cryptographiques, par le biais de diverses attaques, dans le but de détecter des failles dans ces derniers et les améliorer en conséquence. Il est donc avantageux pour un algorithme cryptographique d'être complètement publié dans le but que des spécialistes étudient sa sécurité.

Une cryptanalyse réussie peut fournir soit le texte en clair, soit la clef. Une tentative de cryptanalyse est appelée *attaque*. Une attaque réussie est appelée *méthode.*

VI.2 Les attaques

On distingue 4 classes d'attaques cryptographiques génériques suivant les informations que peut obtenir le cryptanalyste. Chacune de ces attaques repose sur l'hypothèse que le cryptanalyste dispose de la connaissance complète de l'algorithme de chiffrement :

- L'attaque à texte chiffré seulement (*Ciphertext-only attack)*, où le cryptanalyste ne connaît qu'un ensemble de textes chiffrés ; il peut soit retrouver seulement les textes en clair, soit retrouver la clef. Deux types d'attaques appartiennent à cette classe :
  - L'attaque en force (*Brute force attack* ou *Exhaustive key search attack*) : Le cryptanalyste essaie toutes les combinaisons de clés possibles jusqu'à l'obtention du texte clair. Avec des ordinateurs de plus en plus performants et avec les nouvelles méthodes de calculs distribués, l'attaque en force reste encore un moyen de cryptanalyse très efficace, malgré les nouvelles longueurs de clefs.
  - L'attaque par l'analyse statistique (*Statistical analysis attack*) : En pratique, le cryptanalyste possède souvent des informations sur les statistiques du texte en clair (fréquences des lettres, format ASCII, présence d'un mot particulier, en-têtes de fichiers…), ce qui lui permet de décrypter quelques parties du texte en clair.
- L'attaque à texte en clair connu (*Known-plaintext attack)*, où le cryptanalyste connaît non seulement les textes chiffrés, mais aussi les textes en clair correspondants ; son but est alors de retrouver la clef. Du fait de la présence, dans la plupart des textes chiffrés, de parties connues (en-têtes de paquets, champs communs à tous les fichiers d'un type donné,...), ce type d'attaques est le plus courant.
- L'attaque à texte en clair choisi (*Chosen-plaintext attack)*, où le cryptanalyste peut, de plus, choisir des textes en clair à chiffrer et donc utiliser des textes apportant plus d'informations sur la clef. Si le cryptanalyste peut de plus adapter ses choix en fonction des textes chiffrés précédents, on parle d'attaque adaptative.
- L'attaque à texte chiffré choisi, qui est l'inverse de la précédente : le cryptanalyste peut choisir des textes chiffrés pour lesquels il connaîtra le texte en clair correspondant; sa tâche est alors de retrouver la clef. Ce type d'attaques est principalement utilisé contre les systèmes à clef publique, pour retrouver la clef privée.

VI.3 Fiabilité des systèmes cryptographiques

VI.3.1 Problème des clefs

Pour que le système soit fiable, il est nécessaire que les clés de chiffrement utilisées soient suffisamment sûres. Avec les protocoles actuels, la sûreté d'une

clé dépend de sa longueur. Cependant, plus la clé est longue, plus la transaction ou la communication va être lente. Il existe donc un compromis entre sécurité, et rapidité. Enfin, pour déchiffrer un document sans posséder la clé, il est nécessaire de disposer d'ordinateurs dont la puissance de calcul est très élevée pour casser le protocole. La "dépense" nécessaire pour casser le protocole doit donc être disproportionnée par rapport à la valeur de l'information protégée. Aujourd'hui, une clef de longueur 1024 bits (longueur typiquement utilisée pour le protocole RSA), nécessiterait plusieurs milliards d'années de calcul pour être cassée. Cependant, ce système dépend de l'évolution de la technique. Un algorithme jugé incassable aujourd'hui ne le sera peut-être plus dans quelques années. Le Challenge RSA 5, lancé en 1997, et qui consistait à casser par force brute un message chiffré par un algorithme RC5 à clef de 64 bits, a été remporté en 2002 par le projet Distributed.net [Web12]. L'opération aura duré cinq années, mais elle prouve que des clefs de 64 bits peuvent être cassées et qui plus est, par des entreprises non étatiques. Le projet Distributed.net fédérait des milliers d'internautes, afin d'utiliser le temps libre de leur ordinateur pour tester la totalité des clefs possibles. C'est le 14 juillet 2002 que le PIII-450 d'un internaute de Tokyo a renvoyé la clef. Celle-ci était "0x63DE7DC154F4D039" et produisait le texte clair suivant : « The unknown message is : some things are better left unread ».

On estime, aujourd'hui, que la longueur de clef d'un protocole symétrique ne doit jamais descendre en dessous de 90 bits, pour que le chiffrement reste sûr. Les fonctions de chiffrement sont supposées rendre impossible le décryptage d'un message clair sans la clef. A fortiori, elles doivent protéger le secret des clefs.

VI.3.2 Principe de Kerckhoffs [Sti03]

Une fonction de chiffrement E est une fonction bijective qui transforme un message clair m en un message chiffré c = E(m). Si un attaquant connaît c, il doit être très difficile de retrouver m (chiffrement partiellement cassé) ou E(**.**) (chiffrement totalement cassé). De plus, E est une fonction paramétrée par une clé secrète k appartenant à l'espace K des clés possibles.

Etant donné que l'ensemble des fonctions bijectives de {0,1} vers {0,1} est de cardinal 2N!, l'ignorance de l'adversaire est quantifiée par $\log_2(2N!)$ bits. Cette ignorance est bien sûr utopique, car quelque soient les précautions prises, en pratique, l'attaquant finit par acquérir une certaine quantité d'informations sur la communication chiffrée, ce qui lui permet de monter des attaques. Avec l'hypothèse de la seule ignorance de la clef, casser le chiffrement, revient à tester les |K| clefs possibles ou *attaque par force brute*. L'ignorance du cryptanalyste est réduite à $\log_2(|K|)$ bits où |K| est la taille de la clef, ce qui est plus réaliste.

D'une manière générale, on suppose toujours que le cryptanalyste connaît le détail des algorithmes, fonctions mathématiques ou protocoles employés dans un crypto-système. Même si ce n'est pas toujours le cas en pratique, il serait risqué de se baser sur le secret des mécanismes utilisés pour assurer la sécurité d'un système.

En 1883 dèjà, A. Kerckhoffs, présente une série de principes élémentaires devant être suivis pour assurer la sécurité d'un crypto-système. On ne retient aujourd'hui que le suivant : « toute méthode de chiffrement doit être supposée connue de l'adversaire, la sécurité du système ne doit reposer que sur le secret de la clef ».
Le principe de Kerckhoffs est une heuristique qui se défend par deux arguments :

- Il existe des algorithmes cryptographiques publics qui n'ont pas été cassés à ce jour, par exemple, RSA et DES.
- Il existe des algorithmes propriétaires (i.e. qui violent le principe de Kerckhoffs) qui ont été cassés.

L'exemple le plus illustratif est certainement la machine Enigma pendant la seconde guerre mondiale.

VI.3.3 Sécurité inconditionnelle

La sécurité inconditionnelle implique que la connaissance du message chiffré n'apporte aucune information sur le message en clair. Dans ce cas, la seule attaque possible est la recherche exhaustive de la clef secrète. Une cryptanalyse possible est d'utiliser la répétition de la clef et de rechercher des motifs dans le texte chiffré. Ceci montre que clef secrète doit être au moins aussi longue que le texte en clair pour obtenir une sécurité inconditionnelle. Le seul système de chiffrement qualifié de *inconditionnellement sûr* est le chiffre de Vernam(1917) qui utilise justement une clef aussi longue que le texte clair. C'est pour cette raison qu'on le désigne par « chiffrement par masque jetable » (One time pad). Il est basé sur la relation suivante :

$$\forall M, K / |M| = |K| : (M \oplus K) \oplus K = M$$

Dans ce cas la fonction de chiffrement est : $E_K(M) = M \oplus K$

et la fonction de déchiffrement est : $D_K(C) = C \oplus K$

Tous les autres systèmes sont théoriquement cassables.

En pratique, on utilise des systèmes de chiffrement *pratiquement sûrs* c'est-à-dire des systèmes pour lesquels un message chiffré ne permet de retrouver ni la clef secrète ni le message clair en un temps humainement raisonnable. La question n'est plus de savoir si l'attaquant trouvera la clef, mais quand il la trouvera. Par exemple, pour un ordinateur de 1Ghz, c'est-à-dire à 10^9 opérations par seconde et une clef de 128 bits, il y a 2^{128} soit environ $3{,}4 \times 10^{38}$ possibilités de clefs à tester, ce qui donne $3{,}4 \times 10^{31}$ secondes. A titre de comparaison, l'âge de l'univers est estimé à 15 milliard x 365 x 24 x 3600 = $4{,}7 \times 10^{17}$secondes.

Conclusion

Ce chapitre fait une incursion dans le domaine très vaste de la cryptographie. Le lien commun entre ce domaine et celui du marquage numérique est l'étude de la sécurité. Il nous a semblé utile de cerner d'abord les notions cryptographiques de base pour ne pas tomber dans le piège de refaire par marquage des fonctionnalités que la cryptographie propose déjà. Il y a derrière cette idée un certain principe de précaution: le marquage est une science jeune comparée à la cryptographie aux primitives déjà éprouvées. Si le marquage est utile, c'est en complément de la cryptographie, comme nous le proposerons dans une de nos contributions. Ce chapitre montre également qu'il existe en sécurité de grands principes dont le plus classique est celui de Kerckoffs. Cette partie sert donc d'introduction par l'exemple au thème du chapitre suivant.

Chapitre 3

La dissimulation de données

Introduction

L'homme a toujours ressenti le besoin de dissimuler des informations, bien avant même l'apparition des premiers ordinateurs et de machines à calculer. Avec la création du réseau Internet, qui s'est rapidement imposé comme outil essentiel de communication, ce besoin s'est encore accru. En effet, la communication sur Internet pose de plus en plus des problèmes stratégiques et sécuritaires pour beaucoup d'entités communicantes. Les transactions faites à travers le réseau peuvent être interceptées, modifiées ou volées. Avec les nouveaux formats de fichiers multimédia, les outils de sécurité traditionnels tels que la cryptographie s'avèrent souvent inefficaces. Ceci est particulièrement vrai pour le cas des images où il est devenu possible de réaliser des contrefaçons très élaborées, sans laisser aucune trace, en utilisant des outils de retouche très puissants tels que MS-Photoshop. Les outils de dissimulation de données sont des solutions potentielles dans ces cas-là et les dernières années ont vu beaucoup de chercheurs se pencher sur cette nouvelle approche de sécurité.

Le but de ce chapitre est de présenter les différentes techniques de dissimulation de données et de décrire les services de sécurité dans lesquels ils peuvent être implémentés, de même que les attaques auxquelles ils peuvent être confrontés. Bien que le procédé de dissimulation de données soit applicable à tout type de document numérique (texte, image ou son), on ne s'intéressera dans ce travail qu'au seul cas des images numériques.

I- La stéganographie

I.1 Définition introductive

Le mot stéganographie tire son origine d'une éthymologie grecque : steganos, signifiant caché et graphos, signifiant écriture, ce qui donne, littéralement, « écriture cachée ». La stéganographie est donc l'art de dissimuler un message secret au sein de données d'apparence anodine de façon à ce que sa présence soit imperceptible. La nature des données ne revêt pas d'importance : il peut s'agir d'un texte en clair ou de sa version chiffrée. Ce type de données est habituellement considéré comme inoffensif, incapable de contenir des informations autres que celles normalement prévues. De plus, le message n'a à priori aucun lien avec les données qui le véhiculent.
Alors qu'avec la cryptographie habituelle, la sécurité repose sur le fait que le message ne sera sans doute pas compris, avec la stéganographie, la sécurité repose sur le fait que le message ne sera sans doute pas détecté.

La stéganographie est une technique déjà très ancienne. Parmi les astuces historiques, on peut citer les encres invisibles à base de jus de citron, de lait ou de certains produits chimiques, les messages cachés dans des oeufs durs en écrivant sur la coquille à l'aide d'une solution de vinaigre et d'alun, les minuscules trous d'épingle dans des caractères sélectionnés, les infimes changements dans l'écriture manuelle des caractères ou dans leur espacement, les micro textes photographiés et réduits à un point de moins d'un millimètre de diamètre puis déposés sur une lettre apparemment anodine) ... etc. Ces astuces ont sans cesse évolué, et on a vu au fur et à mesure du temps la naissance de nouveaux procédés toujours plus efficaces.

Avec l'avènement de l'ère numérique, des procédés plus perfectionnés encore sont rendus possibles par l'utilisation des ordinateurs, d'un côté, et les nouveaux formats de fichiers d'un autre côté, en particulier les images. Certains ont allégué que la stéganographie aurait joué un rôle dans la préparation des attentats du 11 septembre 2001 aux Etats-Unis. Les terroristes se seraient échangé divers messages et plans cachés dans des photos publiées dans des sites Internet peu moraux [Web10].

En bref, cette technologie est en pleine expansion actuellement, et petit à petit elle prend sa place dans le domaine de la sécurité. Si autrefois la stéganographie avait juste de l'intérêt pour l'armée, maintenant elle gagne une grande popularité auprès diverses classes de citoyens et ses applications se multiplient dans divers domaines. Il s'agit alors de stéganographie « moderne » dont une définition pourrait être :

« Dissimulation d'une information secrète dans un flux de données numériques, tel qu'un fichier texte, une image numérique ou du son ». Les données hôtes sont alors dénotées par stégo-medium.

I.2 Mode d'opération

Pour pouvoir communiquer de façon secrète, il faut d'abord pouvoir communiquer tout simplement. Afin de récupérer le message secret, le correspondant doit connaître un secret et/ou la technique pour extraire ce message du stégo-médium. Il est évident que ce message caché peut être lui même codé et/ou signé en utilisant des méthodes cryptographiques, la stéganographie n'étant plus, alors, que la dernière étape d'encodage.

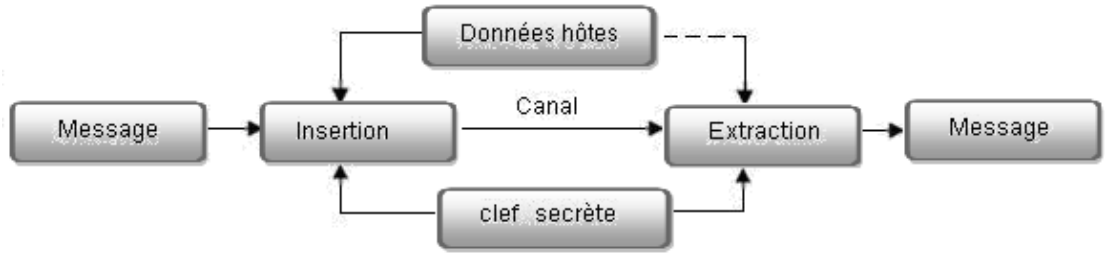


Figure 3.1 Schéma générique de la stéganographie

I.3 Les applications de la stéganographie

De nombreux usages peuvent exister dans des domaines très variés mais tous sensibles. On peut citer à titre d'exemple :

— De nos jours, on parle beaucoup de liberté d'expression et de la répression de l'usage de la cryptographie sur l'internet. Un argument utilisé par les états pour interdire la cryptographie est de dire que les citoyens honnêtes n'ont rien à cacher. Cette logique, appliquée par exemple au courrier, interdirait l'usage des enveloppes et n'autoriserait que les cartes postales. La stéganographie permettra alors de communiquer en toute liberté même là où l'utilisation de la cryptographie n'est normalement pas permise ou soulèverait des suspicions.

— Elle peut servir à publier des informations ouvertement mais à l'insu de tous, des informations, qui pourront ensuite être révélées et dont l'antériorité sera incontestable et vérifiable par tous.

— Elle peut être utilisée pour la dissimulation des données hautement confidentielles ou interdites au grand public, par exemple dans le domaine militaire ou médical;

— Elle peut être utilisée en espionnage industriel : toute entreprise a des secrets à protéger (information stratégique, formule chimique d'un nouveau produit, code source d'un logiciel propriétaire…). La stéganographie peut rendre le vol de ce genre d'information improbable voire impossible.

— la stéganographie peut être aussi utilisée à de mauvaises fins : un hacker peut tenter de dissimuler n'importe quel code malveillant dans un media amovible ou encore de dissimuler des fichiers MP3 propriétaires sur un serveur donné;

I.4 Mise en œuvre (Où et comment cacher l'information secrète)

Il existe autant d'endroits où cacher de l'information qu'il existe de formats et de types de données, les plus fréquemment utilisés étant les plus anodins. Bien qu'on ne s'intéresse ici qu'au cas des images, on peut citer quelques possibilités pour les autres formats.

I.4.1. Message transporté dans un texte

Le placement des ponctuations, l'introduction de variations orthographiques ou typographiques, le choix entre des synonymes ou des formes grammaticales, l'espacement entre les mots sont des façons simples d'ajouter de l'information sans perturber l'information originale.

D'autres techniques plus subtiles mais aussi plus délicates à mettre en œuvre, consistent à offrir d'autres clefs de lecture en n'utilisant que certaines lettres ou les lettres dans un certain ordre.

Pour les textes écrits**,** le fait de décaler une lettre de quelques pixels ne pose aucun problème sur une imprimante à laser et est pratiquement invisible à l'œil nu. En jouant sur les inter lettrages d'un texte très long et à raison de deux valeurs d'espacement correspondant à 1 et 0, il est possible de transmettre un message sous forme papier, qui ne révèlera son vrai sens qu'une fois analysé par un scanner ayant une bonne précision.

I.4.2 Message transporté dans une image

La stéganographie d'image numérique exploite les limites du système visuel humain (SVH). Ce dernier a une très basse sensibilité aux petits changements dans la luminance, et par conséquent, les variations dans les basses fréquences de l'image peuvent être utilisées pour cacher une grande quantité d'information. D'autres détails du codage de l'image, telle la palette de couleur, peuvent également être utilisés pour cacher des informations.

Typiquement les images numériques sont stockées dans des fichiers à raison de 24 bits/pixel ou de 8 bits/pixel. Evidement, une image de 24 bits fournit plus d'espace pour cacher les informations mais, elles sont généralement très volumineuses (1024 sur 768 pixels) et ne sont transmises sur un réseau (Internet) qu'au format compressé. Dans ce cas, c'est dans les choix du niveau de compression que se fera l'insertion. Les approches d'insertion les plus fréquentes sont les suivantes:

- **Usage des bits de poids faible d'une image**

La technique de base dite LSB (pour Least Significant Bit), consiste à modifier les bits de poids faible des pixels codant l'image pour contenir les bits du message secret [FG99]. une image numérique est une suite de pixels dont on code la couleur a l'aide d'un triplet d'octets; par exemple pour une couleur RGB sur 24 bits, chaque octet indique l'intensité de la couleur rouge, vert ou bleu (Red, Green, Blue) par un niveau parmi 256. Passer d'un niveau *n* au niveau immédiatement supérieur (*n+1*) ou inférieur (*n-1*) ne modifie que très peu la teinte du pixel, or c'est ce que l'on fait en modifiant le bit de poids faible de l'octet.

Cette technique très basique, s'applique tout particulièrement au format d'image BMP, format sans compression destructive, avec codage des pixels entrelacé sur 3 octets comme énoncé ci-dessus. Réciproquement, tout procédé de compression-décompression d'images avec pertes tel que le format JPEG est susceptible de détruire un message stéganographié de cette façon. On parle alors de *stérilisation*.

- **Manipulation de la palette de couleurs d'une image**

Certains formats graphiques tel que GIF ou PNG permettent le stockage des couleurs de l'image par référence à une palette de couleurs insérée dans le même fichier. Pour une image contenant 256 couleurs dans sa palette, il y a factoriel 256 façons différentes de stocker cette image. En utilisant un code connu entre l'émetteur et le récepteur de l'image (une clef), on peut donc communiquer un message de petite taille caché dans la permutation des couleurs dans la palette de l'image. Les deux images sont visuellement identiques, mais le stockage de celles-ci est différent.

- **Message caché dans le domaine de compression d'une image**

L'idée dans cette technique n'est pas de cacher une information dans les couleurs ou dans la palette mais dans les choix de compression. On distingue les algorithmes de compression conservative (lossless) et les algorithmes de compression non conservative (lossy).

- Avec des algorithmes de compression conservative tel que Zip ou Gzip, on peut choisir la puissance de compression. En consommant plus de temps calcul et/ou plus de mémoire pour les opérations intermédiaires, on peut obtenir de meilleurs résultats de compression. Ainsi deux fichiers compressés de tailles différentes peuvent être décompressés en deux fichiers identiques.

- Avec la compression non conservative comme le format JPEG (joint photographic Experts Group) on peut compresser des images à nuance continue à moins de 10% de leur taille originale , tout en perdant un peu de qualité. Le format JPEG est à double étape de compression. La première consiste à découper l'image en bloc de 8 fois 8 pixels et de transformer ce bloc par la transformée du cosinus discrète (DCT). Parmi les 64 coefficients de la DCT de chaque bloc, on n'en considère que 8 choisis parmi les fréquences moyennes. Trois de ces coefficients sont choisis avant insertion, et pour coder un "1" on leur rajoute une certaine valeur, et on leur enlève cette même valeur

pour coder un "0". Une fois tous les blocs compressés, il faut coder les coefficients obtenus en consommant le moins d'espace possible. Cette deuxième compression n'introduit pas de perte et est similaire dans les principes à ce que l'on peut retrouver dans Zip ou Gzip. Là encore, on peut introduire des bits d'informations supplémentaires.

I.4.3 Message transporté dans du son

Dans les formats sonores, il existe à peu près les mêmes possibilités de cacher des messages que dans les images. Le bruit de fond par exemple, constitué de faibles variations dans les basses fréquences, imperceptible pour l'oreille, peut être utilisé pour cacher une grande quantité d'information. Afin de rester indécelable, le bruit artificiel doit posséder les propriétés statistiques d'un vrai bruit de fond. Dans ce contexte, et sans entrer dans les détails, on peut distinguer quatre méthodes pour dissimuler les données : le codage du bit du poids faible, phase du codage, le spectre dispersé et la dissimulation des données par l'écho.

Dans un fichier sonore au format MIDI, par analogie à la permutation de la palette de couleurs, il est possible de permuter les différentes pistes.

Dans un fichier sonore avec compression sans perte, on peut cacher de l'information dans des variations imperceptibles du son, les bits faiblement significatifs.

Dans un fichier sonore avec compression avec perte, on peut cacher de l'information dans les choix de compression.

I.5 La stéganalyse

Alors que l'objectif de la stéganographie est l'action d'éviter le soupçon des messages cachés au sein d'autres données, le but de la stéganalyse est de découvrir et rendre inutilisables de tels messages cachés.

Le problème ici est différent de celui de la cryptanalyse : Le cryptanalyste dispose du message chiffré qu'il veut décrypter, alors que le stéganalyste est un observateur qui soupçonne une communication cachée et qui veut l'empêcher. Le message est inconnu, et la technique utilisée l'est aussi. Pour compliquer la situation, souvent le message est chiffré avant d'être dissimulé. Pourtant des moyens sérieux existent : Pour interdire toute communication cachée il faut pouvoir intercepter et transformer toutes les communications (car elle peuvent potentiellement servir de transport). Un firewall possède les propriétés appropriées pour ce genre de contrôle absolu des communications. Celui qui veut empêcher une communication cachée se doit néanmoins de laisser passer le message de couverture tout en détruisant le message caché. Il peut aussi tout simplement détruire tout message suspect :

- L'observateur peut comparer les propriétés statistiques de la communication qu'il soupçonne et les comparer avec celles d'une communication ne contenant

pas de messages cachés. De trop grandes différences peuvent être l'indice d'une communication cachée.

- Puisque le bruit de fond ou les basses fréquences sont des bonnes cachettes, il faudra y rajouter son propre bruit, ou filtrer le bruit existant. Il s'agit donc d'une sorte de dégradation du message. On pourra aussi décoder/décompresser puis coder/compresser le message pour le débarrasser des cachettes utilisant les particularités de certain codage.

- Un logiciel de stéganographie laisse sur le stégo-médium une série d'octets caractéristiques ou *signature*. Parfois, le simple fait d'utiliser un format d'image particulier peut être une signature.

- En pratique, la stéganographie est couplée avec la cryptographie afin de rendre le message inintelligible s'il est découvert. Le principal problème est que lorsque l'on chiffre un fichier avec de la cryptographie forte, le flux de données se présente comme un flux aléatoire d'octets. Et curieusement, un flux de données aléatoires est rare dans le domaine de l'informatique. Donc, un flux d'octets aléatoires apparaissant soudainement devient détectable.

En conclusion, la propriété la plus importante dans tout système stéganographique est l'indétectabilité par analyse statistique. En d'autres termes, il doit être difficile pour un attaquant de distinguer entre le médium original et le stégo-médium. A ce jour, une lutte efficace contre la stéganographie reste difficile à mettre en œuvre même s'il existe des pistes intéressantes.

II- Le marquage numérique

II.1. Définition introductive

Le marquage numérique (ou encore tatouage, filigranage, watermarking …) consiste à inclure dans un document numérique (image, son, vidéo. . .), une marque indélébile, typiquement quelques bits d'information, pouvant être de différentes natures et destinée à faire plus tard des assertions sur le document. Elle pourra, par exemple, identifier le propriétaire et/ou authentifier le document ou encore à renseigner sur les permissions attachées au document. L'extraction/détection de la marque ne pourra se faire que dans certaines conditions précises déterminées par le propriétaire ou l'expéditeur du document [DR99].

Il est important que d'un côté l'extraction ou la suppression de cette information du support soit difficile voire impossible (on parle de *robustesse*), et d'un autre côté que la distorsion introduite par la marque soit *imperceptible*, c'est-à-dire que la déformation doit être suffisamment faible pour que l'utilisateur ne puisse pas différencier le document marqué de l'original. Cette notion d'imperceptibilité et d'insertion dans la trame même du document rejoint la traduction littérale du terme "*digital watermark*" ou "filigrane électronique". Ceci n'est pas sans rappeler ce qui se fait avec les billets de banque, où les fibres sont marquées au moment de la sortie du bain d'eau, ce qui est à l'origine du terme

anglais *water-mark*. De la même manière que sur un billet de banque, le filigrane électronique est d'abord invisible et n'est révélé que par une transformation spécifique. L'intérêt d'une telle opération est que le marquage est indépendant du format de stockage des données, puisqu'il est intrinsèque au document.

Le document marqué est destiné à être distribué à grande échelle, il est donc amené à subir des déformations. Celles-ci peuvent être involontaires (par exemple : compression d'une image au format JPEG, puis décompression) ou volontaires (pirate voulant endommager le marquage). La **robustesse** à de telles attaques est l'une des propriétés importantes d'une méthode de marquage.

La troisième contrainte importante du marquage est la quantité d'information que l'on peut insérer, ou **capacité** : pour une fiabilité de détection donnée, plus l'on insère d'information, plus la déformation est importante. On doit donc trouver un compromis entre trois objectifs antagonistes : imperceptibilité, robustesse et capacité. C'est le cadre applicatif qui va déterminer l'importance relative de ces contraintes entre elles. Intuitivement, on sent bien que plus on cachera d'information, moins elle sera robuste, et inversement. En général, c'est la contrainte d'imperceptibilité qui est la plus forte : on ne veut pas que le marquage dégrade le contenu au point qu'il devienne inutilisable.

La **sécurité**, au sens cryptographique du terme, de la méthode de marquage, constitue une quatrième contrainte indépendante des trois premières. Elle concerne par exemple la génération de la clé secrète, ainsi que le protocole d'échange général. La méthode de marquage doit également respecter le principe énoncé par Kerckhoffs : l'algorithme lui-même doit pouvoir être rendu public, la sécurité ne dépendant pas de son caractère secret.

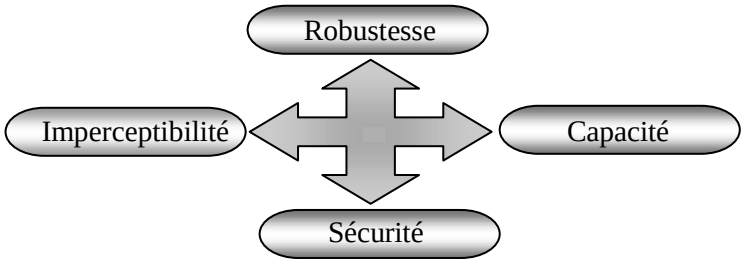


Figure 3.2 Interaction des contraintes de marquage

Il faut aussi noter la différence entre **détection** et **extraction** de la marque :
– La détection cherche à répondre à la question : *Le contenu contient-il une marque?*
– L'extraction cherche à répondre à la question : *Quelle est la marque que l'on a insérée dans le contenu ?*

Le but initial du marquage numérique était de décourager la copie et la distribution illicite de matériel numérique, ainsi que la protection de la propriété intellectuelle sur les données digitales, particulièrement l'image et l'audio, mais aujourd'hui, beaucoup d'autres champs d'application sont apparus, dans différents domaines (vidéo, code source de logiciels, bases de données …).

Dans ce travail, nous nous intéresserons uniquement au marquage d'image que nous développerons dans ce qui suit.

II.2 Mode de fonctionnement

Le marquage numérique d'images consiste en deux étapes distinctes : L'insertion de la marque par des procédés plus ou moins sophistiqués (dans le domaine spatial, dans le domaine fréquentiel, par étalement de spectre …) et l'extraction de la marque par des procédés généralement inverses. La figure 3.3 résume le schéma générique du marquage numérique.

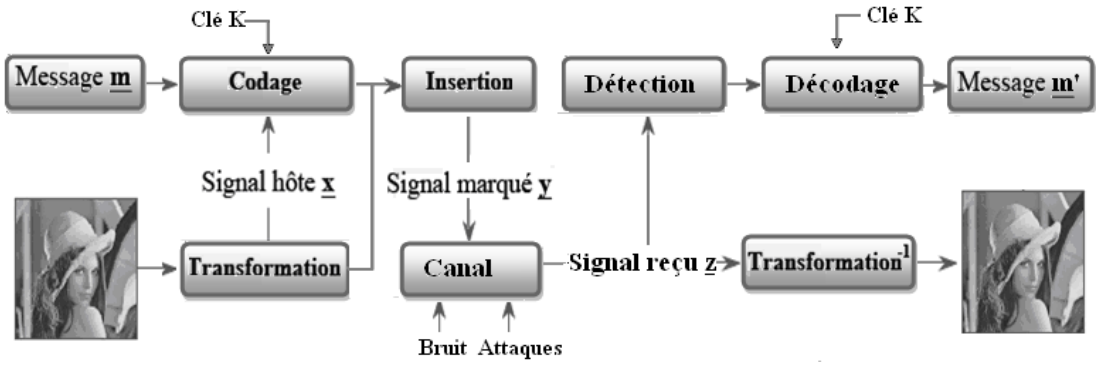


Figure 3.3 Schéma générique du marquage numérique

Un message m contenant L bits d'information est transformé, selon une clé k, en une marque w qui est ensuite insérée dans le document x (appelé *document hôte*) pour donner un document marqué y. C'est la phase d'**insertion**.

Ici, w est exprimé sous la forme d'un bruit qui est ajouté au document, la déformation dépendant de la puissance du bruit. La clef k est secrète et spécifique au marqueur. y est ensuite copié et attaqué, ce qui est modélisé par la transmission dans un canal soumis à du bruit. Le document reçu est appelé z. La réception d'un document consiste en deux parties : d'une part, la détection de la marque et d'autre part, s'il est présent, son décodage. La phase de **détection** consiste à prouver la présence d'une marque dans z grâce à k. La phase de **décodage** consiste à calculer une estimation m' de m.

- Si le document original n'est pas utilisé à la réception, l'algorithme de marquage est qualifié d'**aveugle**. Dans le cas inverse (beaucoup moins intéressant en pratique), l'algorithme est qualifié de non aveugle ou à décodeur informé.

- Si le document original est utilisé dans la construction de w, on parlera de **marquage informé**.

- Lorsque plusieurs marques sont insérées (correspondant souvent à plusieurs utilisateurs), on parle de **marquage multiple**.

- Si le système de marquage permet d'extraire la marque il est dit de **Type I,** s'il se contente de la détecter et l'estimer à l'aide d'une mesure de confiance, il est dit de **Type II.**

- Si la clef ayant servi à l'insertion est disponible chez l'extracteur, le marquage est qualifié de privé, si l'extraction de la marque ne nécessite pas la connaissance d'un secret, le marquage est qualifié de asymétrique. Cela implique que tout le monde est capable de lire la ou les marques de l'image marquée sans pouvoir les effacer. Cela pourrait se faire par un marquage sans clef ou alors par un marquage avec clef privée et une extraction avec la clef publique correspondante, dans un schéma analogue à celui de la cryptographie asymétrique.

II.3 Applications du marquage

Le marquage numérique trouve aujourd'hui de nombreuses applications dans différents domaines, même si l'objectif initial était la **gestion des droits d'auteur** au niveau des œuvres numériques. Les applications commerciales commencent à voir le jour, particulièrement dans le domaine de l'image [Web6] et de l'audio. A titre d'exemple, Digimarc, firme pionnière, rassemble des brevets de base sur le marquage (notamment celui de l'estampillage, défini plus loin) dont elle vend la licence. Elle est également auteur du module de marquage du logiciel de traitement d'image Photoshop. Son concurrent, Verance, fournit les outils de contrôle de flux audiovisuel Broadcast Verification et Confirm Media [Web7]. La compagnie Liquid Audio fournit également un système de marquage audio. Le SDMI (Secure Digital Music Initiative) est un consortium de compagnies pour un projet de marquage audio. Les associations japonaises JASRAC et RIAS sont également actives dans ce domaine. Nextamp et MediaSec [Web9], filiales de Thomson, s'intéressent au marquage vidéo pour une application de suivi des transactions : le document téléchargé contient le nom de l'acheteur. L'institut Fraunhofer (créateurs du mp3) a annoncé en 2006 avoir développé un logiciel de marquage audio commercialisable [Web12]. Mais commençons par lister les différentes applications, dans les trois médias possibles, même si par la suite on ne s'intéressera qu'au seul cas des images.

- **Identification du propriétaire**

La marque insérée doit identifier indéniablement le propriétaire de l'œuvre numérique. Par exemple on peut marquer une image par un texte visible identifiant l'entreprise qui détient les droits commerciaux de l'image. Cependant, cette méthode qui présente déjà le désavantage de détériorer la qualité de l'image n'est pas sans faille : L'exemple le plus célèbre pour lequel la perte du texte sur l'image a causé beaucoup de problèmes à son propriétaire est la photographie de Lena que l'on peut voir sur la figure 3.4.

Figure 3.4 L'image Lena

C'est l'image la plus utilisée dans la recherche en traitement d'image. Elle est apparue dans d'innombrables articles de conférences et de journaux scientifiques sans que jamais aucune référence ne soit faite au propriétaire des droits de diffusion, à savoir : Playboy Entreprise, Inc. L'image a été scannée à partir d'un numéro du magazine, mais seule la partie haute de l'image a été conservée. La partie basse, contenant le copyright, a été tout simplement supprimée.
La marque contenant la preuve de propriété sera donc insérée, préférentiellement, de manière invisible.

- **Protection des droits d'auteurs**

Plus généralement, le marquage permet de protéger les droits d'auteur. La protection des droits vise à assurer que l'identifiant des ayant-droits sur le contenu sera toujours présent dans le contenu [DR99]. En effet, dans le monde analogique, un original se distingue d'une copie par sa meilleure qualité. Dans le monde numérique, copier un contenu nécessite seulement de dupliquer les symboles qui le composent. On peut donc faire des copies *parfaites* à l'*infini*. La notion d'*original* n'existe plus. La protection des droits numériques cherche à résoudre ce problème.
Certaines législations imposent de déposer les droits d'un document auprès d'un tiers de confiance, qui délivre ensuite un identifiant. C'est cet identifiant qui sera marqué dans le document, dans le but d'empêcher sa manipulation. Cet enjeu est très important**,** par exemple, pour les maisons de disques regroupées dans le consortium SDMI face à l'échange de mp3. La protection du droit d'auteur doit être envisagée sous trois points vus différents [Bas00]:

- Sans réseau de distribution : Un exemple typique de ce scénario survient lorsque, devant une cour de justice, un propriétaire doit prouver qu'une donnée numérique lui appartient face à un pirate qui aurait distribué celle-ci en se faisant passer pour le propriétaire.

- Avec un réseau de distribution, mais sans tiers de confiance. Dans ce cas le pirate cherchera à utiliser par exemple Internet pour se procurer la donnée numérique et ensuite supprimer la signature de l'œuvre marquée. Dans ce scénario, le propriétaire devra donc se prémunir des failles du réseau de distribution en plus de s'assurer de la robustesse de la marque insérée.

- Avec un réseau de distribution et avec un tiers de confiance. C'est la même situation que pour le deuxième scénario avec une troisième personne en plus qui peut déterminer la propriété légale de l'œuvre numérique en cas de conflit.

- **Contrôle de copie**

Le marquage est utilisé ainsi, au niveau du matériel, pour limiter la diffusion des données copiées. Dans ce scénario deux cas peuvent être envisagés :
- La marque est présente dans le document, donc le matériel fonctionne.
- La marque a disparu, donc la copie est illicite (moindre qualité du document) et le matériel refuse de fonctionner.

- **Accès conditionnel**

La marque correspond souvent à des informations qui donnent accès ou non (selon les droits que l'on a acquis à l'achat) à des fonctionnalités supplémentaires: ajout du son à une vidéo, possibilité de pouvoir enregistrer une vidéo, possibilité de pouvoir écouter un morceau en entier…

- **Traçabilité dans un système commercial**

Face à une situation de piratage, les ayant droits des données numériques piratées chercheront à mettre en place deux types de stratégies complémentaires :

- Rendre impossible la copie des données numériques en mettant en place des techniques cryptographiques ;

- Rechercher les pirates et les filières pirates pour les condamner, obtenir des frais de dédommagement face au préjudice et les empêcher de nuire.
Dans ce dernier cas, le marquage permet de retrouver la personne originelle qui a permis le piratage de l'oeuvre [Bas00]. Il faut alors insérer l'identité du vendeur et celle de l'acheteur chaque fois qu'une transaction est envisagée. Les propriétaires successifs du document, et donc les sources de copie d'un document, peuvent ainsi être identifiés. On parle d'**estampillage** (ou en anglais, fingerprinting). L'idée du fingerprinting est celle de l'empreinte digitale : pouvoir s'assurer que tel utilisateur n'a pas essayé d'enfreindre la bonne marche du protocole d'attribution des droits. Pour cela, on vient cacher dans chaque version du même contenu demandé par plusieurs utilisateurs, un identifiant propre à chacun de ces utilisateurs. Ainsi, si le contenu est retrouvé dans un circuit pirate, on est en mesure d'identifier l'origine de la fuite. En effet, le plus grand facteur de piratage vient actuellement du fait que les données informatiques (fichiers audios, vidéos ou autres) sont très facile à diffuser via, entre autres, les réseaux point-à-point (peer-to-peer, P2P). Il n'est pas rare de pouvoir trouver sur les systèmes de partage (GNutella, Kazaa, BitTorrent) des films américains avant leur diffusion dans les salles autres qu'américaines. Ces copies, appelées *screeners*, sont réalisées directement dans une salle de cinéma à partir d'un caméscope standard. Bien que la qualité de ces vidéos soit faible comparée à celle d'un film, l'impact économique est important. Afin d'inciter les salles de cinéma à empêcher de telles pratiques, une empreinte indiquant la date ainsi que la salle de diffusion permettrait d'identifier le cinéma "coupable" si une copie du film est disponible

sur les réseaux d'échanges. Il est évident que la faute n'incombe pas aux réseaux P2P mais aux personnes malhonnêtes mettant illégalement des fichiers à disposition d'autrui.

- **Contrôle et surveillance d'un flux de diffusion**

Cette application concerne la diffusion du contenu à travers les réseaux télévisés, radiophonique ou à travers Internet. Le marquage de documents peut s'avérer très utile dans ce cas puisqu'il permet de vérifier automatiquement si l'œuvre a été diffusée ou non sur différentes chaînes.

- **Authentification et contrôle d'intégrité**

C'est bien sûr l'application qui nous intéresse le plus dans ce travail. La protection de l'intégrité vise à s'assurer que le contenu reçu est conforme à l'état original. Dans ce cas, le marquage est volontairement vulnérable aux attaques (fragile) dans le but de détecter une manipulation éventuelle du document. C'est l'*absence* de la marque qui prouvera que le contenu est suspect. Plusieurs utilisations peuvent être envisagées :

- Justifier auprès d'un tribunal l'authenticité de documents tels que des enregistrements de caméras de surveillance (authentification de contenu).

- Le propriétaire d'une œuvre cherche à vérifier si le contenu de son œuvre a été modifié. Ce scénario se produit fréquemment dans le domaine militaire où il faut sans cesse vérifier que les informations reçues par des alliés n'ont pas été altérées par un adversaire. Dans le domaine médical, le marquage de document peut aussi s'avérer très utile. Par exemple, lors d'une opération dentaire, le dentiste est amené à prendre une radio de la dentition pour repérer les dents à soigner. Si celui-ci est fraudeur, il peut modifier la radio numérique afin de faire apparaître une fausse carie et justifier ainsi auprès de l'organisme de la sécurité sociale la nécessité d'opérer le patient et donc lui faire facturer cette opération coûteuse. L'organisme de Sécurité Sociale cherchera donc à marquer les radios numériques à partir de l'appareil qui les produit.

- L'utilisateur cherchera à s'assurer si l'œuvre qu'il s'est procuré est bien la version originale authentique et non pas une simple copie ou une version manipulée (par exemple, lorsqu'il s'agit de photos de presse).

L'intérêt d'une technique de marquage fragile dépend entre autres de la possibilité de localiser les zones de l'image manipulées, ou encore de déterminer le type de la manipulation effectuée. On parle alors de « marquage révélateur » (*tell-tale watermarking*).

- **Le marquage légiste** (*forensic watermarking*)

Ce marquage regroupe les applications qui peuvent directement entraîner l'intervention des tribunaux : authentification, marquage fragile pour authentification d'un témoignage ou de la validité d'un chèque. Il inclut notamment un scénario proche de l'estampillage, dans lequel on autorise les

copies d'un document, tout en pouvant remonter à la source du piratage. Le but est alors d'attaquer le pirate en justice, à titre dissuasif pour les autres utilisateurs. Ce scénario est souvent évoqué pour un contenu musical ou pour le cinéma en ligne.

- **Contenus auto-indexés**

L'indexation vise à décrire la sémantique d'un contenu [Web7]. Elle se manifeste sous la forme d'une série d'annotations à l'image pour la classer au sein d'autres images par exemple. Ces méta-données peuvent éventuellement trouver leur place dans un format approprié réservant dans l'en-tête du fichier la place nécessaire pour les stocker. Un problème se pose au transcodage : lorsque l'on change le format de représentation du contenu, ces méta-données peuvent disparaître, si le format de destination ne tient pas compte de ces méta-données. Par contre, si l'on insère ces méta-données par marquage, avec une robustesse limitée destinée à résister uniquement au changement de format, alors on permet une gestion plus souple des contenus en ne privilégiant pas un standard de représentation plus qu'un autre. On parle donc de **document auto-indexé** lorsque la marque contient sa propre description, afin de permettre par exemple son stockage dans une base de données sans problème de changement de format.

- **Amélioration de contenu**

Enfin, un marquage peut servir à insérer une information supplémentaire dans le document, servant à renseigner sur ce dernier sans contrainte de sécurité ou de robustesse. On peut ainsi ajouter des informations sur l'interprète d'une chanson ou une traduction en langage des signes dans un document vidéo, ou être dirigé vers un site internet en scannant une publicité…

Les **contenus augmentés** servent donc à augmenter les possibilités offertes par les contenus, sans changer le système de diffusion en vigueur. Par exemple, en cachant une carte de profondeur dans une image classique, on permettra à l'utilisateur disposant du matériel nécessaire de voir en 3D. Ceux qui ne disposent pas du matériel nécessaire pourront toujours visualiser l'image comme n'importe quelle autre image. Un autre exemple est celui du projet Artus [Web11], dans lequel on vient cacher dans le flux vidéo MPEG-2 les informations nécessaires à l'animation d'un avatar 3D en langue des signes, pour rendre intelligibles les programmes aux sourds ayant pu se procurer le décodeur adéquat.

II. 4 Caractérisation du schéma de marquage

Quelque soit le problème de marquage considéré, la marque cachée dans l'image doit remplir certaines conditions essentielles, dont les plus importantes sont l'imperceptibilité, la spécificité, la robustesse, et la résistance aux attaques.

- Elle doit être **imperceptible** au sens où l'image marquée doit être visuellement équivalente à l'image originale. Non seulement, il ne faut pas dénaturer l'image, mais en plus, si la marque est visible, elle pourra être facilement

éliminée. Il existe pourtant certains algorithmes de marquage qui autorisent la visibilité de la marque, comme le cas d'un logo passé en filigrane sur une image pour assurer le copyright, l'exigence principale du schéma de marquage étant alors, la robustesse.

Figure 3.5 Exemple de marquage visible

- Quoiqu' imperceptible, la marque doit être suffisamment **spécifique** pour être clairement identifiable lors de son extraction. Selon l'information qu'est censée coder la marque, le destinataire, le propriétaire, les droits associés à l'image doivent être clairement identifiés, et une personne non autorisée ne doit pas pouvoir générer de telles marques. Une marque trop peu perceptible serait peu robuste et, plus grave, pourrait être détectée à tort. Dans le cas où la technique de marquage est censée conduire à l'élaboration de preuves légales, il faut que les marques soient assez spécifiques pour ne jamais condamner un innocent.

- Un problème de marquage peut être un problème de copyright : son but est alors de permettre aux propriétaires de l'image de prouver leurs droits sur cette image et également de repérer d'où proviennent les copies illégales de cette image. Dans ce cas la marque doit être **robuste**. En effet, l'image pourra être modifiée; elle peut être attaquée, dans le but de corrompre ou de supprimer la marque, mais des opérations usuelles telles que la compression, l'impression, la scannérisation la dégradent également : elles suppriment des informations dans les zones les plus lisses de l'image, la sous échantillonnent, modifient sa taille, etc ... La marque doit être suffisamment résistante pour rester décelable tant que la dégradation du medium par ces transformations naturelles reste peu signifiante.

- Plus que des transformations naturelles, le medium va subir l'**attaque** de pirates voulant effacer cette marque. Les attaques les plus simples (légère rotation ou translation d'une image, rognage de quelques lignes ou colonnes) obtiennent déjà des résultats dévastateurs sur les méthodes initialement imaginées [FGM00], et les chercheurs ont mis en évidence des attaques beaucoup plus perfectionnées [CFF05]. Les pirates sont supposés, d'après les lois de Kerckhoffs [Sta03], connaître l'algorithme de marquage et donc être en mesure de développer des attaques spécifiques à cet algorithme. Là encore, il faut que la destruction de la marque ne puisse se faire sans détérioration

significative du medium. Il faut aussi, dans le cas où le marquage s'est fait grâce à la connaissance d'un secret, que les pirates ne puissent avoir accès à ce secret par l'étude du medium.

- La quantité d'information insérée dans le marquage, ou **charge utile** (*payload*), est très variable selon les algorithmes et applications proposés. Pour l'insertion d'un copyright, on peut par exemple vouloir insérer une information similaire à la norme ISBN utilisée pour les livres, soit de 60 à 70 bits d'information [LSL00]. Un message encore plus long peut être inséré si l'image est considérée comme un canal de communication caché. Une charge utile de 1000 bits ou plus est parfois envisagée [MDC04]. A l'inverse, beaucoup d'auteurs proposent d'insérer une information binaire (présence du marquage ou non), pour une application à la protection de copie.

- Un algorithme de marquage d'images peut aussi avoir pour but de repérer toute modification non autorisée de l'image. La marque que l'on va alors inclure dans l'image devra, au contraire du cas précédent, être **fragile**, de sorte que n'importe quel changement de l'image entraîne une modification de la marque et soit donc détecté.

- Un cas intermédiaire est celui des schémas **semi-fragiles.** Ces derniers combinent à la fois les propriétés du marquage robuste et fragile. Comme les robustes, ils tolèrent certains changements de l'image, comme des rotations, translations ou addition de bruit et comme les marquages fragiles, ils sont capables de déterminer les régions où l'image a été brutalement modifiée et celles ou elle reste authentique. Par conséquent les marquages semi fragile arrivent à différencier les changements légers, comme l'ajout d'un bruit, à des changements destructeurs, comme la compression JPEG. C'est dans cette problématique que nous nous placerons dans nos travaux.

- D'autres champs d'intérêt englobent le **marquage sans perte**, ou marquage réversible [FG01], où l'on désire pouvoir récupérer de façon exacte le document initial. Ce marquage intéresse particulièrement des domaines comme la médecine et la défense, et nous aurons l'occasion de l'étudier en détail puisqu'il est à la base de nos travaux.

- Beaucoup de schémas de marquage d'images sont inspirées des méthodes usuelles de codage et de compression. Dans ces méthodes, les informations sont souvent ajoutées dans le domaine **transformé** de l'image (DCT : Discrete Cosine Transform et DFT : Discrete Fourier Transform), contrairement à d'autres techniques où la marque est insérée dans le domaine **spatial** modifiant directement les valeurs des pixels. Comme aucun traitement initial n'est requis, ces algorithmes sont très rapides et permettent de travailler en temps réel. De plus, elles offrent souvent une bonne résistance aux opérations géométriques. Les algorithmes fonctionnant avec la DCT ne sont pas très résistants aux transformations géométriques car celles-ci affectent grandement les coefficients de la DCT. Au contraire, l'espace de Fourier possède des

propriétés d'invariance aux translations et rotations qui augmentent la fiabilité de la méthode. D'autres domaines de travail peuvent également être observés, tels le domaine multi-résolutions et le marquage fractal. Chaque espace de travail possède ses propres avantages et inconvénients.

- On peut aussi discerner entre les schémas de marquage selon la manière d'inclure la marque :

  - Les schémas **additifs** : Lors de l'insertion, le signal représentant la marque est ajouté à certaines composantes du médium. Pour y parvenir, il s'agit d'adapter la marque au médium, afin que le signal qu'elle représente ne soit ni trop faible (risques de non détectabilité et problèmes de robustesse), ni trop fort (effacement du signal initial, et donc trop grande dégradation de celui-ci). La génération de la marque se fait généralement par étalement de spectre (voir paragraphe II.5.3).

  - Les schémas **substitutifs** : La différence majeure entre les schémas additifs et substitutifs provient de la phase d'insertion. Dans ces derniers, au lieu d'ajouter un signal au médium, certaines composantes sont remplacées afin que le stégo-médium exhibe une propriété caractéristique. Lors de la phase de détection, si cette particularité est présente, le médium est considéré comme marqué avec une marque donnée.

II .5 Quelques algorithmes de marquage

Les algorithmes de marquage se distinguent les uns des autres essentiellement par les quatre points clés suivants :

- La manière de sélectionner des points (ou des blocs) dans le document hôte qui porteront l'information cachée;

- Le choix d'un espace de travail pour réaliser l'opération de dissimulation (dans le domaine spatial, ou transformé comme la DCT, les Ondelettes ou Fourier-Melin);

- La stratégie utilisée pour mettre en forme l'information à cacher avant son enfouissement (redondance, code correcteur);

- La manière de mélanger intimement le message avec le signal hôte (modulation); l'idée de base consiste le plus souvent à imposer une relation binaire entre les bits du message et des caractéristiques choisies de l'image porteuse;

II.5.1 Domaine spatial

- **Modification des bits de poids faible : LSB**

Pour s'assurer de l'invisibilité de la marque, les premiers algorithmes allaient inscrire la marque dans les bits de poids faible de la luminance de l'image. Cette marque est très facile à modifier ou à enlever. En effet, une étude statistique

des bits de poids faible de l'image renseigne le pirate sur l'existence du marquage qu'il peut ensuite enlever à loisir.
On peut décider de cacher les emplacements des bits marqués avec une clef secrète, ce qui empêchera l'écriture d'une autre marque à la place. Par contre, la marque ne résistera pas à une compression jpeg où à un bruit blanc gaussien additif.

- **Technique du "Patchwork"**

Pour donner à la technique précédente un peu de solidité, on peut décider de répéter un grand nombre de fois le même bit pour qu'une étude statistique nous donne le bit marqué. Le raffinement proposé par la technique du patchwork est double :

1. Pour chaque bit forcé à 1 par le marquage, on force un autre bit à 0. Ainsi, les propriétés globales statistiques de l'image sont inchangées;

2. De plus, pour rendre cette marque invisible localement, on utilise une clef qui va coder l'emplacement des bits à 0 et des bits à 1. L'extraction de la marque se fait alors par un calcul de la somme des différences entre les positions des bits donnés par la clef.

Cette technique est plus robuste que la première; en effet, l'ajout d'une forte redondance permet de compenser les effets du bruit blanc additif. Toutefois, ce marquage ne résiste pas à de petites déformations géométriques, ni même à la compression JPEG.

Il existe d'autres approches dans le domaine spatial, qui consistent à moduler l'information dans la composante bleue à différents endroits de l'image, créant une pseudo modification proportionnelle à la luminance, ou encore le fait de découper l'image en différents blocs de taille variable, où l'information est stockée dans la moyenne des valeurs des pixels de ces blocs. Néanmoins, les marquages dans le domaine spatial résistent très mal aux attaques de type géométrique. Le simple fait d'appliquer une rotation corrompt la plupart du temps la marque.

II.5.2 Domaine fréquentiel

- **Algorithme de Koch et Zhao**
  Constatant la mauvaise performance des algorithmes de marquage dans le domaine spatial vis-à-vis de la compression JPEG, Koch et Zhao [KZ95] ont proposé un algorithme se fondant sur le marquage dans le domaine fréquentiel. L'idée de base est d'extraire un certain nombre de carrés de 8x8 pixels de l'image, de calculer la transformée discrète en cosinus (DCT) de ces blocs et d'aller marquer un bit sur les moyennes fréquences correspondantes, sachant que la modification des basses fréquences de l'image la changerait trop et que les hautes fréquences sont enlevées par la compression JPEG.

- **L'espace engendré par la transformée de Fourier-Mellin** : Les transformations géométriques de l'image marquée conduisent fréquemment à l'impossibilité d'extraire la marque pour de nombreux algorithmes. Ce constat a conduit à envisager l'implantation de la marque dans un espace transformé présentant une invariance aux opérations géométriques usuelles de l'image. Dans l'article [OP97], Ó Ruanaidh et al. préconisent l'usage de la transformée de Fourier-Mellin pour assurer la restitution de la marque après que l'image ait subi une translation et/ou une rotation et/ou un changement d'échelle. L'espace invariant est obtenu, d'une part grâce à la propriété de la transformée de Fourier qui répercute une translation de l'image exclusivement sur la phase et laisse invariant l'amplitude, et d'autre part, par un changement de repère, de cartésien vers logarithmique-polaire, ce qui ramène les opérations de rotation et de changement d'échelle à une translation.

- **Le domaine ondelettes** : les transformées en ondelettes qui, comme la transformée DCT, fait l'objet de nombreuses études dans le contexte du codage ont également trouvé un écho dans la communauté du marquage d'image [Wak02]. Cet intérêt repose d'une part sur les analyses en termes psychovisuels menées afin d'optimiser les tables de quantifications des codeurs, d'autre part sur l'aspect multi-échelle de telles transformées propice à une répartition plus robuste du marquage. Ce gain en robustesse apporté par l'usage d'une transformée ondelettes est particulièrement significatif si l'on considère les algorithmes de compression de type EZW (Embedded Zero-tree Wavelet) qui seront vraisemblablement intégrés dans la nouvelle norme de compression JPEG-2000.

II.5.3 Autres approches

- **Marquage par étalement de spectre**

Empruntée aux télécommunications militaires, la technique de l'étalement de spectre consiste à envoyer un message sur un grand spectre de fréquences de telle manière que, à toute fréquence, la puissance du signal émis soit inférieure à celle du bruit. Ainsi, localement, l'émission est toujours imperceptible et ce n'est qu'en écoutant sur l'ensemble du spectre d'émission et avec la connaissance du procédé utilisé que l'on pourra entendre le message émis.

Certains algorithmes de marquage d'images suivent ce principe. Un exemple est de sélectionner des blocs de bits, tous de la même taille, de se donner une suite pseudo-aléatoire de la taille des blocs et de rajouter à tout bit d'un bloc la marque "xor" le bit correspondant de la suite pseudo-aléatoire [CKLS97]. Cette technique d'étalement de spectre rappelle beaucoup celle du Patchwork. Elle perd toute sa robustesse face à des déformations géométriques de l'image.

- **Marquage fractal**

Une première technique de marquage fondée sur la compression fractale a été présentée par Puate et Jordan [PJ98] dés 1998. La compression fractale repose sur la détermination d'un IFS (Iterated Functions System) qui permet de représenter l'image comme un attracteur.

Un IFS est constitué d'un ensemble de fonctions $w_i : K \rightarrow K$, contractantes et définies sur un compact (K, d). L'application w qui, à une partie A de K, est contractante pour le métrique de Hausdorff, admet un unique point fixe, l'attracteur de l'IFS. Le principe de la compression fractale est de déterminer les w_i dont l'attracteur est l'image qu'on souhaite compresser et qui, pour simplifier le problème, sont des fonctions affines. On partitionne alors l'image en carrés R_i de taille n par n, appelés *range blocks*, et on recherche des carrés de taille égale ou différente, appelés *domain blocks*, transformables par les w_i en *range blocks* via une transformation spatiale et en niveaux de gris. En compression, les *domain blocks* sont en général plus gros que les *range blocks*, mais cette contrainte n'intervient plus dans le cadre du marquage.

Le procédé de marquage élaboré par Puate et Jordan [PJ98] associe à chaque *range block* deux ensembles de *domain blocks* possibles. La marque, composée sur un alphabet binaire, détermine l'ensemble dans lequel on doit chercher le *domain block* qui minimise l'erreur au sens des moindres carrés.

Une autre technique, proposée dans Bas, Chassery et Davoine [Bas02], met en oeuvre l'étude des similarités dans l'image. La méthode consiste à rechercher un IFS sur l'image pour y dissimuler de nouvelles similarités servant de marque. En forçant ainsi les similarités qui définissent la marque, on conserve le contrôle sur le code fractal (i.e. l'IFS). Pour la détection, il suffit alors de rechercher le *range block* qui minimise l'erreur au sens des moindres carrés pour un *domain block* donné et celui désiré.

II.6 Les Attaques sur le marquage numérique

Comme vu précédemment, un des points forts d'un marquage efficace réside dans sa robustesse. Néanmoins, certaines transformations basiques peuvent effacer la marque, ou du moins, potentiellement l'altérer, à côté d'autres tentatives délibérées émanant de pirates. Toutes ces transformations, volontaires ou involontaires, ayant une influence directe sur la marque, sont appelées des attaques.

Il existe deux grands types d'attaques sur les images marquées [CFF05] :

- Les attaques liées à l'image (ou au signal), dites *aveugles*, dont le but est clairement une suppression simple d'une potentielle donnée masquée dans l'image, en ignorant son contenu. Cela se résume à des transformations plus ou moins violentes. Ces transformations ont pour but de rendre illisible la marque.

- Les attaques plus malicieuses, dont le but et de retrouver la marque. Pour cela il suffit de récupérer par différents moyens la clef utilisée lors du marquage de l'image originale. On pourra alors modifier la marque, la lire, la supprimer, la recopier sur d'autres supports...

II.6.1 Les attaques basiques involontaires

II.6.1.1 Les transformations géométriques

- **Symétrie horizontale**

Le fait simplement d'inverser horizontalement une image est très souvent fatal à une grande catégorie de schémas de marquage. Cette transformation peut sembler au premier abord bien trop brutale pour conserver le sens d'une image, mais il peut passer inaperçu pour un paysage, ou même pour un film ou aucune scène d'écriture n'intervient. Un exemple est illustré à la figure 3.6.

Figure 3.6 Exemple de symétrie horizontale

- **Recadrage**

Ces transformations concernent surtout la mise en page de diverses images scannées. Cela peut être une simple rotation de quelque degrés, ou bien un découpage brutal d'une partie de l'image. Ces types de recadrage peuvent être des attaques très efficaces.

Figure 3.7 Exemple de découpage simple

- **Mise à l'échelle :** Le fait d'étirer horizontalement ou verticalement une image est souvent utilisé dans la mise en page également.

Figure 3.8 Exemple de Rotation (7°), Mise à l'échelle (120%) et découpage

- **Composition d'images, mosaïque :** Il s'agit d'utiliser un découpage d'une image d'une façon beaucoup plus violente et qui se prête assez bien aux pages HTML. Il suffit de découper l'image en autant de morceaux que l'on désire (plus il y a de morceaux plus l'attaque à des chances d'aboutir), puis de recoller ces morceaux au moment de l'affichage en créant, par exemple en HTML, un tableau dont chacune des cellules contiendra un morceau de l'image. Cette attaque est très peu applicable en pratique, et heureusement car elle est d'une rare efficacité si l'on se donne les moyens de bien découper l'image.

Figure 3.9 Exemple de mosaïque d'image

II.6.1.2 Les transformations fréquentielles

Ces transformations modifient essentiellement les coefficients de la DCT.

- Bruitage et Filtrage : Le bruit est une altération de l'image : toute l'information pertinente dans l'image n'est pas simplement accessible. Des exemples de bruit artificiel peuvent être le *bruit gaussien* qui consiste en ajouts successifs de valeurs générées aléatoirement à chaque pixel d'une image, ou encore le bruit *sel*

et poivre qui transforme aléatoirement des pixels de l'image en pixel noir ou blanc.

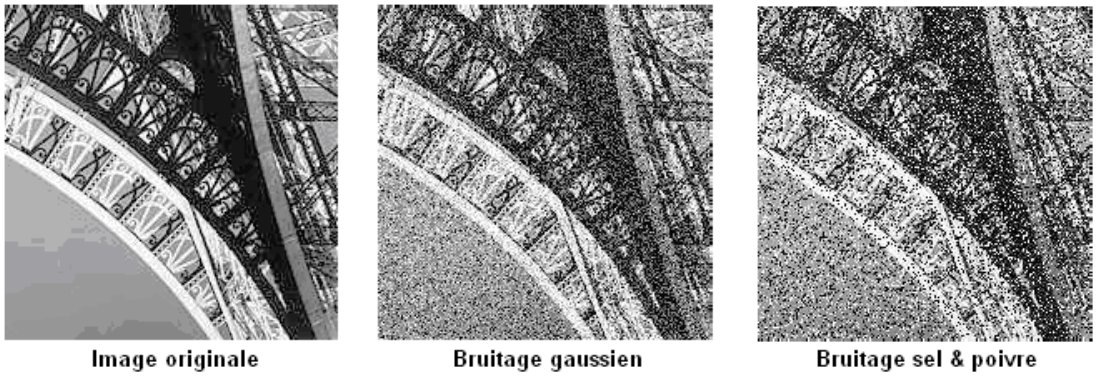


Figure 3.10 Exemple de bruitage d'une image

Pour récupérer l'information pertinente dans l'image on peut utiliser différents types de filtres :

Filtre passe-bas : Faisant partie de la catégorie des filtrages linéaires, il utilise la transformée de Fourier pour travailler dans l'espace des fréquences de l'image et dans lequel il ne laisse alors passer que les basses fréquences. En fait, il ne s'agit ni plus ni moins que d'un produit de convolution du signal avec une fonction passe bas.
Filtre passe-haut : Toujours dans les filtrages linéaires, et souvent appelé "*Sharpen"* par allusion au fait qu'il a pour but d'accentuer les contours, il s'agit simplement de l'inverse du filtre passe-bas, car il ne conserve que les hautes fréquences. Cette attaque est certainement la moins efficace des transformations car elle conserve le bruit, et que c'est souvent à ce niveau là que se situe la marque.

Filtre médian : Ce filtre, non linéaire, remplace la valeur d'un pixel par la médiane des valeurs de ces voisins. Il est plus robuste que le précèdent pour différents types de bruits artificiels, donc plus efficace en tant qu'attaque.

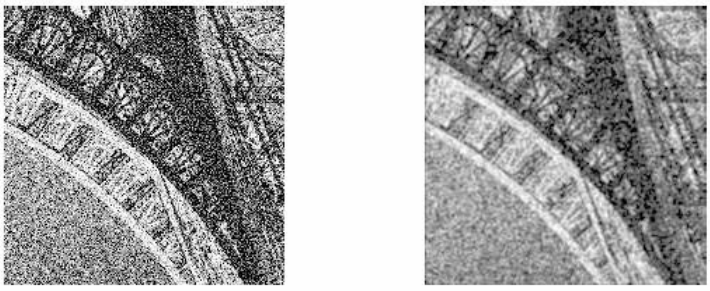


Figure 3.11 Exemple de filtrage linéaire

- **Les compressions** : Les compressions avec pertes sont souvent une succession des différentes transformations vues précédemment, ce qui en fait des attaques

involontaires souvent très efficaces. Un exemple peut être la compression JPEG. Ce type d'attaque s'applique aussi à tout ce qui est conversion de format, par exemple du JPEG vers du GIF.

Figure 3.12 Exemple de perte lors de compression JPEG

II.6.2 Les attaques volontaires

Plusieurs outils académiques disponibles sur l'Internet permettent de "lessiver" la marque d'une image en essayant d'altérer au minimum l'image originale. Ces programmes, appelés *crackers*, sont mis à la disposition des chercheurs pour mettre à l'épreuve leurs algorithmes de marquage robuste. Ils perturbent l'image de telle sorte que, même si la marque reste présente dans l'image, celle-ci est difficile, voire très difficile à extraire, sans recourir à l'image originale afin de se recaler. Ces perturbations sont souvent des transformations géométriques de manière aléatoire. Parmi les outils actuellement disponibles qui réalisent une telle perturbation, les plus référencés sont Unzign et surtout Stirmark [Web3]. UnZign modifie sensiblement la taille de l'image et Stirmark crée des déformations locales et simule un processus d'impression suivie d'une digitalisation de l'image à l'aide d'un scanner.

II.6.3 Les attaques de nature cryptologique

Ce type d'attaques est beaucoup plus intéressant, car il demande des connaissances en traitement du signal ainsi qu'une analyse sérieuse de l'algorithme de marquage. Les attaques malicieuses sont différentes des attaques aveugles car le pirate va s'attacher à trouver la faiblesse du système et selon cette faiblesse, il ciblera son attaque. Par bien des aspects, cela se rapproche beaucoup de la cryptanalyse

- **Exemple d'attaque sur le copyright**

Dans ce cas précis, le pirate va chercher à semer le trouble sur l'origine de l'image marquée et publiée sur l'Internet, par exemple. En effet, il ne sert à rien d'ajouter une autre marque à l'image divulguée, l'auteur ayant toujours à sa disposition la version originale. Le pirate essaie plutôt de recréer une image originale (c'est à dire sans marquage) en y soustrayant une fausse marque. Ainsi, il existe deux

personnes prétendant avoir la copie originale du contenu divulgué sur Internet. Il est impossible de confondre l'usurpateur.

- **Exemple d'attaque de collusion**

Les attaques de collusion mettent en jeu plusieurs pirates qui associent leurs efforts dans le but de rendre le processus d'attribution des droits par marquage inopérant. En disposant de plusieurs versions du même contenu, marquées différemment (typiquement dans une application d'estampillage), on vient forger un contenu, contenant un peu de toutes les versions disponibles, perceptuellement semblable au contenu original (donc à chacune des versions autorisées et marquées).
Dans ce cas, lorsqu'on présente le contenu forgé au décodeur du marquage, il peut arriver que :

- le contenu forgé soit déclaré ne contenir aucune marque ;
- le contenu forgé soit déclaré contenir plusieurs marques dont disposaient les pirates.

Dans les deux cas, le processus d'attribution des droits a été bluffé. En image, cela peut se faire très facilement en créant une mosaïque à partir des versions marquées disponibles pour l'attaque.

Conclusion

La dissimulation de données est un domaine jeune qui progresse rapidement. Si la robustesse aux attaques géométriques reste le talon d'Achille des techniques de marquage, la propriété la plus importante dans tout système stéganographique est l'indétectabilité par analyse statistique. Même si de grands progrès ont été faits ces dernières années, beaucoup reste à faire du côté de la sécurité des algorithmes de stéganographie et de marquage, pour pouvoir rivaliser avec la maturité des outils purement cryptographiques.

Les travaux de cette thèse sont dirigés vers une application particulière, à savoir la dissimulation de données en vue de l'*authentification* des données hôtes. Les techniques proposées sont conçues comme des alternatives ou des variantes des techniques générales de marquage existantes. Cependant, on s'intéressera plus particulièrement au marquage semi-fragile de préférence au marquage fragile, et au marquage sans perte qui supprime les contraintes de distorsion minimale. Les techniques proposées sont applicables à toute image numérique, toutefois, les applications pratiques que nous envisageons se situent dans le domaine médical. Dans le chapitre prochain, nous exposerons un état de l'art sur les techniques de marquage destinées à assurer spécialement l'authentification d'images numériques.

Chapitre 4

Etat de l'art sur l'authentification des images par marquage numérique

Introduction

Le réseau Internet constitue aujourd'hui, la plus grande bibliothèque du monde, en permettant à un nombre de plus en plus important de personnes, d'accéder à toutes sortes d'informations, par-delà frontières et cultures. Les informations puisées peuvent être de nature textuelle, sonore ou principalement des images. La croissance exponentielle du trafic des images est renforcée par l'apparition importante d'appareils photos numériques et l'utilisation de téléphones portables. Les images sont des données particulières du fait de la quantité importante d'information et de leur disposition bidimensionnelle. La transmission des images soulève donc un nombre conséquent de problèmes qui ne sont pas tous encore résolus à l'heure actuelle. Les utilisateurs attendent des solutions performantes permettant notamment d'assurer la protection des droits d'auteur, mais également de garantir l'authenticité des images échangées. Cette demande est d'autant plus forte que les techniques de manipulation d'images sont de plus en plus sophistiquées et accessibles au grand public, et que les exemples de falsifications de documents deviennent, malheureusement, de plus en plus fréquents. Dans ce contexte, le marquage d'image, bien qu'étant un domaine de recherche relativement récent, peut apporter des éléments de réponse complémentaires aux méthodes de cryptographie classiques.

Le problème de l'intégrité des images est encore peu abordé par la communauté « marquage numérique » et de nombreuses questions restent

ouvertes. Ceci est dû à ce qu'un service d'intégrité remet partiellement en cause certains paramétrages communément établis en marquage d'image destinés à assurer des fonctions de sécurité plus classiques telles que les droits d'auteur, notamment en termes de quantité et nature des informations cachées ainsi qu'en termes de robustesse. En effet, pour de telles fonctions, nous avons vu au niveau du chapitre précédent, que la marque était indépendante de l'image et est de taille limitée couramment à quelques dizaines de bits. Au contraire, les techniques de marquage propres à l'authentification sont gourmandes en capacité et nécessitent une robustesse limitée voire inexistante. De plus, et contrairement aux techniques classiquement employées en sécurité pour assurer cette fonction, la plupart des méthodes proposées, privilégient une intégrité en termes de contenu à une intégrité numérique stricte.
L'objectif de ce chapitre est de dresser un panorama des différentes méthodes permettant d'assurer un service d'intégrité adapté aux images par le biais du marquage numérique. Les systèmes d'authentification des images peuvent être regroupés de plusieurs manières suivant qu'ils assurent un service d'intégrité stricte ou bien une intégrité en termes de contenu, suivant le mode de stockage des données d'authentification ou bien encore selon la nature des informations qu'ils enfouissent dans l'image à protéger. Nous introduirons cette notion d'intégrité sémantique particulière aux images, ainsi que les critères à prendre en considération pour construire un système d'authentification performant. Plusieurs algorithmes significatifs seront détaillés afin de présenter les notions de base fréquemment usitées et d'introduire progressivement les notions clés associées à ce type de service.

I. Notions d'intégrité

La notion d'intégrité est un concept bien connu en sécurité. Sa définition repose sur une décision binaire qui garantit que les données reçues sont rigoureusement identiques à celles émises. Cette définition est en principe applicable à tout type de documents numériques, néanmoins, dans la pratique elle s'avère être beaucoup trop stricte et inadaptée pour les documents de type images. En effet, l'interprétation que l'on a d'une image dépend principalement des éléments la constituant plutôt que des valeurs numériques des pixels ou de sa résolution. En d'autres termes, le problème de l'intégrité des images se pose principalement en termes de contenu sémantique; c'est-à-dire la détection des modifications du document pouvant engendrer une gêne dans sa visualisation et/ou une erreur dans son interprétation (modification de la légende, disparition d'un visage, *etc*.). Dans le but d'assurer un service d'intégrité approprié aux images, il est donc primordial de distinguer les manipulations malveillantes consistant à détourner le contenu initial de l'image, des manipulations liées à son utilisation ou son stockage sous une forme numérique (conversion de format, compression, ré-échantillonnage, filtrage, *etc.*) réalisés par des fournisseurs de contenu ou les utilisateurs eux-mêmes. Malheureusement cette distinction n'est pas toujours aisée d'un point de vue informatique et dépend en partie du type d'image et de son utilisation. Par exemple, dans le cas particulier de l'imagerie médicale, des manipulations anodines, comme une simple compression, voire le processus de marquage lui-même, peuvent causer la disparition de certains signes visibles d'une pathologie

faussant alors le diagnostic du médecin. Dans ce contexte, l'utilisation de méthodes issues de la cryptographie classique sera plus appropriée pour garantir « une intégrité stricte » du document.
On pourrait être tenté d'utiliser pour cela des fonctions de hachage cryptographiques telles mentionnées dans le chapitre 2, mais celles-ci s'avèrent mal adaptées, car on souhaite ici préserver non pas l'intégrité numérique, mais l'intégrité perceptive.
En effet, selon le contexte applicatif, on peut souhaiter soit détecter tout type de modification, soit un ensemble donné de transformations (interdites). On peut par exemple autoriser certains taux de compression, des changement d'échelle, ... Dans le premier cas, on utilisera des techniques de marquage fragile où le marquage disparaît à la moindre manipulation comme mentionné au chapitre 3, et dans le deuxième des méthodes de marquage semi-fragile où le marquage résiste aux manipulations autorisées. On parle alors d' « intégrité souple ».

II. Exemples classiques de manipulations malveillantes

Les messages véhiculés par les images ont un impact considérable. En effet, le réalisme d'une photographie est tel que nous avons tendance à prendre pour réelles des scènes qui ne le sont pas. Toutes les images, y compris celles réalisées en toute innocence, ont la capacité d'être détournées de leur sens. Les manipulations, qui avant, nécessitaient des moyens coûteux sont désormais à la portée de tout le monde. Avec les progrès des techniques de traitement d'image et du tout numérique elles deviennent quasi indécelables. Dans ce contexte, un service d'intégrité d'image n'a bien évidemment pas la prétention de vérifier la véracité des événements, mais de déceler des manipulations qui auraient pu y être apportées *a posteriori* (*i.e.* entre la prise de la photographie et sa diffusion), dans le but de détourner le contenu de l'image ou de rendre impossible toute interprétation. Des exemples célèbres de manipulations intentionnelles d'images sont là pour attester de l'impact considérable que peuvent avoir ces dernières sur la société. La photographie truquée diffusée en 1995, dans l'émission de France 3, « La marche du siècle », où de jeunes « beurs » avaient été transformés à leur insu en redoutables intégristes, est un bel exemple de falsification d'image, ainsi qu'un véritable scandale journalistique qui a fait couler beaucoup d'encre à l'époque [Web5]. Les éléments ajoutés à la photographie créés de toutes pièces au moyen d'un logiciel de retouche à des fins de manipulation, exposèrent au grand jour le problème de la numérisation des images. Un autre exemple [Web6] qui a fait le tour du monde est celui de la photographie publiée en une du quotidien autrichien « *Neue Kronen Zeitung* » en 1998, qui prétend illustrer l'agressivité des manifestants opposés à l'entrée du parti de Haider dans le gouvernement autrichien. Par un truquage numérique, on a recadré la photographie et raccourci la distance entre un manifestant et un policier apparemment directement frappé. En réalité, comme l'atteste l'image originale diffusée par la suite par l'agence Reuters, une distance de près de deux mètres séparait les deux protagonistes. Ainsi, l'utilisation de l'image ou la vidéo comme élément à charge, sans être couplée par des techniques d'authentification, devient plus que douteuse et critiquable à l'heure où les caméras de surveillance envahissent les villes, les stades et les routes.

Plusieurs autres champs d'application de l'authentification d'images peuvent être identifiées. Nous citons, à titre non exhaustif, les domaines potentiels suivants:

- Archivage des images médicales: Les données d'authentification des patients peuvent être insérées au moment de la prise des images par l'hôpital afin de protéger leurs droits. Ainsi, en cas d'erreur médicale, ces images peuvent être utilisées par la justice.
- Enregistrement d'interrogatoires pour des enquêtes criminelles dans lesquelles la modification malveillante de certaines scènes (par les enquêteurs, par exemple) pourrait aboutir à des décisions juridiques graves si elle n'est pas détectée.
- Capture de scènes d'accidents à des fins d'assurance et des fins médico-légales : L'application d'une technique d'authentification d'images et de vidéos pourrait être utile dans la protection des droits des différentes parties, incluant la société d'assurance impliquée dans des accidents ou des catastrophes naturelles.
- Pendant les crises internationales, la contrefaçon d'images pourrait être utilisée à des fins de propagande ou pour manipuler l'opinion publique. Par conséquent, la télédiffusion est un autre secteur où l'authentification d'images est fortement applicable.
- Domaine militaire: L'authentification d'images permet aux services secrets militaires d'authentifier si les médias qu'ils ont reçus proviennent vraiment de leurs correspondants/alliés ou d'espions/ennemis et de vérifier si le contenu est vraiment original ou a été falsifié. Dans le cas où le contenu a été manipulé, un schéma d'authentification efficace pour ce type d'application, est celui qui permet, en plus, la localisation des modifications.

III. Caractéristiques d'un système d'authentification d'image

Dans cette section, on se propose de définir un schéma générique d'un système d'authentification d'image dont différentes formulations ont été initialement proposées par Wu et Liu [WL98] et Lin et Chang [LC00]. Le schéma général d'un tel système est donné en Figure 4.1 : Généralement, mais pas toujours, une même clef secrète disponible lors de la phase du marquage et de la phase d'authentification est utilisée pour générer une marque destinée à être insérée dans l'image hôte. L'image ainsi marquée est transférée à travers un canal de communication (Internet, satellite, etc…) ou sauvegardée dans une base de données. Pour authentifier une image marquée reçue ou extraite de la base, la même clef est utilisée pour extraire la marque aussi bien que pour générer la marque originale, puis les deux marques sont comparées. La différence des deux marques, comparée à un seuil de tolérance, constitue la sortie du système, et atteste donc de l'authenticité ou de la non authenticité de l'image.

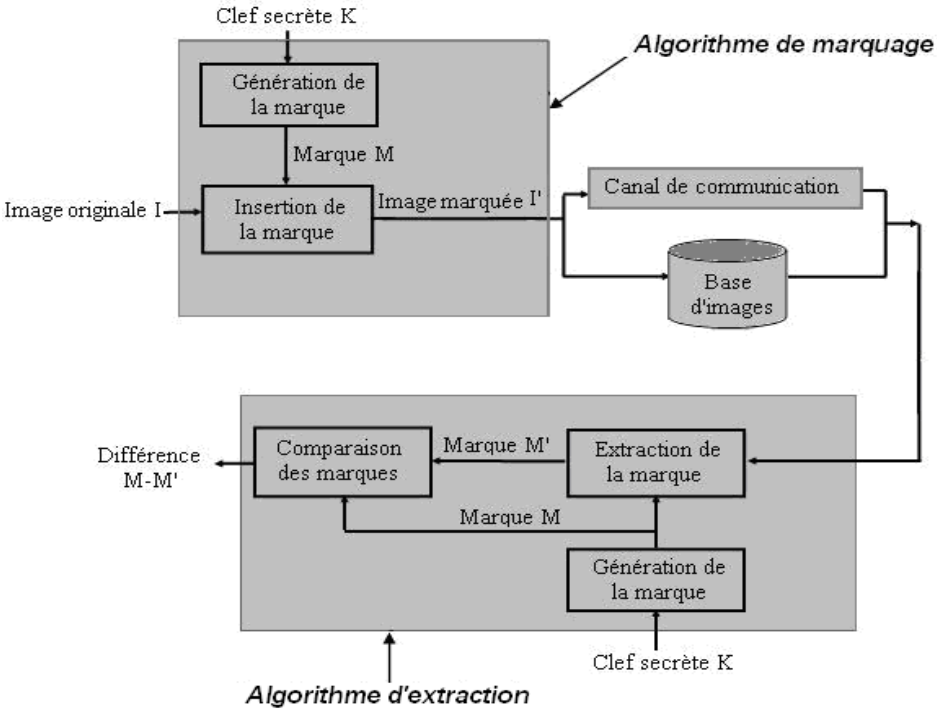


Figure 4.1 Schéma générique d'un système d'authentification

Pour être efficace, un système d'authentification d'images doit satisfaire les critères suivants [WL98] :

– **sensibilité** : le système doit être capable de déceler des manipulations pouvant modifier l'interprétation que l'on a d'une image, telles que des recadrages (cropping) ou des retouches locales;

– **tolérance** : le système doit être tolérant vis-à-vis des algorithmes de compression avec perte tels que Jpeg, et plus généralement vis-à-vis des manipulations bienveillantes (générées, par exemple, par les fournisseurs de contenu multimédia);

– **localisation des régions altérées** : le système doit être en mesure de donner à l'utilisateur une information visuelle permettant d'identifier les régions de l'image qui ont été manipulées.

– **reconstruction des régions altérées** : le système doit éventuellement permettre une restauration partielle des zones de l'image qui ont été manipulées ou détruites, afin de donner à l'utilisateur la possibilité de se faire une idée sur le contenu original de ces régions.

En plus des critères précédents, d'autres contraintes techniques sont également à prendre en considération :

– **mode de stockage** : il est préférable de cacher les données d'authentification dans l'image elle-même, sous la forme d'une marque, plutôt que dans un fichier séparé comme dans le cas d'une signature externe;

– **mode d'extraction** : suivant que les données d'authentification sont dépendantes ou non de l'image, on optera pour un mode d'extraction de la marque aveugle ou semi-aveugle. En mode d'extraction aveugle, la marque représentant les données d'authentification est récupérée à partir de l'image marquée seule (éventuellement manipulée), alors qu'en semi-aveugle il s'agit principalement de vérifier la présence de telle marque dans une image (généralement via un score de corrélation). Il est bien évident qu'un mode d'extraction non aveugle est dénué de sens pour un service d'intégrité dans la mesure où il fait appel à l'image originale;

– **algorithme asymétrique** : Contrairement aux services de sécurité plus classiques comme la preuve de propriété (copyrighting) où l'on peut se contenter d'une même clef secrète pour l'insertion et l'extraction de la marque, un service d'intégrité nécessite de préférence l'utilisation d'un algorithme de marquage asymétrique dans la mesure où tout un chacun doit pouvoir s'assurer de l'authenticité d'une image;

– **visibilité** : les données d'authentification doivent être invisibles (dans les conditions normales de visualisation). Il s'agit de faire en sorte que l'impact visuel du marquage (*i.e.* la distorsion) soit le plus faible possible afin que le document marqué reste fidèle à l'original;

– **robustesse et sécurité** : les données d'authentification ont tout à gagner à être protégées par des méthodes de chiffrement de manière à éviter qu'elles soient falsifiées ou manipulées;

– **protocoles** : enfin, les protocoles tiennent également une place prépondérante dans tout système d'authentification d'images. En effet, l'algorithme ne permet pas à lui seul de garantir l'authenticité d'une image. Il est nécessaire de définir en plus un ensemble de spécifications décrivant les conventions et les règles du système, comme par exemple la gestion des clefs ou bien encore éviter qu'une image déjà protégée, puisse l'être à nouveau, a fortiori si elle a été manipulée.

IV. Revue des méthodes existantes

IV.1 Marquage fragile

Les premières méthodes proposées pour assurer un service d'intégrité étaient basées sur l'utilisation d'un marquage fragile, par opposition au marquage robuste classiquement utilisé pour la protection des droits d'auteur. Le principe de ces approches est d'insérer une marque ou un logo binaire (généralement prédéfini et indépendant des données à protéger) dans l'image d'origine, de telle manière que les moindres modifications apportées à l'image se répercutent également sur la marque insérée. Pour vérifier l'intégrité d'une image, il suffit alors de vérifier

localement la présence de cette marque. Les techniques proposées en ce sens sont les suivantes, réparties en différents domaines de travail.

IV.1.1. Insertion dans le domaine spatial

a- Insertion de somme de contrôle ou « checksum » dans les LSB

Une des premières techniques utilisées pour vérifier l'intégrité d'une image visait à insérer des valeurs de « checksums » dans les bits les moins significatifs (LSB) des pixels de l'image. On sait en effet que ces bits sont très sensibles aux manipulations.
L'algorithme proposé par Walton en 1995 [Wal95] consiste à sélectionner, de manière pseudo-aléatoire (en fonction d'une clé), des groupes de pixels et de calculer, pour chacun d'eux, une valeur de « checksum ». Ces valeurs sont obtenues à partir des nombres formés par les 7 bits les plus significatifs (MSB) des pixels sélectionnés, et sont ensuite insérées sous forme binaire au niveau des bits de poids faible. Ci-après, de manière plus détaillée l'algorithme tel qu'il était proposé à l'origine :

Algorithme d'insertion
1. Soit N suffisamment grand ;
2. Diviser l'image en blocs de taille 8 × 8 pixels ;
3. Pour chaque bloc Bi :
– définir un ordre de parcours pseudo-aléatoire (selon par exemple une clé secrète et l'indice du bloc Bi) des 64 pixels ($p_1, p_2, \ldots, p_{64}$) ;
- – générer une séquence pseudo-aléatoire de 64 entiers ($a_1, a_2, \ldots, a_{64}$) du même ordre de grandeur que N ;
- – la valeur de checksum S est alors calculée de la manière suivante :

$$S = \sum_{j=1}^{64} (a_j . g(p_j) \bmod N$$

avec $g(pj)$ le niveau de gris du pixel p_j en ne tenant compte que des 7 MSB.
- – coder et chiffrer S en binaire ;
- – insérer la séquence binaire résultante au niveau des LSB des pixels du bloc.

Algorithme de vérification
Il est dual de celui d'insertion. Il consiste à vérifier pour chaque bloc, la valeur de « checksum » recalculée à partir des MSB des pixels de l'image testée, avec celle de l'image originale codée au niveau des LSB.
Cette méthode garantit d'une part, en insérant les données d'authentification directement au niveau des LSB de l'image, une distorsion visuelle minime, quasi imperceptible par l'œil humain. D'autre part, elle a l'avantage d'être simple, rapide et sensible à la moindre modification de l'image (*i.e.* réponse binaire équivalente à une intégrité stricte). Si on échange, par exemple, les MSB de deux pixels quelconques d'un même bloc, la valeur de S s'en trouvera automatiquement modifiée car chaque pixel p_j est multiplié par un coefficient a_j différent. De plus, l'ordre de parcours des pixels p_j ainsi que les valeurs des coefficients a_j sont dépendants du bloc, ce qui rend impossible un éventuel « copier/coller » entre deux blocs différents d'une même image.

On remarquera cependant, qu'ainsi défini, il est possible avec cette méthode d'intervertir deux blocs homologues (*i.e.* de même position) de deux images protégées avec la même clé, sans que le système ne décèle une perte d'intégrité (depuis différentes améliorations ont été proposées pour pallier à ce type d'attaque). Par contre si l'image est légèrement recadrée ou compressée, le système détecte une perte d'intégrité alors que le contenu sémantique de l'image reste inchangé.

b- Schéma de Yeung et Mintzer [YM97]

Cette méthode encode un logo binaire dans les bits de poids faible, et la décision quant à l'authenticité de l'image s'effectue par rapport au logo qu'on sait y avoir caché.

Les données sont insérées en utilisant une LUT (Look Up Table) mettant l'espace des valeurs possibles des pixels en relation avec l'ensemble binaire {0,1}. Il insère ainsi un logo binaire, de la taille de l'image à marquer. Pour chaque pixel de l'image, la LUT permet d'obtenir la valeur binaire correspondante. Si celle-ci est identique à la valeur du bit à insérer, le pixel de l'image n'est pas modifié. Si la valeur est différente, ce pixel est ajusté jusqu'à obtention de la bonne valeur.

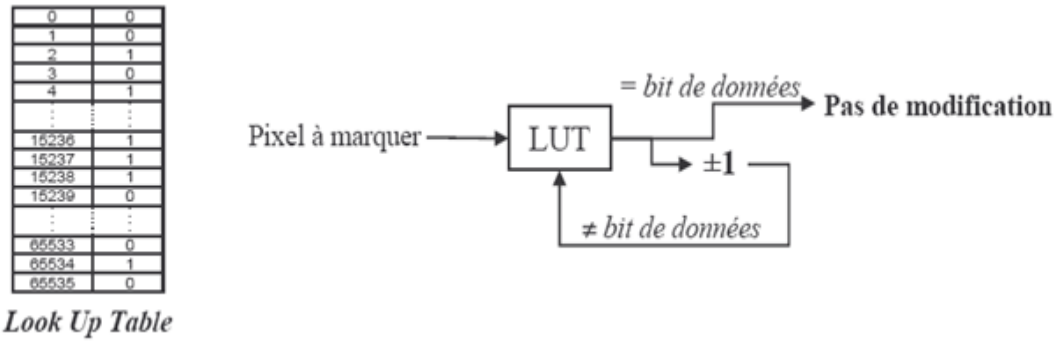


Figure 4.2 Mécanisme d'insertion par la méthode Yeung et Mintzer [YM97]

Cette méthode souffre cependant du fait que les relations testées au niveau de chaque pixel ont 50% de chances d'être vérifiées naturellement. La sécurité de cette technique a été évaluée dans [FG99] et une amélioration a été proposée dans [FGM00]. Cependant, notons que si l'attaquant dispose du dispositif d'authentification qui retourne une réponse binaire pour chaque pixel, ce procédé reste vulnérable.

Dans le but d'améliorer le comportement de cette technique, une autre approche a été étudiée : l'authentification hiérarchique. On utilise pour cela une représentation hiérarchique de l'image, ce qui rend les blocs de pixels dépendants les uns des autres. Une adaptation de la méthode précédente a été proposée pour cette approche [FGM00].

c- Schéma de Wong

Une autre proposition, beaucoup plus élaborée et reposant sur la combinaison d'outils de traitement du signal et cryptographiques (hachage et chiffrement) a été publiée dans [Won98]. Elle insère une empreinte de l'image dans l'image elle-même. L'empreinte recalculée à la détection sera comparée à celle qui a été insérée, mettant ainsi en valeur les éventuelles modifications. Les fonctions de

hachage MD5 ou SHA256 génèrent une empreinte sous forme de signature de l'image.
Cette méthode insère au niveau des LSB, un logo binaire, permettant d'identifier le propriétaire de l'image, et une empreinte de l'image. L'image et le logo sont découpés en blocs. L'empreinte de chaque bloc de l'image (de laquelle ont été supprimés préalablement les bits de poids faible) est calculée. Le flux de l'empreinte est alors ajouté, grâce à un OU exclusif, au bloc du logo qui sera alors inséré au niveau des LSB du bloc.

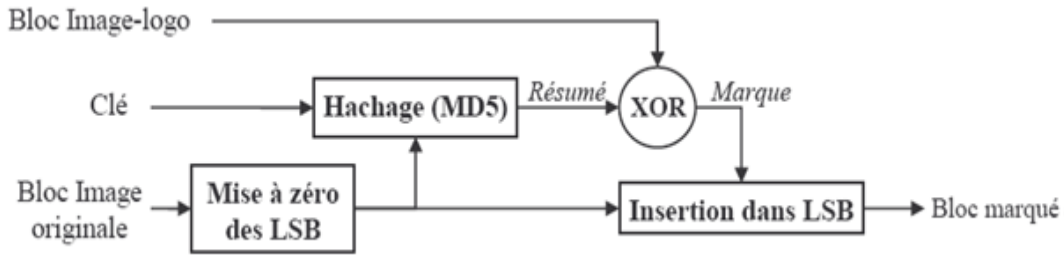


Figure 4.3 Mécanisme d'insertion par la méthode de Wong

L'extraction se fera en recalculant le résumé des blocs de l'image marquée sans les LSB. Le résumé obtenu sera additionné aux LSB grâce à un OU exclusif, ce qui permet, si l'image n'a pas été dégradée, de retrouver le logo. En cas de dégradation de l'image marquée, des modifications évidentes du logo apparaissent.

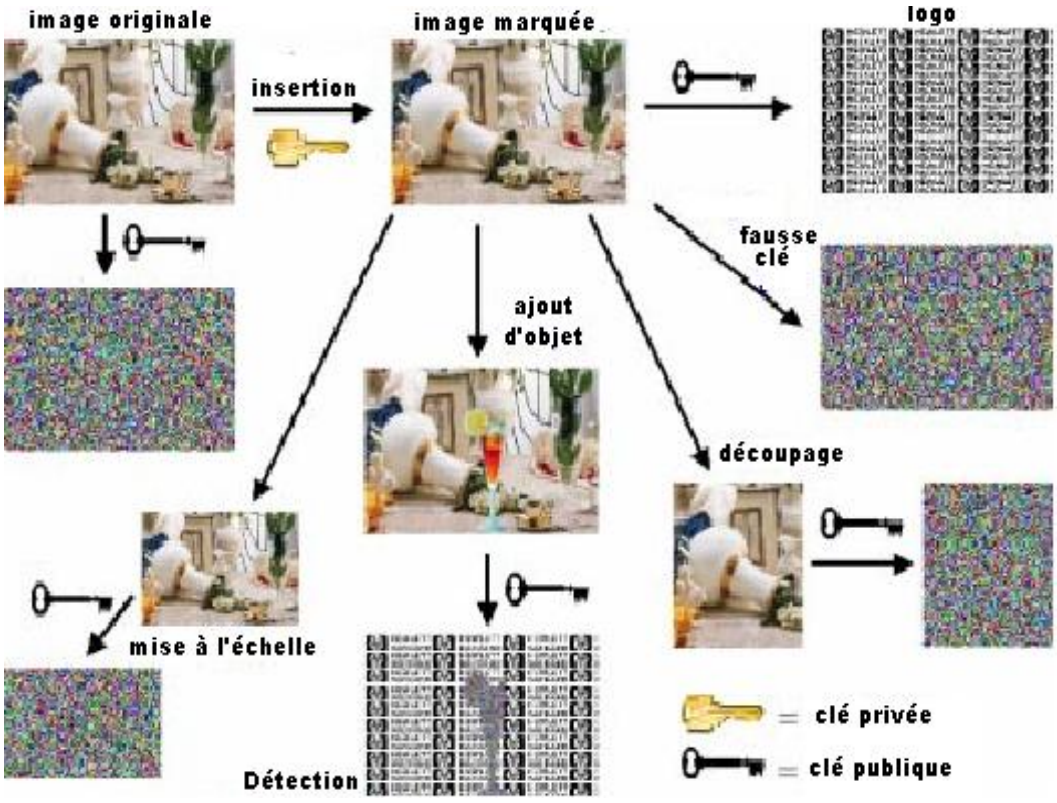


Figure 4.4 Fonctionnement de la méthode de Wong [Won98]

Cette méthode est très sensible à l'attaque de Holliman et Memon [HM00] où l'attaquant, disposant d'un ensemble d'images authentiques protégées par le même logo et la même clé, cherche à estimer le contenu d'une image originale à partir du collage d'autres blocs authentiques pris dans un dictionnaire. L'extracteur retrouve alors une empreinte du bloc exact. D'autres travaux se sont particulièrement focalisés sur la résistance à cette attaque [KVH00] [LLC00].
Cette faille dans l'algorithme de Wong est d'autant plus importante que, dans le domaine de l'imagerie médicale qui nous intéresse particulièrement, les bases de données d'images de même modalité et d'une même partie anatomique sont très courantes. Il est alors aisé d'ajouter ou de supprimer un signe pathologique sur une image.
Enfin, notons que Queluz [Que98] propose un schéma d'authentification pour des portions de l'image, qui permet de localiser les régions altérées et de détecter si le contenu provient du montage de morceaux d'images. Cette méthode offre une très bonne résistance face aux attaques possibles.

d- Self-embedding

Fridrich et Goljan [FG99] ont, quant à eux, développé une technique utilisant également les LSB comme support, mais dans le but, cette fois-ci, de cacher suffisamment d'informations afin de pouvoir non seulement déceler d'éventuelles manipulations, mais aussi de permettre une reconstruction partielle des régions détériorées.
L'idée de base consiste à découper l'image en blocs 8×8, à en calculer les coefficients DCT (Transformée en Cosinus Discrète) en ne tenant compte bien évidemment que des MSB. Ces coefficients DCT sont ensuite quantifiés à l'aide de la table de quantification correspondant à une compression Jpeg d'une qualité de l'ordre de 50 %. La matrice quantifiée résultante est alors encodée sur 64 bits et insérée au niveau des LSB des pixels d'un autre bloc. Le bloc servant de support au marquage doit être suffisamment éloigné afin d'éviter qu'une manipulation locale de l'image ne détériore à la fois l'image et les données de reconstruction correspondantes.
Comme pour toutes les méthodes de marquage utilisant les LSB comme support, l'impact visuel est très faible. Par contre, la qualité des régions restaurées est nettement inférieure à celle d'une compression Jpeg 50 %, mais largement suffisante pour informer l'utilisateur sur le contenu original de ces régions. Les auteurs ont également proposé une variante afin d'améliorer légèrement la qualité de la reconstruction en utilisant cette fois-ci les deux bits de poids faible comme support (la matrice quantifiée étant alors codée sur 128 bits). La reconstruction est certes meilleure, mais l'image marquée perd sensiblement en qualité.
Le principal inconvénient de cette méthode est lié à la nature très fragile du marquage qui ne garantit pas, dès lors que plusieurs régions de l'image ont été manipulées, une restauration correcte. En effet, les données de reconstruction correspondant à un bloc erroné peuvent elles aussi être altérées si les LSB les supportant ont eux aussi été modifiés. Ce problème est d'autant plus vrai lorsque l'image subie des manipulations globales, même « faibles », comme un filtrage passe-bas ou une compression Jpeg.

IV.1.2. Insertion dans le domaine transformé

a- Zhu [Zhu99] insère la marque dans le domaine DCT (Transformée en Cosinus Discrète). La marque est constituée à partir d'un bruit blanc généré grâce à une clé. Cette clé est nécessaire à l'extracteur qui régénère le bruit blanc et le compare à la marque extraite. L'erreur ainsi calculée est comparée à des seuils pour déterminer l'intégrité de l'image. Si cette méthode ne permet pas la localisation des dégradations, elle peut par contre les quantifier, et repérer les fréquences les plus atteintes.

b- Kundur [KH99] utilise quant à lui la DWT (Discret Wavelet Transform) et insère les données en quantifiant certains coefficients des images détails de tous les niveaux. Le choix de ces coefficients est déterminé de façon à ce que les données insérées soient étalées spatialement et sur toutes les résolutions. La quantification de ces coefficients suivant les données à insérer s'effectue en découpant l'espace des réels suivant un pas de quantification. A chaque pas est associé, alternativement la valeur 0 ou 1. Ainsi, si au coefficient d'ondelettes correspond la valeur binaire à insérer, le coefficient n'est pas modifié. Par contre, si les valeurs ne correspondent pas, le coefficient est modifié.

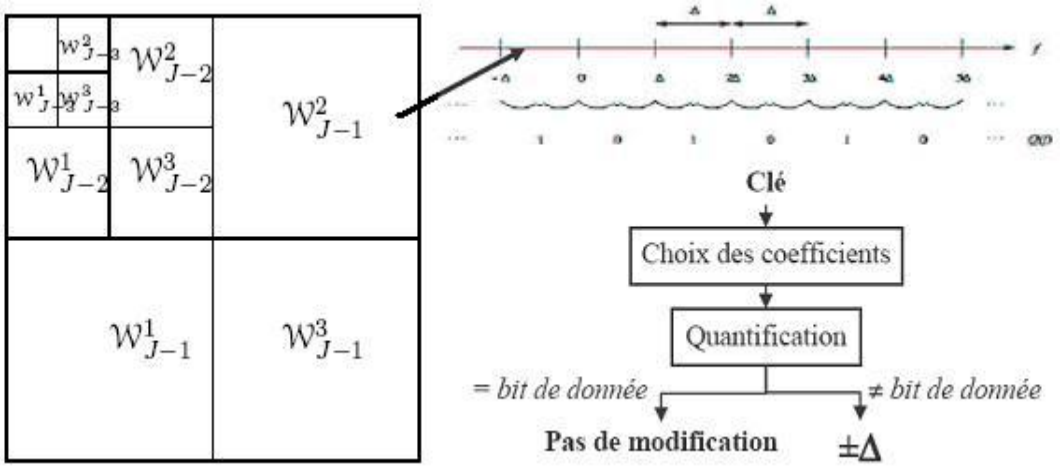


Figure 4.5 Mécanisme d'insertion dans la méthode de Kundur

IV.2 Marquage semi-fragile

D'une manière générale, on peut légitimement se poser la question de l'intérêt des méthodes de marquages fragiles vis à vis des techniques cryptographiques classiques, dans la mesure où elles ne garantissent également qu'une intégrité stricte. Face à ce constat de semi-échec, les recherches s'orientent actuellement vers des approches dites semi-fragiles. Les méthodes ayant recours à un marquage semi-fragile se distinguent des méthodes fragiles dans la mesure où elles offrent une robustesse accrue face à certaines manipulations d'image. L'objectif recherché est de pouvoir discriminer des opérations malveillantes, comme par exemple l'ajout ou la suppression d'un élément important de l'image, de transformations globales «raisonnables» ne portant pas atteinte au contenu sémantique de l'image.

L'utilisation de telles méthodes a été initialement motivée par le fait que les images sont généralement transmises et stockées sous une forme compressée et que pour la majorité des applications, les pertes liées au processus de compression n'affectent pas l'intégrité de l'image au sens de son interprétation. Plus tard, le champ des transformations autorisées s'est élargi pour répondre à des besoins spécifiques dans des applications particulières.

IV.2.1 Exemple de méthode transparente à la compression Jpeg

Lin et Chang [LC00] proposent un algorithme d'authentification robuste à la compression Jpeg. Les composantes significatives d'une image de façon perceptuelle correspondent en général aux basses et moyennes fréquences. Si l'on modifie les basses fréquences, l'impact visuel est important. Les hautes fréquences (représentatives des détails) sont enlevées par la compression JPEG. Il s'avère donc judicieux de marquer un bit dans les moyennes fréquences du bloc donné. Dans leur article, les auteurs ont mis en évidence et démontré deux propriétés d'invariance des coefficients DCT vis-à-vis de la compression Jpeg.
La première propriété énonce que si on donne à un coefficient DCT, quel qu'il soit, une valeur entière multiple d'un pas de quantification prédéfini *Qm* supérieur à tous les pas de quantification possibles d'une compression Jpeg acceptable (*i.e*. facteur qualité de 50 % environ), alors cette valeur peut être recalculée exactement après une compression Jpeg acceptable. La deuxième propriété définit une règle d'invariance de la relation d'ordre entre les coefficients homologues de deux blocs DCT vis-à-vis de la compression Jpeg. En effet, lors de la compression, les différents blocs DCT d'une image sont tous divisés par la même table de quantification, de ce fait, la relation qui lie les coefficients de mêmes coordonnées de deux blocs reste inchangée après le processus de quantification. La seule exception est que dans certains cas, des inégalités strictes peuvent devenir de simples égalités, par le biais de la quantification. Le système d'authentification proposé par Lin et Chang repose donc sur ces deux propriétés. La première est utilisée pour définir un support de marquage robuste à la compression Jpeg, tandis que la seconde sert à générer les données d'authentification proprement dites. Les étapes d'insertion et d'authentification peuvent se résumer ainsi :

Algorithme 1 : Génération des bits d'authentification
1. Découper l'image originale en blocs 8 × 8
2. Apparier les blocs deux par deux en fonction d'une clé secrète
3. Pour chaque paire de blocs (p, q) :
– sélectionner un ensemble B de n coefficients DCT (autres que la composante continue) ;
– générer la signature binaire φ de la paire de blocs à l'aide de la règle suivante :

$$\phi(\nu) = \begin{cases} 1, & F_p(\nu - F_q(\nu) \geqslant 0 \\ 0, & F_p(\nu) - F_q(\nu) < 0 \end{cases}$$

avec $v \in B$ et $F(v)$ la valeur du coefficient v.
– insérer les bits d'authentification suivant l'algorithme 2.

La signature binaire obtenue est ensuite cachée en partie dans chacun des deux blocs de la paire. L'algorithme de marquage utilisé est relativement simple puisqu'il s'agit de définir une relation d'égalité entre les LSB des coefficients DCT prédéfinis avec les bits de la signature.

Algorithme 2 : Insertion du marquage

1. Sélectionner un ensemble *E*, de *n*/2 coefficients DCT, avec $E \cap B = \emptyset$;
2. Pour cacher un bit d'authentification $\varphi(v)$ dans un coefficient DCT ω :

$$\text{Soit} \qquad f'_p(w) = \left\lfloor \frac{F_p(w)}{Q'_m(w)} \right\rfloor$$

$$\tilde{\tilde{F}}_p(w) = \begin{cases} f'_p(w).Q'_m(w), & si\ LSB(f'_p(w)) = \phi(v) \\ \left(f'_p(w) + signe\left(\frac{F_p(w)}{Q'_m(w)} - f'(w) \right) \right).Q'_m(w), & \text{sinon} \end{cases}$$

avec *signe(x) =0 si x< 0 et 1 sinon*

La vérification de l'intégrité d'une image est réalisée simplement en extrayant les bits d'authentification des coefficients DCT recevant le marquage et en comparant la signature extraite avec celle obtenue à partir des blocs de l'image testée. Si les deux signatures correspondent parfaitement, la paire de blocs est alors jugée intègre, dans le cas contraire cela signifiera que l'un des deux blocs, voire les deux, ont été manipulés.

Les auteurs ont eux-mêmes proposé de nombreuses améliorations à cette méthode [LC01], notamment l'ajout de bits de reconstruction. L'intérêt de ces bits supplémentaires est double. Ils permettent d'une part, comme leur nom l'indique, de reconstruire partiellement les blocs erronés, et d'autre part d'aider à localiser précisément les zones de l'image qui ont réellement été altérées (*i.e.* lever l'ambiguïté sur l'identification des blocs erronés). Les bits de reconstruction sont obtenus à partir d'une version sous-échantillonnée et compressée de l'image originale, et sont ensuite insérés de la même manière que les bits d'authentification dans quatre blocs de l'image originale.

IV.2.2 Marquage par région

Le marquage par région consiste à découper l'image que l'on souhaite protéger en blocs relativement grands (de l'ordre de 64 × 64 pixels) et d'insérer, dans chacun d'eux, une marque « relativement robuste ». Lorsque l'on souhaite vérifier l'intégrité de l'image, on teste la présence de la marque dans les différents blocs. Dans le cas où la marque est présente avec une probabilité élevée dans chacun des blocs, on peut affirmer que l'image testée est intègre.

La technique *Variable-Watermark Two-Dimensional* (VW2D) décrite par Wolfgang et Delp [WD99] reprend le principe décrit précédemment, à savoir de

cacher une marque binaire différente $W(b)$ dans chaque bloc b d'une image X. Ils préconisent de générer une marque binaire pseudo-aléatoire à partir de « *m*sequences » à la manière des travaux initiés par Van Shyndel *et al* [VTO98]. L'utilisation de « *m*-sequences » est en effet motivée par le fait qu'elles ont d'excellentes propriétés d'auto-corrélation, ainsi qu'une très bonne robustesse à l'ajout de bruit. Dans le système d'authentification proposé, la séquence binaire {0, 1} est transformée en une séquence de {−1, 1}, puis arrangée de manière à former un bloc de même taille que le bloc de l'image auquel elle va être modulée. La modulation de la marque et de l'image est réalisée très simplement en ajoutant ou en supprimant un niveau de gris au pixel correspondant (équation 1) :

$$Y(b) = X(b) + W(b) \qquad (1)$$

avec X l'image originale, et Y l'image marquée.
Ensuite, pour déterminer si la marque recherchée est bien présente dans un bloc, on calcule un score statistique δ (équation 3) basé sur un calcul de corrélation (équation 2) entre l'image (marquée et attaquée) et la marque :

$$A(b).B(b) = \sum_i \sum_j A(i, j)B(i, j) \qquad (2)$$

$$\delta(b) = Y(b).W(b) - Z(b).W(b) \qquad (3)$$

avec Z, l'image à tester, la marque W est supposée connue.
Si $\delta < T$, avec T un seuil fixé par l'utilisateur, le bloc est alors jugé authentique.
En jouant sur la valeur de T, on tolère des changements plus ou moins importants dans l'image. De ce fait il est possible d'affiner la détection en définissant plusieurs seuils correspondant à plusieurs niveaux de dégradation pour les blocs (par exemple : intègre, légèrement altéré, très dégradé, complètement modifié).
Cependant, dans la pratique, cette méthode n'offre qu'un intérêt limité dans la mesure où il est nécessaire de stocker au minimum, pour chaque bloc b d'une image, le résultat de la corrélation entre le bloc marqué $Y(b)$ et la marque cachée $W(b)$.
Fridrich [FG99], [FG00] propose une technique similaire, mais préconise, pour des raisons de sécurité, de rendre le marquage dépendant de la région de l'image dans laquelle il est inséré. La marque binaire utilisée correspond à un signal pseudo-aléatoire généré à partir d'une clé secrète, du numéro du bloc et d'un M-tuplet de bits représentatifs de la portion d'image considérée. Chaque bloc est ensuite marqué en utilisant une technique d'étalement de spectre, similaire à celle proposée par Ó Ruanaidh [OP97]. D'après l'auteur, la marque offre une bonne robustesse aux opérations classiques de traitement d'image telles que de petits ajustements de contraste ou de luminosité, l'ajout de bruit, l'application de filtres passe-bas ou passe-haut, l'égalisation d'histogramme, ou bien encore une compression Jpeg de l'ordre de 50 %, permettant ainsi de distinguer des changements liés à l'utilisation d'une image, des manipulations malveillantes.

IV.2.3 Les ondelettes

Les techniques basées sur les ondelettes sont actuellement de plus en plus fréquentes ou en cours d'investigation. Parmi celles-ci on peut citer celle de Kundur et Hatzinakos [KH99] et celle de Lin et Chang [LC00]. Le principe de la méthode proposée par Lin et Chang est de choisir, tout d'abord, un bruit pseudo-aléatoire et une ondelette de base, qui constituent le secret du système d'authentification. Puis, de décomposer l'image en 4 sous-bandes (LL, LH, HL et HH) en fonction de l'ondelette de base choisie au départ. L'étape suivante revient à substituer la sous-bande HH par le bruit pseudo-aléatoire et à effectuer ensuite la transformation en ondelettes inverse afin d'obtenir l'image marquée. Il est intéressant de noter que le fait de modifier uniquement la sous-bande HH (*i.e*. hautes fréquences) n'entraîne pas de dégradations visibles.
Le processus d'authentification consiste alors à effectuer la même décomposition que lors de la phase d'insertion, puis à corréler la sous-bande HH obtenue avec le bruit pseudo-aléatoire. Si l'image n'a subi aucune manipulation, le résultat du test ressemblera à une matrice de points uniformément répartis. Dans le cas contraire, la distribution perdra son caractère uniforme dans les régions où l'image a été manipulée. Les auteurs font remarquer que cette méthode est « perméable » à certaines manipulations telles qu'un flou ou un rehaussement des contours, dans la mesure où les changements ne sont pas trop importants.
Expérimentalement, cette méthode permet également de laisser passer une légère compression Jpeg. Par contre, les auteurs ne démontrent pas la robustesse de leur méthode face à des attaques spécifiques visant par exemple à substituer la sous-bande HH ou au contraire à la préserver (*i.e*. modifier l'image, puis réinsérer la sous-bande HH de l'image originale protégée). En d'autres termes, est-ce que le choix de l'ondelette de base comme secret est suffisant pour éviter ce type d'attaques ?
Une autre solution reposant sur la transformée en ondelettes est présentée dans [VVB03], et on trouve dans [YC05] la description d'un système récent dédié aux images codées au format JPEG2000.

IV.2.4 Marquage de caractéristiques de l'image ou basé sur le contenu

L'authentification basée sur le contenu est utilisée pour authentifier un ensemble de caractéristiques perceptuelles de l'image, appelé "contenu", plutôt que l'image elle-même. Le contenu détermine comment l'être humain interprète la signification sémantique de l'image.
L'idée de base dans cette méthode, consiste à extraire certaines caractéristiques de l'image originale (couleur, formes, contours, texture) et à les cacher ensuite dans l'image sous la forme d'un marquage robuste et invisible, au sens classique du « copyright ». En général, le contenu est représenté par un vecteur appelé « vecteur de caractéristiques » ou *« feature vector »*, et l'authentification est jaugée à travers la distance *d* entre le vecteur de caractéristiques de l'image originale *I* et celui de l'image à authentifier *I'* :

$$d = \left\| vecteur(I) - vecteur(I') \right\|$$

Si la distance *d* est supérieure à un seuil *S* prédéfini, dépendant de l'application (d< S), le contenu est jugé authentique, sinon, l'image est considérée comme étant manipulée.
Différents schémas d'authentification basée sur le contenu peuvent être envisagés, suivant la mise en oeuvre de l'extraction de caractéristiques et la comparaison correspondante montrée dans la Figure 4.6.

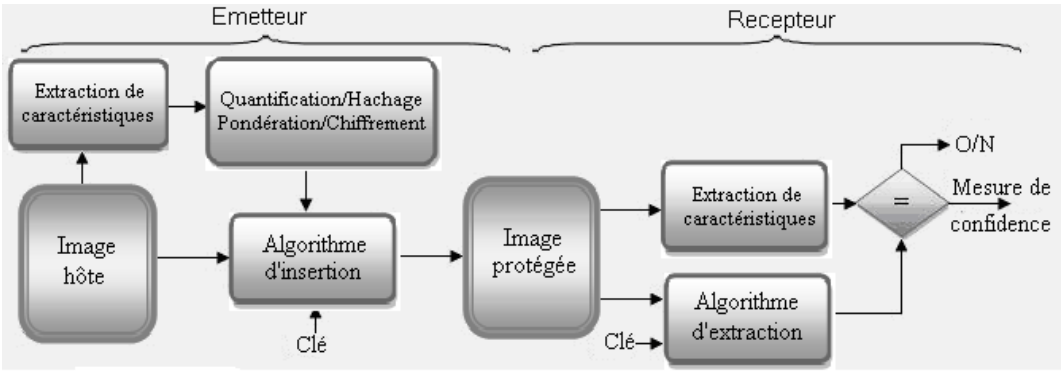


Figure 4.6 Marquage de caractéristiques

L'efficacité d'un système d'authentification basée sur le contenu se mesure en termes de taux de fausses alarmes et de taux de faux rejets.
Les caractéristiques utilisées dans un tel schéma doivent satisfaire les exigences suivantes :
1. Identifier l'image de manière univoque ;
2. Etre invariantes à des manipulations modérées qui préservent la qualité perceptuelle de l'image, telles la compression, le débruitage...
3. Etre hautement sensibles aux modifications qui affectent le contenu.
4. La localisation des modifications est une fonctionnalité supplémentaire très souhaitable.
5. La taille des données représentant ces caractéristiques doit être suffisamment faible pour faciliter la comparaison.
6. La complexité de calcul doit être raisonnable pour permettre des applications effectives.

Le choix des caractéristiques de l'image est primordial dans la mesure où il va conditionner les manipulations que l'on pourra détecter et celles qu'on laissera passer. De plus, ce choix dépend également du type d'image considéré (peinture, image satellite, image médicale, photo, *etc.*), ainsi que de l'application visée. D'une manière générale, on sélectionne les caractéristiques de l'image en fonction de leur stabilité face aux différentes attaques.
Typiquement, on recherchera des caractéristiques qui sont invariantes face à une compression Jpeg, à de faibles transformations géométriques, à un filtrage/débruitage, mais sensibles à des retouches locales de l'image, comme par exemple la luminance moyenne par bloc. Cette technique impose également de nouvelles contraintes, principalement en termes de robustesse et de capacité d'insertion. En effet, il est impératif, d'une part, d'extraire la marque sans erreur sous peine d'avoir un taux élevé de fausses alarmes. D'autre part, la précision de

la détection des régions de l'image qui ont été manipulées est directement liée à la quantité d'information cachée dans l'image. Il est donc nécessaire de trouver un bon compromis pour la taille de la marque afin de satisfaire à la fois aux deux contraintes : robustesse et sensibilité de la détection. Une des difficultés de dissimuler les attributs caractéristiques de l'image sous la forme d'un marquage réside dans le fait que l'image marquée est légèrement modifiée par l'insertion de la marque elle-même. Bien que ces variations soient imperceptibles à l'œil, elles affectent légèrement les caractéristiques intrinsèques de l'image. De ce fait, les caractéristiques de l'image originale et celles de l'image marquée ne sont plus exactement les mêmes, et on risque alors de détecter des régions altérées alors que l'image n'a pas été manipulée. Ce risque de fausses alarmes est plus ou moins important en fonction du type des caractéristiques choisi et de l'algorithme d'insertion utilisé. Dans [RD00] ce problème a été résolu grâce à un processus de marquage itératif. Ce processus est initialisé en marquant une première fois l'image originale avec ses propres caractéristiques, puis, de manière itérative, on extrait les nouvelles caractéristiques de l'image marquée que l'on insère à nouveau dans l'image originale sous la forme d'un nouveau marquage. Seule l'image originale est marquée pour éviter d'accumuler des distorsions liées au processus de marquage. De cette manière, grâce à ce processus itératif, les caractéristiques contenues dans le marquage coïncident quasi parfaitement avec celles de l'image une fois marquée.

IV.3 Marquage réversible

Une limite évidente à l'utilisation de l'authentification par marquage numérique est la distorsion infligée à l'image hôte par le processus d'insertion. Même si cette distorsion est souvent minime, elle peut ne pas être acceptable dans certaines applications, particulièrement dans les domaines militaire et médical. Il est donc souhaitable de disposer de schémas d'authentification capables de supprimer toute distorsion de l'image après une vérification positive de la marque. Les schémas offrant cette possibilité sont qualifiés de réversibles (ou inversibles). Il est à noter cependant, que la moindre attaque ne permet plus d'assurer la réversibilité du système.

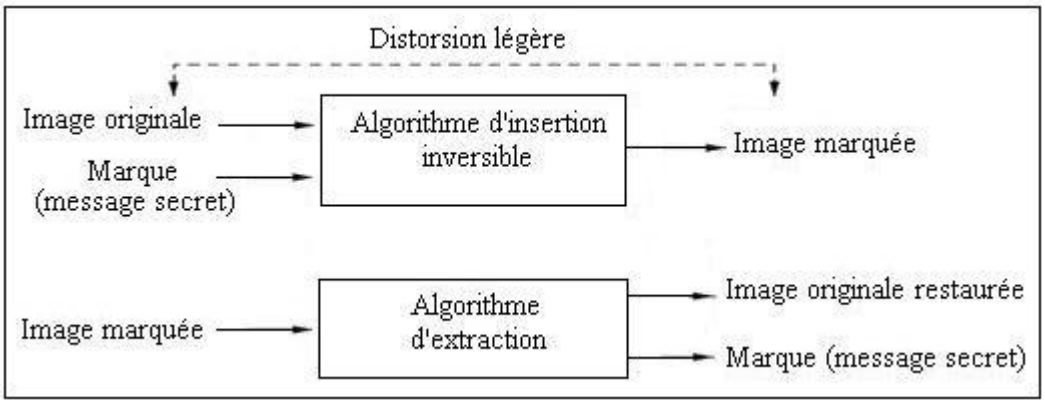


Figure 4.7 Schéma bloc du marquage réversible

On peut classer les schémas réversibles en trois classes :

IV.3.1 Les schémas basés compression

L'approche généralement suivie dans de tels algorithmes, est d'utiliser une certaine forme de compression sans perte sur certaines caractéristiques extraites de l'image, par exemple les LSB, pour libérer de l'espace dans l'image hôte, destiné à contenir les données compressées ainsi que les données d'authentification (hash, MAC ou signature). Pour authentifier l'image reçue, ces données sont décompressées et les données d'authentification sont extraites pour révéler l'image « présumée » originale. A nouveau, des données d'authentification sont extraites de l'image présumée originale puis comparées à celles extraites. Si elles correspondent, ceci signifie que l'image est authentique.
Il est à noter que la moindre attaque ne permet plus d'assurer la réversibilité du système.

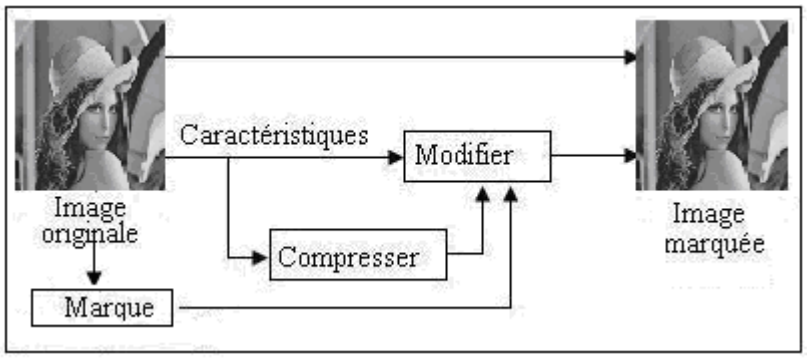


Figure 4.8 Marquage réversible basé compression

Les algorithmes représentatifs de cette classe les plus intéressants sont les deux suivants :

- Dans le schéma proposé par Fridrich, Goljan, et Du [FGD01], d'abord un condensé de 128 bits est calculé à partir de tous les coefficients DCT. De chaque block DCT, certains coefficients des fréquences moyennes sont sélectionnés via une clef secrète et leurs bits les moins significatifs (LSB) sont compressés par une méthode de compression sans perte. La compression est stoppée quand suffisamment d'espace est libéré pour contenir le condensé. Les LSBs compressés sont concaténés au condensé et le tout est inséré aux emplacements des LSB des coefficients sélectionnés.

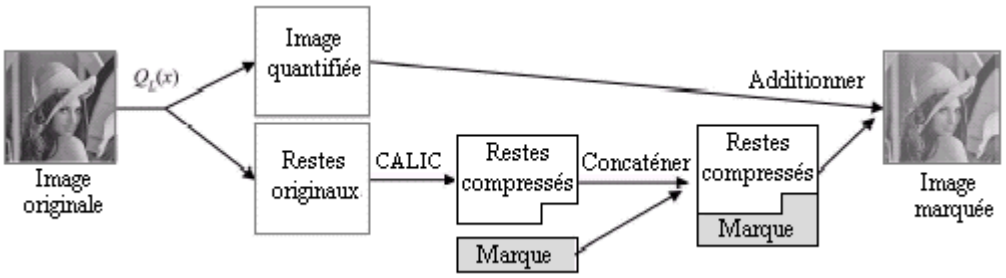
Pour vérifier l'authenticité d'une image, le même protocole est appliqué, pour sélectionner les mêmes coefficients des fréquences moyennes afin d'extraire de leur LSBs le flux de bits compressés ainsi que le condensé. Le flux de bits est décompressé pour rétablir les LSB des coefficients des fréquences moyennes sélectionnés. La même fonction de hachage est utilisée sur tous les coefficients DCT, obtenant un nouveau condensé. Si ce dernier correspond à celui extrait et décompressé, l'image reçue est déclarée authentique.

L'inconvénient de cette méthode est que le condensé ne représente qu'une signature globale de l'image, ce qui signifie que si une attaque locale a lieu sur les coefficients, l'algorithme peut juste dire que l'image est non authentique sans pouvoir localiser les positions attaquées.

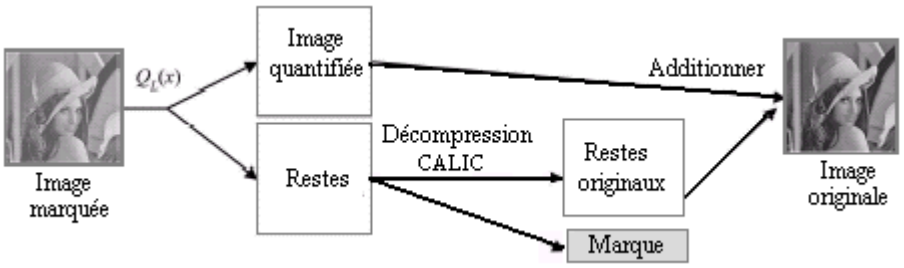
- Le deuxième schéma est celui proposé par Celik et al en 2005 [CSST05] utilisant la quantification scalaire. Chaque pixel de l'image est quantifié par la quantification scalaire suivante :

$$Q_L(x) = L \times \left\lfloor \frac{x}{L} \right\rfloor$$ où L est un paramètre donné et $\lfloor \ \rfloor$ la fonction palier.

Les restes de la division entière sont compressés par une version adaptée de l'algorithme de compression CALIC et la marque binaire représentant le condensé de l'image convertie en base L. Le tout est concaténé puis additionné à l'image quantifiée.
Le schéma global de cet algorithme est donné en Figure 4.9.



(a) Processus d'insertion



(b) Processus d'extraction

Figure 4.9 Illustration du schéma de Celik et al.

Les détails de l'algorithme d'insertion sont donnés à travers l'exemple simple suivant :

Soit l'image originale de 4x4 pixels $I = \begin{pmatrix} 20 & 37 & 7 & 22 \\ 35 & 12 & 32 & 13 \\ 22 & 12 & 18 & 23 \\ 12 & 23 & 12 & 26 \end{pmatrix}$

et soit la marque M = $\{10\ 0010\ 1011\}_2$ représentant le condensé de l'image et le paramètre L=5. L'image quantifiée est alors :

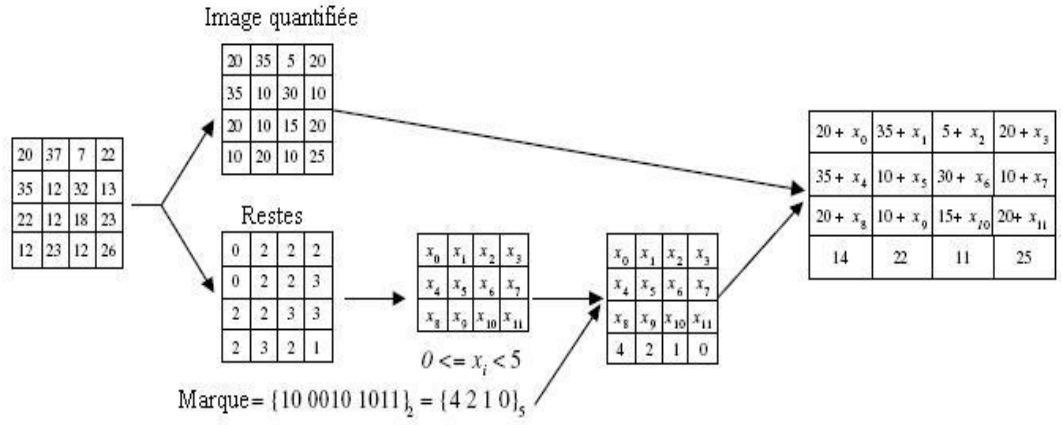
$Q = \begin{pmatrix} 20 & 35 & 5 & 20 \\ 35 & 10 & 30 & 10 \\ 20 & 10 & 15 & 20 \\ 10 & 20 & 10 & 25 \end{pmatrix}$ et les restes $R = \begin{pmatrix} 0 & 2 & 2 & 2 \\ 0 & 2 & 2 & 3 \\ 2 & 2 & 3 & 3 \\ 2 & 3 & 2 & 1 \end{pmatrix}$

L'insertion s'opère alors de la manière suivante :

- Les 16 restes sont compressés par l'algorithme CALIC pour ramener ces 16 chiffres à 12, dénotés respectivement par $\{x_0, x_1, \ldots, x_{11}\}$. Il faudrait, bien sûr, que ces 12 chiffres puissent être décompressés, ultérieurement, pour donner les 16 restes originaux.
- La marque M est convertie de $\{10\ 0010\ 1011\}_2$ à $\{4\ 2\ 1\ 0\}_5$ puis concaténée à la suite des 12 chiffres des restes compressées, pour donner $\{x_0, x_1, \ldots, x_{11},4,2,1,0\}$.

- L'image marquée est obtenue en additionnant ces seize chiffres à l'image quantifiée. Ce qui donne :

$$I' = \begin{pmatrix} 20 + x_0 & 35 + x_1 & 5 + x_2 & 20 + x_3 \\ 35 + x_4 & 10 + x_5 & 30 + x_6 & 10 + x_7 \\ 20 + x_8 & 10 + x_9 & 15 + x_{10} & 20 + x_{11} \\ 10 + 4 & 20 + 2 & 10 + 1 & 25 + 0 \end{pmatrix}$$



Dans la phase de vérification, la première étape du processus d'insertion est de nouveau appliquée sur l'image suspecte. Les 12 premiers chiffres sont décompressés pour recouvrir les 16 restes originaux, et les 4 derniers représentent la marque insérée. La marque est supprimée, et l'image originale est recouverte dans son intégralité. La même fonction de hachage est appliquée sur l'image reconstruite, obtenant un nouveau condensé. Si ce dernier correspond à la marque extraite, l'image reçue est déclarée authentique.

Il est à noter que ce type de marquage réversible a une robustesse très limitée, car toute perte de données compressées entraîne l'échec du processus de vérification.

En dehors de ces deux schémas, Xuan et al. [XZCSNS02] et Van Leest et al. [VVB03] ont aussi présenté des concepts similaires consistant à compresser les données de hautes et moyennes fréquences transformées au préalable par une transformée en ondelettes.

IV.3.2 Les schémas utilisant l'expansion de la différence

Le deuxième type de schémas réversibles, est celui utilisant l'expansion de la différence entre pixels adjacents pour insérer l'information d'authentification secrète. Ces schémas opèrent généralement en générant certaines valeurs représentant les caractéristiques de l'image, puis ces valeurs sont « élargies » de telle sorte à insérer les bits de la marque. Nous introduisons ici les deux schémas les plus représentatifs de cette classe, à savoir ceux de Tian [Tia02] et Wu et Tsai [WT03] et dont nous avons, tour à tour, implémenté une version modifiée dans nos travaux.

- **Schéma de Tian**

Dans ce schéma, d'abord on définit une transformation entière qui opère sur 2 pixels adjacents x et y par :

$$l = \left\lfloor \frac{(x+y)}{2} \right\rfloor$$

La transformation entière inverse peut être représentée par :

$$\begin{cases} x' = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \text{ et} \\ y' = l - \left\lfloor \frac{h}{2} \right\rfloor \qquad \text{où } h = x - y \end{cases}$$

Le processus d'insertion s'effectue alors de la manière suivante :
1) Pour chaque paire de pixels adjacents $\{x_i, y_i\}$, calculer l_i et h_i.
Par exemple, pour x_o=106 et y_o=100, nous avons que $l_o = \left\lfloor \frac{(106+100}{2} \right\rfloor$
et h_o = 106-100=6.
2) Calculer $h'_i = 2 \text{ x } h + w_i$ où w_i est le i^{ème} bit de la marque.
Pour le même exemple, si $w_0 = 0$, alors $h'_0 = 6 \times 2 + 0 = 12$. Si $w_0 = 1$, $h'_0 = 6 \times 2 + 1 = 13$.

3) Obtenir x'_i et y'_i par substitution des paramètres l_i et h'_i dans la transformation entière.
Pour notre exemple on aura :

- Si w_0 =1, $\begin{cases} x'_0 = l_0 + \left\lfloor \frac{h'_0+1}{2} \right\rfloor = 103 + \left\lfloor \frac{(13+1}{2} \right\rfloor = 110 \text{ et} \\ y'_0 = l_0 - \left\lfloor \frac{h'_0}{2} \right\rfloor = 103 - \left\lfloor \frac{13}{2} \right\rfloor = 97. \end{cases}$

- Si w_0 =0, $\begin{cases} x'_0 = l_0 + \left\lfloor \frac{h'_0+1}{2} \right\rfloor = 103 + \left\lfloor \frac{(12+1)}{2} \right\rfloor = 109 \text{ et} \\ y'_0 = l_0 - \left\lfloor \frac{h'_0}{2} \right\rfloor = 103 - \left\lfloor \frac{12}{2} \right\rfloor = 97 \end{cases}$

4) Répéter ces étapes pour toutes les paires de pixels.
5) L'image marquée peut être obtenue par substitution de tous les x'_i, y'_i aux x_i, y_i.

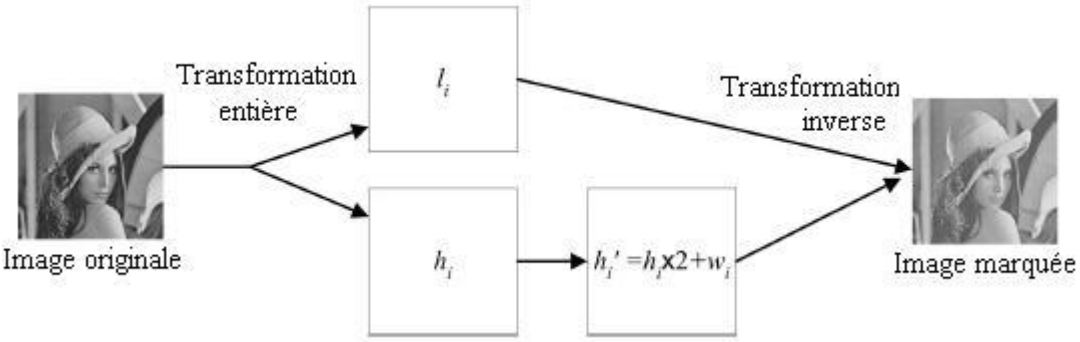


Figure 4.10 Insertion par le schéma de Tian

Il est clair que dans ce schéma, c'est h qui représente les caractéristiques générées à partir de l'image originale et que c'est cette valeur qui est « élargie » pour insérer la marque.

Dans la phase d'extraction, les paires de pixels sont une nouvelle fois formées et les l_i et h_i sont recalculés. Par exemple, si $x'_0 = 110$ et $y'_0 = 97$, alors :

$$l_o = \left\lfloor \frac{(x'_0 + y'_0)}{2} \right\rfloor = \left\lfloor \frac{(110+97)}{2} \right\rfloor = 103 \quad \text{et}$$

$$h'_0 = x'_0 - y'_0 = 110 - 97 = 13 = \{1101\}_2.$$

En ôtant le bit le moins significatif de h'_0, on obtient le bit de la marque $w_0 = 1$ et $h_0 = \left\lfloor \frac{13}{2} \right\rfloor = 6$. Finalement, les pixels originaux sont reconstitués en calculant

$$x_0 = l_0 + \left\lfloor \frac{h_0 + 1}{2} \right\rfloor = 103 + \left\lfloor \frac{(6+1}{2} \right\rfloor = 106 \quad \text{et} \quad y_0 = l_0 - \left\lfloor \frac{h_0}{2} \right\rfloor = 103 - \left\lfloor \frac{6}{2} \right\rfloor = 100.$$

Il faudrait noter que, insérer des données dans l'expansion de la différence, peut conduire à un phénomène de débordement (overflow). Par exemple, si x = 200 et y =10, on obtient $l = 105$, $h = 190$, et $h_0 = 380$ ou 381. La valeur du pixel modifié x' devient alors:

$$105 + \left\lfloor \frac{380}{2} \right\rfloor = 295 \text{ ou } 105 + \left\lfloor \frac{381}{2} \right\rfloor = 295.$$

Il est clair que cette valeur est invalide et un seuillage est alors nécessaire pour éviter le problème de débordement. Cependant, ceci peut conduire à un autre problème. Supposons, par exemple, que le seuil vaut 10 et $h_o = 6$. La valeur de h'_o est alors $h'_o = 12$ ou 13.

Supposons maintenant que, dans la phase d'extraction, la valeur $h'_o = 12$ est obtenue. Deux situations se présentent : $h_o = 6$ (insertion) et $h_o = 12$ (pas

d'insertion). Nous avons donc besoin d'une information supplémentaire nous renseignant si la paire de pixels courante est une position d'insertion ou non.
Pour résoudre ce problème, Alattar [Ala04] proposa d'insérer cette information additionnelle qu'il nomma "location map", à son tour, dans l'image marquée. Nous proposerons à notre tour au niveau du prochain chapitre une autre manière de résoudre le problème d'overflow.

- **Schéma de Wu et Tsai [WT03]**

Ce schéma exploite le fait que le système visuel humain est moins sensible aux changements des pixels de fort contraste qu'aux changements des pixels de faible contraste. Il est donc intuitif qu'on peut cacher plus de bits dans les régions non homogènes de l'image que dans les régions homogènes, à visibilité égale. Il se base sur le calcul de la différence entre les valeurs des pixels adjacents et en fonction de la grandeur de cette différence il est possible d'insérer une quantité plus ou moins grande de bits d'information secrète. Pour ceci, une table de différence est construite pour déterminer le nombre de bits secrets pouvant être insérés dans chaque paire de pixels consécutifs. Le schéma bloc de l'algorithme de Wu et Tsai est décrit dans la Figure 4.11.

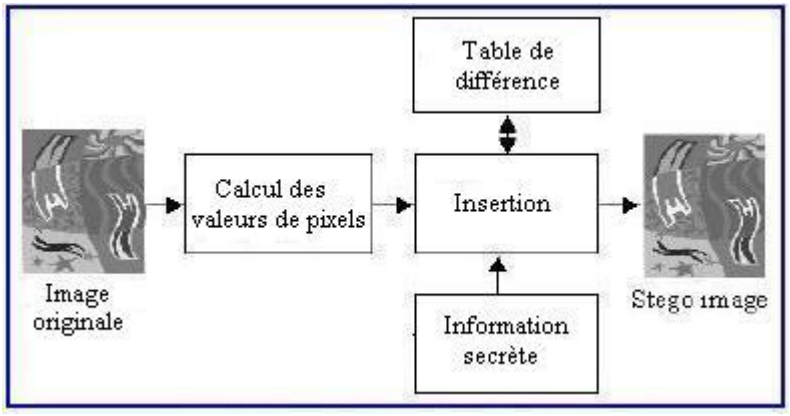


Figure 4.11 Schéma bloc de l'algorithme de Wu et Tsai

Dans une image à niveaux de gris, les valeurs possibles de la différence entre deux pixels consécutifs sont situées dans l'intervalle [-255, 255]. D'abord, toutes les valeurs de différence négatives sont transformées en valeurs positives dans l'intervalle [0…255]. Ces valeurs sont ensuite partagées en six intervalles prédéterminés, la largeur de chacun d'entre eux étant une puissance de 2. Le nombre de bits secrets pouvant être insérés est alors donné par le logarithme base 2 de la largeur de l'intervalle correspondant.

Table 4.1 Table des différences

Intervalle	0-7	8-15	16-31	32-63	64-127	128-255
Largeur	8	8	16	32	64	128
Nb de bits	3	3	4	5	6	7

Les grandes lignes de l'algorithme peuvent être ainsi énoncées :

- L'image est divisée en un ensemble de blocs disjoints de deux pixels consécutifs p_j et p_{j+1}.

- Pour chaque bloc de deux pixels consécutifs p_j et p_{j+1} , calculer la valeur absolue de la différence de leur valeur

$$d_j = \left| p_{j+1} - p_j \right|$$

- Suivant la différence calculée et les intervalles prédéfinis dans la table des différences, le nombre de bits qui peuvent être insérés peut être déterminé. La valeur de d_j nous renseigne sur la nature de la région dans laquelle se trouvent p_j et p_{j+1} :

Si $d_j \approx 0$, alors p_j et p_{j+1} sont dans une région homogène,

Si $d_j \approx 255$, *alors* p_j et p_{j+1} sont dans une région de contours.

- Soit s_j une valeur secrète à insérer dans les deux pixels consécutifs p_j et p_{j+1}. Une nouvelle différence est calculée selon l'équation suivante :

$$d_j^{'} = b_j + s_j$$

où b_j, appelé base, est la borne inférieure de l'intervalle dans lequel le d_j correspondant se situe. Ceci permettra d'assurer que la valeur de la nouvelle différence appartient toujours à l'intervalle d'origine. La valeur temporaire $m_j = d_j^{'} - d_j$ est alors calculée et les pixels originaux p_j et p_{j+1} sont transformées en $\left(p_j^{'}, p_{j+1}^{'}\right)$ pour cacher les bits de sj, suivant les règles suivantes :

$$\left(p_j^{'}, p_{j+1}^{'}\right) = \begin{cases} \left(p_j - \dfrac{m_j + 1}{2}, p_{j+1} + \dfrac{m_j - 1}{2}\right), & \text{si} \quad m_j \text{ est impair et } \left|d_j\right| \text{ est impair} \\ \left(p_j - \dfrac{m_j - 1}{2}, p_{j+1} + \dfrac{m_j + 1}{2}\right), & \text{if} \quad m_j \text{ est impair et } \left|d_j\right| \text{ est pair} \\ \left(p_j - \dfrac{m_j}{2}, p_{j+1} + \dfrac{m_j}{2}\right), & \text{if} \quad m_j \text{ est pair} \end{cases}$$

Il est évident qu'une petite valeur de m_j engendrera moins de distorsion de l'image qu'une plus grande valeur. Quand une paire de pixels est traitée avec succès, cet algorithme garantit que la valeur de la différence modifiée appartient toujours à l'intervalle original.
Il est à noter également que cette technique à une capacité supérieure à celle de Tian, qui se contente d'insérer un seul bit secret par paire de pixels.

IV.3.3 Les schémas utilisant le décalage d'histogramme

Parmi la classe de schémas réversibles utilisant le décalage d'histogramme nous introduisons les deux algorithmes suivants :

- Dans [VDM01], Vleeschouwer et al. proposent le schéma d'interprétation

circulaire dans lequel l'image originale est d'abord segmentée en blocs de pixels voisins, puis les trois étapes suivantes sont appliquées à chacun :

1) Chaque bloc B est aléatoirement partitionné en deux zones Z_a and Z_b, et les deux histogrammes des valeurs de pixels correspondants H_a et H_b sont calculés.

2) Si le bit courant de la marque est 1, on diminue chaque bin dans H_b excepté le plus petit qui est décalé vers le plus grand. Si au contraire le bit courant de la marque est 0, on augmente tous les bins excepté le plus grand qui est décalé vers le plus petit.

3) Répéter ces trois étapes pour tous les blocs.

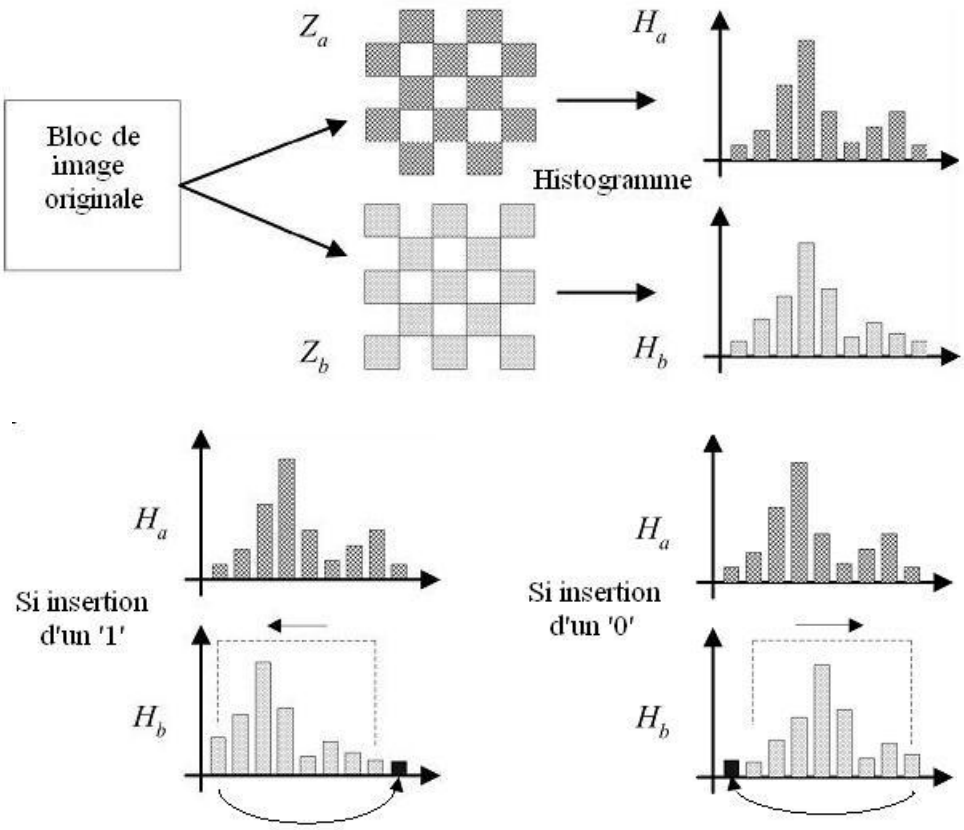


Figure 4.12 Insertion par l'algorithme de Vleeschouwer

Cette méthode tire parti de la constatation, que les pixels voisins d'une image sont dans la plupart des cas, similaires, ce qui entraîne que H_a et H_b soient aussi similaires pour la majorité des blocs de l'image. Dans le cas normal, on peut donc supposer que le pic maximum de Z_a et le même que celui de Z_b. Cependant, dans les cas où le plus haut bin, ou le plus petit, sont décalés vers l'autre versant, ceci peut entraîner une grande distorsion. Par exemple, un fond blanc peut devenir noir.

Les mêmes auteurs ont proposé dans [VDM03] une amélioration de leur algorithme dans laquelle ils proposent une transformation bijective qui décale au plus deux régions de bins et évite ainsi les distorsions extrêmes.

- Dans [VVB03] [03], Van Leest et al. ont proposé un autre schéma de marquage réversible basé sur une fonction de transformation qui introduit des « gaps » dans l'histogramme des blocs de l'image. La fonction de transformation fait correspondre au niveau de gris du pixel en entrée, une valeur unique en sortie de telle façon qu'une ou deux valeurs dans l'intervalle ne sont pas utilisées, donc laissant un ou deux gaps. Les gaps ainsi « dégagés » sont alors utilisés pour cacher la marque. Par exemple, une fonction de transformation possible est celle qui fait correspondre au domaine de [0, 1, 2, …, x, x+1, x+2, x+3, …, x'] l'intervalle [0, 1, 2, …, x, x+2, x+3, x+4, …, x' + 1], laissant la valeur x+1 sans correspondance. Le schéma de marquage opère alors de la manière suivante :

- Pour insérer le bit '1'on incrémente de 1 le niveau de gris de tout pixel de valeur x le transformant en x+1,

- Pour insérer le bit '0', on ne change rien.

On remarquera que le gap correspondant au niveau de gris x+1 est partiellement rempli après le processus d'insertion, et que la capacité d'insertion est déterminée par le nombre d'occurrences du niveau de gris x. En choisissant une fonction donnant plus de gaps, la capacité peut être augmentée au prix d'une plus grande dégradation de la qualité visuelle.
Les emplacements des gaps doivent eux-mêmes être insérés dans l'image pour permettre à l'algorithme de vérification d'extraire l'information d'authentification et rétablir exactement l'image. Les auteurs ont démontré dans leurs expérimentations qu'un taux d'insertion approximatif de 0.06 à 0.60 bits par pixel pouvait être atteint pour un taux de PSNR de 45-50 dB.

L'inconvénient de ce schéma est la nécessité de transmettre dans l'image, l'information supplémentaire renseignant sur la position des gaps. De plus, et vu le temps de calcul insignifiant de l'extraction de la marque, un attaquant peut très bien faire une recherche exhaustive sur les 256 niveaux de gris possibles en supposant à chaque fois que le niveau à l'essai est le gap.

V. Les attaques contre les systèmes d'authentification

Deux catégories d'attaques principales doivent être prises en considération lors de l'élaboration de tout système d'authentification : les attaques ciblant l'image et les attaques ciblant le système d'authentification.

- **les attaques ciblant le contenu de l'image** ont pour but de manipuler l'image sans tenir compte des mesures de protection de l'algorithme d'authentification. Ces attaques peuvent à leur tour être classées en deux types :
  - Les manipulations locales, telles l'ajout ou la suppression d'objets dans l'image;
  - Les manipulations globales, comme le changement d'échelle, le découpage, ou encore l'égalisation d'histogramme.

- **les attaques ciblant le système d'authentification :**

Suivant le type du marquage (fragile/semi-fragile) et le type du système (symétrique/asymétrique) différentes attaques peuvent être montées :

- Une des attaques les plus courantes contre les systèmes à base de marquage fragile, consiste à tenter de modifier une image protégée sans affecter la marque qu'elle contient, ou bien encore à tenter de créer une nouvelle marque que le détecteur considérera comme authentique. Prenons par exemple le cas volontairement simplifié où l'intégrité d'une image est assurée par une marque fragile, indépendante du contenu, insérée dans les LSB des pixels. Il est clair que si on modifie l'image sans se préoccuper de savoir quels sont les bits affectés par la manipulation, on a toutes les chances pour que la marque soit dégradée et l'attaque détectée. Par contre, si on prend soin de modifier l'image sans toucher aux LSB, la marque restera intacte et le système ne décèlera aucune falsification.
- D'un point de vue plus général, dès lors que l'intégrité est assurée par une marque indépendante du contenu de l'image à protéger il est possible d'imaginer une attaque qui recopie une marque valide d'une image dans une autre. C'est la « Copy Attack » de Kutter et al. [KVH00]. De cette manière la deuxième image se retrouve alors protégée. Cette attaque peut très bien être appliquée sur la même image ; dans ce cas, la marque est dans un premier temps retirée de l'image, l'image est ensuite manipulée, et enfin la marque est réinsérée dans l'image manipulée, trompant ainsi le système d'authentification.
- Dans le même esprit, la « Collage-Attack » proposée par Fridrich *et al.* [FGM00] consiste à créer une image contrefaite de toutes pièces à partir d'une banque d'images protégées par la même marque et la même clé. Cette attaque ne présuppose aucune connaissance *a priori* sur la marque binaire cachée, ni sur la clé secrète utilisée. Son principe est relativement simple puisqu'il consiste à remplacer chaque pixel de l'image à manipuler par le pixel qui lui est le plus similaire parmi les pixels de même position des images de la base. La difficulté de cette méthode est de disposer d'une banque d'images suffisamment variées pour obtenir une image falsifiée de bonne qualité visuelle.

  Une autre attaque classique consiste à essayer de trouver la clé secrète utilisée pour générer la marque. Ce type d'attaque, également appelé «Brute Force Attack », est très connu par la communauté « sécurité ». Une fois la clé trouvée, il devient alors très facile pour un pirate de falsifier la marque d'une image protégée avec cette clé. La seule parade efficace est d'utiliser des clés de grande taille de manière à rendre cette attaque très dissuasive en termes de temps de calcul.

Il est clair que les attaques visant le contenu de l'image sont naïves et ne peuvent réussir que dans le cas où l'image n'est pas marquée. Plus sophistiquées, sont les

attaques visant le système d'authentification et dont le but est justement de rendre les attaques visant le contenu indécelables.

Deux autres facteurs très importants intervenant dans la fiabilité d'un système d'authentification sont le taux de faux positifs et le taux de faux négatifs, qu'on désire les plus faibles possible. Le taux de faux positifs est le taux d'occurrence des cas où le détecteur extrait une marque qui n'a pas été insérée. Au contraire, le taux de faux négatifs désigne le taux d'occurrence des cas où le détecteur échoue à extraire une marque réellement insérée. Le taux de faux positifs et le taux de faux négatifs sont en général en conflit avec une faible distorsion de l'image, car assurer de faibles taux positifs et négatifs signifie généralement insérer une plus grande quantité d'information d'authentification, ce qui entraîne immanquablement une plus grande distorsion.

Enfin, les attaques et défis sus-cités ne constituent en aucun cas une liste exhaustive, puisque de nouvelles attaques de plus en plus perfectionnées seront certainement élaborées dans le futur.

Conclusion

Ce chapitre concerne l'utilisation du marquage numérique à des fins d'authentification d'images. Nous avons d'abord introduit la nécessité d'authentifier les images dans cette ère numérique en donnant quelques exemples célèbres de falsification d'images, puis identifié les deux catégories de techniques d'authentification disponibles , à savoir, stricte et souple. Les caractéristiques de chacune ont été comparées et les raisons pour lesquelles l'authentification souple est préférée ont été argumentées.

Nous avons ensuite défini un schéma générique d'un système d'authentification d'image et recensés les critères que ce dernier doit satisfaire pour être efficace.
En fonction de la nature du marquage et des exigences de l'application, les schémas de marquage destinés à l'authentification ont été classés en trois classes principales, à savoir, fragiles, semi-fragiles, et réversibles. Des algorithmes représentatifs de chaque classe ont été décrits en détail puis discutés.

Finalement, nous avons identifié les différentes attaques en montrant comment elles pouvaient être montées et comment elles pouvaient être évitées, puis nous les avons classées en deux catégories : les attaques ciblant le contenu de l'image et les attaques ciblant le système d'authentification.

A la lumière de ces observations, nous pouvons conclure que, dans l'état actuel des recherches, il est difficile d'affirmer quelle approche semble la plus appropriée à assurer un service d'authentification adapté aux images. Il n'existe pas, pour l'instant, de solution universelle répondant parfaitement à tous les problèmes mais plutôt un ensemble de solutions reliées à des applications spécifiques. Les méthodes reposant sur un marquage fragile, sont très sensibles à la moindre altération de l'image, n'offrant par conséquent qu'un service d'authentification stricte, relativement éloignée des besoins des utilisateurs. Néanmoins, les techniques de marquage fragiles ont l'avantage, par rapport aux

méthodes classiquement utilisées en sécurité, de permettre une localisation précise des régions qui ont été manipulées. La tendance actuelle s'oriente, cependant de plus en plus vers l'utilisation de méthodes dites semi-fragiles. Ces méthodes sont beaucoup plus tolérantes vis-à-vis des manipulations bienveillantes, telles qu'une compression Jpeg de bonne qualité. Cette souplesse est rendue possible en partie grâce à des algorithmes de marquage à robustesse ciblée, mais aussi par l'utilisation de données d'authentification de haut niveau, basées sur le contenu sémantique de l'image plutôt que sur les valeurs numériques des pixels.

L'utilisation d'une marque dépendante du contenu de l'image permet, d'une part d'accroître la robustesse de la méthode vis-à-vis d'attaques malveillantes comme la « Collage Attack », et d'autre part, en fonction des caractéristiques choisies, une éventuelle réparation partielle des régions altérées. Le défi majeur de ces méthodes reste celui de trouver un bon compromis entre les exigences de faibles taux de distorsion, de faux positifs et de faux négatifs d'un côté, et la robustesse aux manipulations acceptables d'un autre. Plus intéressantes encore, sont les méthodes réversibles qui permettent de supprimer toute distorsion de l'image après que l'authentification ait eu lieu.

Dans le prochain chapitre, nous aborderons notre approche de l'authentification basée sur le contenu, par le biais de la notion de texture d'image. Cette approche a été fortement inspirée de ce qui se fait actuellement dans certains domaines connexes, comme l'indexation, la classification et la recherche d'images basée sur le contenu (CBIR).

Chapitre 5

Extraction de signature par analyse de texture

Introduction

L'analyse est un domaine très important du traitement de l'image. Elle consiste à extraire un certain nombre de propriétés caractéristiques de l'image et à les exprimer sous forme paramétrique. Les paramètres calculés permettent de décrire, de caractériser, de segmenter et d'analyser les images en question. Il est évident que le choix des paramètres dépend surtout de l'application considérée, par exemple de lier ces paramètres avec les propriétés physiques réelles afin de les quantifier ou alors de trouver des similitudes avec des paramètres de référence afin de les identifier.

Les principales informations dans l'interprétation du message visuel pour un observateur humain sont les couleurs, les contours/formes et les textures. La perception de la texture joue un rôle particulièrement important dans le système visuel humain de reconnaissance et d'interprétation et semble permettre une meilleure appréhension de la profondeur et de l'orientation des surfaces [Arv03]. Il est donc naturel que la texture soit utilisée comme une puissante méthode d'analyse, dans laquelle on extrait des descripteurs locaux, ou parfois globaux, représentant des propriétés discriminantes de l'apparence de la surface de l'image, par des méthodes plus ou moins complexes.

Cette méthode d'analyse trouve des applications de plus en plus nombreuses, comme la segmentation des images satellitaires, la reconnaissance de formes, la classification, la reconstitution 3D d'objets à partir de leur texture (Shape from

Texture) ou la recherche d'images par le contenu dans de grandes bases de données.

La notion de texture est utilisée pour traduire un aspect homogène de la surface d'un objet sur une image. La texture se manifeste donc par une information visuelle qui permet de la décrire qualitativement a l'aide des adjectifs suivants : grossière, fine, lisse, tachetée, granuleuse, marbrée, régulière ou irrégulière. De nombreuses études psychovisuelles ont été faites sur le pouvoir de discrimination de texture par le système visuel humain et une conjecture importante est que l'oeil humain utilise intuitivement les statistiques pour différencier entre deux textures.
C'est précisément ces méthodes que nous voulons mettre en œuvre pour extraire des paramètres de texture, en guise de signature basée sur le contenu, destinée à être insérée dans l'image hôte, pour notre système d'authentification. Il faudra au préalable adapter ces paramètres afin de pouvoir discriminer les manipulations d'images autorisées de celles qui seront interdites.

L'application à laquelle nous destinons ce système d'authentification se situe dans le domaine médical, aussi, nous voulons une marque robuste aux transformations de type géométrique qui font partie des gestes courants dans toute pratique médicale, mais sensible aux transformations plus insidieuses telles que le copier/coller ou le déplacement d'objets. Les descripteurs texturaux sont justement plus ou moins invariants aux changements d'échelle, aux translations et à un degré moindre aux rotations. Il nous incombera donc de bien les paramétrer, afin de les adapter à nos besoins.

D'un autre côté, il faudra dans nos choix, et bien que la signature ne soit pas évaluée en ligne, privilégier les descripteurs qui répondent à ces critères:

- Compactes (pas trop nombreux),
- Stables,
- Rapides à calculer,

Ceci, dans un souci de gain de temps, aussi bien que pour une fiabilité de discrimination.
En nous inspirant de la littérature scientifique sur les textures, sur la recherche d'images par le contenu, sur l'indexation et sur la classification, nous avons finalement retenu les approches par les statistiques de premier et de second ordre sur les niveaux de gis. Insistons sur le terme "niveaux de gris". En effet l'analyse de textures se fait traditionnellement sur des niveaux de gris. On serait très tenté de réfléchir à une méthode qui permettrait de caractériser des textures colorées, cependant, un tel type de caractérisation risquerait fort de faire double emploi avec la caractérisation par couleurs. De plus, l'application à laquelle nous destinons notre système d'authentification, à savoir, l'imagerie médicale, est presque exclusivement monochrome.

Les ressources des différents utilisateurs d'une application médicale peuvent varier significativement. Par exemple, les moniteurs de certains dispositifs peuvent ne pas être en mesure d'afficher l'image dans sa résolution d'origine. Il est alors souvent nécessaire de procéder à un recadrage de l'image pour contourner cette limitation. Pour ceci, notre système d'authentification doit être robuste au changement d'échelle. Etant donné que l'information contenue dans l'image est en grande majorité préservée pour un facteur d'échelle supérieur à un, il est plus aisé de concevoir un système robuste sous cette condition que pour le

cas où le facteur est inférieur à 1. Nous envisagerons donc un facteur d'échelle variant entre 0.5 et 1.
Souvent, les radiologues ont besoin de retourner et/ou translater les images médicales pour une meilleure observation ou pour une comparaison avec des radiographies antérieures. Le système d'authentification doit donc être aussi robuste aux rotations et translations.

I- L'analyse d'image par texture

L'analyse de texture regroupe un ensemble de techniques mathématiques permettant de quantifier les différents niveaux de gris présents dans une image afin d'extraire des propriétés discriminantes de l'apparence de la surface.

Plusieurs définitions de la texture ont été avancées dans la littérature dont nous retenons la suivante : " La texture est un attribut relatif à l'arrangement spatial des niveaux de gris des pixels dans une région de l'image. Cet attribut représente un ensemble de primitives arrangées selon des règles particulières de placement. Une primitive est un ensemble connexe, plus ou moins important, de pixels de niveaux de gris à peu près semblables, constituant un motif de base, qui peut être périodique, quasi-périodique ou aléatoire [Arv03].
Une texture peut être qualifiée de grossière, fine, rugueuse, lisse, directionnelle, granulée, ondulée, régulière, irrégulière ou encore linéaire (Figure 5.1).

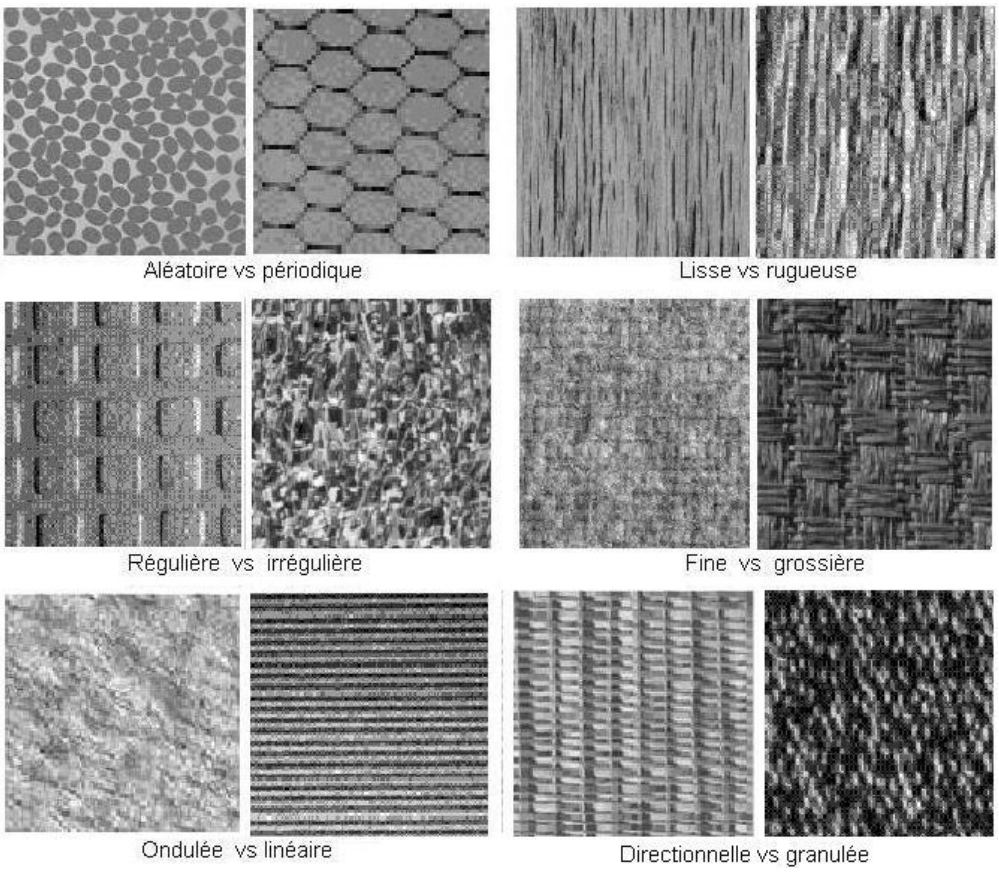


Figure 5.1 Illustration de différentes textures

Une grande variété de techniques d'analyse de texture d'image ont été proposées par différents auteurs [Har79] [GK90] [MM96]. Ces techniques peuvent être classées en trois approches principales :

- L'approche structurelle modélise les primitives qui composent la texture tout en caractérisant les règles d'agencement spatial entre ces primitives. En effet, les textures ordonnées possèdent des primitives qui se répètent dans les images en des positions suivant une certaine loi. Les méthodes structurelles permettent aussi de synthétiser des textures en modifiant ces règles d'arrangement. De telles méthodes semblent adaptées à l'étude de textures périodiques ou régulières.

- L'approche spectrale qui consiste à filtrer l'image à l'aide d'un ensemble de masques spatiaux dans le domaine fréquentiel. La texture est considérée comme un mélange de signaux de fréquences, d'amplitudes et de directions différentes. Les outils les plus utilisés sont les filtres de Gabor et les ondelettes. Les représentations spatio-fréquentielles préservent à la fois les informations globales et locales des signaux. Les textures étant des signaux quasi périodiques ayant une énergie fréquentielle localisée, ces méthodes permettent donc de caractériser la texture à différentes échelles.

- L'approche statistique analyse la distribution spatiale des niveaux de gris en calculant des statistiques locales sur les valeurs des niveaux de gris, qui sont constantes ou varient très peu sur une région texturée. Différentes textures peuvent être discriminées en comparant les statistiques calculées sur différentes sous régions. Ces méthodes sont surtout utilisées pour caractériser des structures fines, sans régularité apparente.

En imagerie médicale a laquelle nous nous intéressons particulièrement, l'étude se fait surtout sur des tissus mous ayant des structures tout à fait aléatoires et le plus souvent non homogènes, c'est pourquoi ce type de méthode sera préférentiellement utilisé.
Nous ne développerons donc ici que les méthodes statistiques qui ont été utilisées lors du développement de notre système.

Les méthodes statistiques modélisent les notions qualitatives usuelles de texture, à savoir, granularité, contraste, homogénéité, répétitivité, fragmentation, orientation, etc. Elles sont utilisées pour caractériser des structures fines, sans régularité apparente.
- La granularité est un trait dominant de la texture et même parfois, par abus de langage, synonyme de texture. Le grain qui donne la granularité est lui-même constitué de pixels voisins possédant le même niveau de gris. La taille et la densité des grains déterminent le niveau de finesse de la texture.
- Le contraste est basé sur le nombre de niveaux de gris et leur taux de variation. Changer le contraste c'est modifier ces paramètres, ce qui modifie la qualité de l'image mais pas sa structure.

- L'orientation est une propriété globale pour une région et traduit la direction générale prise par les motifs ou grains d'une texture.

Suivant le nombre de pixels mis en jeu pour définir la texture locale (un, deux ou plusieurs), les statistiques peuvent respectivement être classées en statistiques de premier ordre, de second ordre ou d'ordre supérieur.
Les méthodes de premier ordre concernent les statistiques sur l'histogramme des intensités des pixels individuels, indépendamment de leur voisinage.
Les méthodes de second ordre prennent en compte non seulement l'intensité du pixel considéré, mais aussi des intensités des niveaux de gris des pixels avoisinants. Elles permettent donc une meilleure discrimination de la texture.
Les méthodes d'ordre supérieur quant à elles étudient les interactions entre plusieurs pixels. Le voisinage est de type mono ou bidimensionnel. La méthode des longueurs de plages de niveaux de gris est la plus souvent utilisée. Elle consiste à compter le nombre de plages d'une certaine longueur j, de niveau de gris i dans une direction donnée. Beaucoup d'auteurs [JDM00], [HSL02], [HR04] s'accordent à dire qu'il est inutile d'évaluer les statistiques d'ordre supérieur, les statistiques de second ordre captant aussi bien l'information texturale. Nous évaluerons donc les paramètres statistiques du premier et du second ordre pour des blocs de l'image en vue d'une caractérisation texturale servant à formuler une signature basée sur le contenu.

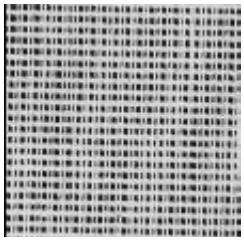
II- Extraction d'attributs de premier ordre

L'analyse par les méthodes de premier ordre se fait au niveau des pixels individuels d'une région de l'image, que nous appellerons région d'intérêt (région of interest ROI). Les paramètres sont calculés à partir de l'histogramme des intensités (ou histogramme du premier ordre). Les paramètres que nous utilisâmes pour caractériser la texture sont les suivants :

1- La moyenne, qui donne la valeur moyenne des niveaux de gris appartenant à tous les pixels de la ROI. Ce paramètre caractérise l'intensité lumineuse de l'image : une image claire aura une moyenne élevée alors qu'une image sombre aura une moyenne faible. Il représente l'emplacement de l'histogramme sur l'échelle des niveaux de gris et est donné par la formule :

$$\mu = \frac{1}{N}\sum_{i,j} g(i, j)$$

où g(i, j) représente la valeur du niveau de gris du pixel (i, j),
et N un facteur de normalisation qui correspond au nombre total de pixels.



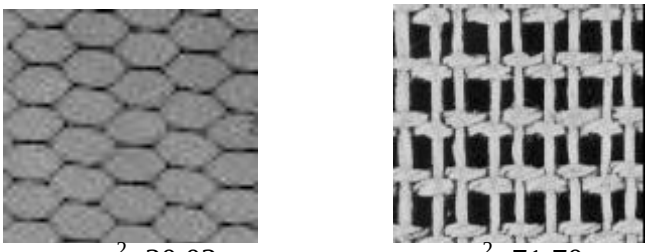
μ=158.43 μ= 78.75

Figure 5.2 Illustration de la moyenne

2- La variance (ou moment d'ordre 2) qui mesure la répartition des niveaux de gris autour de la valeur moyenne. La variance (qui correspond au carré de l'écart type) caractérise le contraste de l'image : elle est faible pour des images peu contrastées et forte pour des images fortement contrastées. Elle est donnée par :

$$\sigma^2 = \frac{1}{N}\sum_{i,j}(g(i,j)-\mu)^2$$

Par exemple, les images ci-dessous correspondent à deux textures ayant la même moyenne (127.7) mais des variances différentes.



σ^2=30.03 σ^2=71.70

Figure 5.3 Illustration de la variance

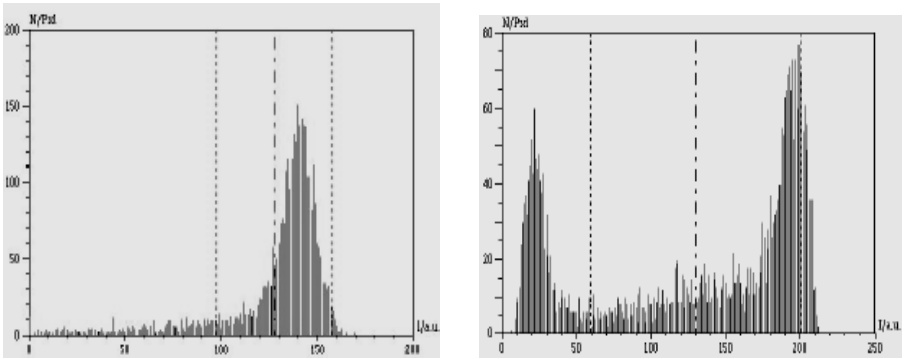
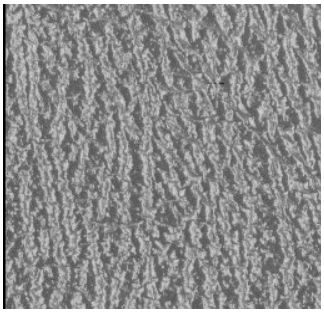


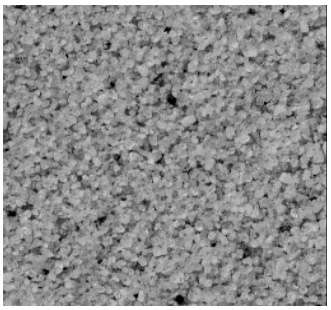
Figure 5.4 illustration de l'écart type

3- Le skewness (ou moment d'ordre 3) qui mesure la déviation de la distribution des niveaux de gris par rapport à une distribution symétrique et reflète donc le degré de symétrie de l'histogramme. Lorsqu'il est nul, l'histogramme est parfaitement symétrique par rapport à la moyenne. S'il est négatif, la pointe de l'histogramme est décalée vers les valeurs inférieures à la moyenne, et s'il est positif, la pointe de l'histogramme est décalée vers les valeurs supérieures à la moyenne. Il est donné par :

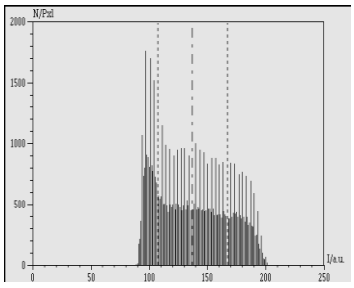
$$SKEW = \frac{1}{N}\sum_{i,j}(g(i,j)-\mu)^3$$



SKEW=0.2474



SKEW =-1.1531



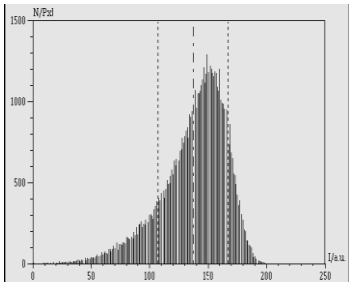


Figure 5.5 Illustration du skewness :

Ces deux images possèdent les mêmes moyenne et écart type, en revanche, leur skewness est différent. La texture de gauche, qui possède un skewness positif, donne un histogramme décalé vers la droite.

4- Le kurtosis qui correspond au moment d'ordre 4 centré autour de la moyenne. Ce paramètre traduit l'apparition fréquente d'un petit nombre de niveaux de gris par rapport aux autres. Il caractérise la forme du sommet de l'histogramme : plus le kurtosis est faible et plus le sommet de l'histogramme est arrondi. Lorsqu'il est élevé, il indique un histogramme en pointe. Il est donné par :

$$KURT = \frac{1}{N}\sum_{i,j}(g(i,j)-\mu)^4$$

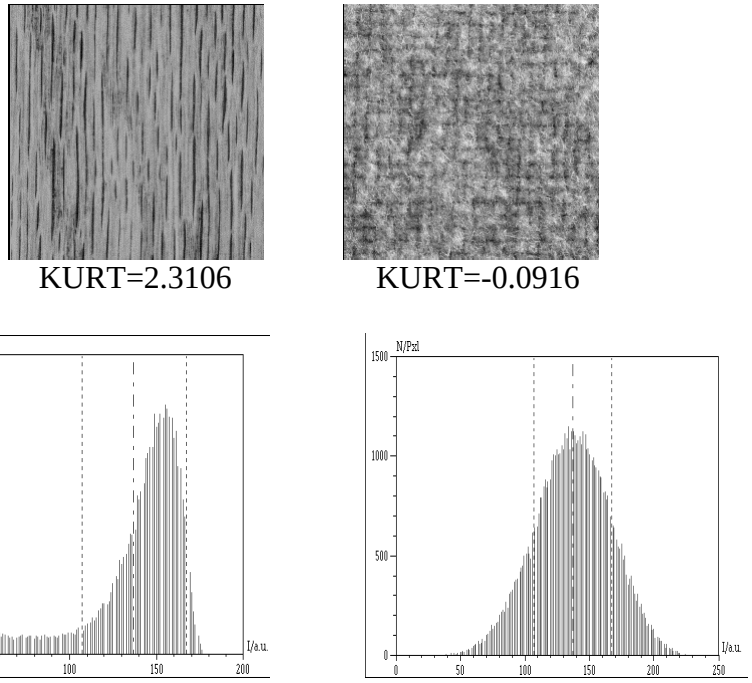


Figure 5.6 Illustration du kurtosis

Les deux textures ci-dessus ont les mêmes moyennes et variances mais des kurtosis différents. On peut constater que la texture au kurtosis le plus élevé possède une distribution décentrée vers les valeurs élevées de niveaux de gris.

III- Extraction d'attributs de second ordre

Dans les méthodes de premier ordre, qui correspondent à une description de l'histogramme des niveaux de gris, il n'y a pas d'informations sur la localisation du pixel. Pour extraire des paramètres plus pertinents, il est nécessaire de recourir à des méthodes d'ordre plus élevé.

Parmi les méthodes de second ordre, Haralick [Har79] suggéra d'utiliser la méthode des matrices de cooccurrence (grey level co-occurrence matrices : GLCMs) pour extraire des statistiques de second ordre, renseignant sur la dépendance spatiale des niveaux de gris.
Dans cette méthode, les fréquences relatives des niveaux de gris de deux pixels séparés par une certaine distance d, dans une direction par rapport à l'horizontale

donnée par un angle θ, sont calculés et rangées dans une matrice P. La taille de la matrice obtenue est Ng*Ng, où Ng correspond au maximum de niveaux de gris de l'image.
La figure 5.7 montre un exemple simple de calcul des P_{ij} à partir d'une petite image 4× 4 composée de quatre niveaux de gris.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Image 4x4

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

P(1, 0°)

4	1	0	0
1	2	2	0
0	2	4	1
0	0	1	0

P(1, 45°)

6	0	2	0
0	4	2	0
2	2	2	2
0	0	2	0

P(1, 90°)

2	1	3	0
1	2	1	0
3	1	0	2
0	0	2	0

P(1, 135°)

Figure 5.7 Exemple d'une image à 4 niveaux de gris et ses 4 GLCMs

Il est à noter que, plusieurs variantes de cette méthode, basées sur le calcul de différences de niveaux de gris existent.
Pour une texture grossière, les valeurs de la matrice sont concentrées sur la diagonale principale. Au contraire, pour une texture fine, les valeurs de la matrice seront dispersées : en effet, pour une telle texture il existe beaucoup de transitions de niveaux de gris. Les statistiques de second ordre permettent donc une analyse texturale efficace et relativement simple, c'est ce qui a fait leur succès auprès de plusieurs auteurs dans beaucoup de domaines de vision par ordinateur tels que la classification, l'indexation, la reconnaissance de formes, ou la recherche d'images par le contenu [MSC96], [MHDG90], [ZSQIT06], [LZ04].

En théorie, il faut calculer toutes les matrices de co-occurrence possibles dans l'image, ce qui est inconcevable. En fait, tous les auteurs s'accordent sur le fait qu'il faut se limiter à une valeur de 1 pour la distance, et des valeurs d'angle de 0°, 45°, 90° et 135°, afin de limiter le nombre de calculs, sans pour autant pénaliser une bonne caractérisation de la texture. Autrement dit, il faut rechercher la co-occurrence de deux pixels voisins dans les orientations horizontale, verticale, ainsi que suivant les deux diagonales.

Une fois la matrice symétrique réalisée, Haralick [Har79] propose d'en extraire 14 indices statistiques différents, renseignant sur la finesse, la directionnalité, la granularité, …etc, de la texture. Ces indices, bien que corrélés, réduisent l'information contenue dans la matrice de cooccurrence et permettent une discrimination précise des textures. Les quatre indices les plus cités dans la

littérature et que nous avons adoptés par conséquent, sont l'inertie, l'entropie, l'homogénéité et l'indice de dissimilarité. Il n'existe pas, à notre connaissance, d'étude rigoureuse pour sélectionner les indices les plus pertinents ou même si certains indices sont plus pertinents que d'autres. Cependant, les indices choisis sont invariants à un décalage des niveaux de gris [Cla02] et supportent par conséquent une éventuelle quantification.

1- L'inertie, ou contraste, correspond à la mesure de l'écart entre les deux pixels formant l'élément constitutif. Il mesure les variations locales des niveaux de gris : si elles sont importantes, alors le contraste sera élevé. Ce paramètre permet aussi de caractériser la dispersion des valeurs de la matrice par rapport à sa diagonale principale. Il est élevé quand les termes éloignés de la diagonale de la matrice sont élevées, c'est-à-dire quand on passe souvent d'un pixel très clair à un pixel très foncé ou inversement. Autrement dit, plus la texture est contrastée, plus cet indice est grand. Il est donné par :

$$Inertie = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-j)^2\, p(i,j)$$

où i, j sont les coordonnées dans la matrice et
p(i,j) les valeurs normalisés de la matrice.

2- L'entropie mesure le degré d'organisation de la texture et varie de façon opposée au second moment angulaire. Elle fournit un indicateur du désordre que peut présenter une texture. L'entropie est faible quand le même couple de pixels se répète souvent, et forte si, au contraire, chaque couple est peu représenté. Elle permet donc de caractériser le degré de granulation de l'image. Plus l'entropie est élevée et plus la granulation est grossière. Elle est donnée par :

$$Entropie = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j)\log p(i,j)$$

3- L'homogénéité ou moment différentiel inverse : Cet indice a un comportement inverse du contraste. Il est d'autant plus élevé que l'on retrouve souvent le même couple de pixels, ce qui est le cas quand le niveau de gris est uniforme ou quand il y a périodicité dans le sens de la translation. Plus la texture possède donc de régions homogènes et plus ce paramètre est élevé. Il est donné par :

$$Homogenéité = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{1}{1+(i-j)^2}p(i,j)$$

4- La dissimilarité : c'est une autre mesure d'homogénéité. Cet indice a un comportement allant dans le sens du contraste : une faible valeur caractérise une texture homogène. Il est donné par :

$$Dissimilarité = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|i-j|.p(i,j)$$

IV. Principe de notre approche

IV.1 Etape de prétraitement

Il n'était pas question d'utiliser les matrices de cooccurrence telles quelles : Premièrement, elles sont de grandes tailles, par conséquent il est difficile d'obtenir une caractérisation rapide et efficace en les utilisant telles quelles. Ensuite, le choix de la taille de la fenêtre de l'image est très important en matière de caractérisation. En effet, les paramètres statistiques sont plus significatifs lorsque la taille d'échantillonnage (taille de la fenêtre) est importante. Toutefois, pour la caractérisation des structures fines, la fenêtre de l'image se veut très petite afin d'assurer une bonne discrimination des structures. Il a été donc décidé, tout comme pour les statistiques de premier ordre, de fixer la région d'intérêt à des blocs 16x16 pixels, d'où l'information texturale est extraite.

Une étape de prétraitement est appliquée à chaque bloc, dans le but de réduire la taille des matrices de cooccurrence, en réduisant l'échelle des niveaux de gris de 256 à 16 niveaux, par un processus de quantification uniforme. Ceci permettra par la même occasion de réduire les temps de calculs. Cette transformation est tout à fait réaliste pour des blocs de taille 16x16 pixels, contenant des valeurs dont beaucoup sont certainement identiques, ce qui donne une image parfaitement fidèle à l'image initiale. De plus, de nombreuses études ont montré qu'il est inutile d'utiliser une matrice de cooccurrence 256x256, les matrices de taille inférieure captant aussi bien l'information texturale [CZ02].
Pour ceci, les 256 niveaux de gris initiaux sont découpés en 16 segments de taille égale à l'aide de la formule de quantification :

$I' = 16I/256$, avec I : l'image originale
et I' : l'image réduite en niveaux de gris.

Ceci permet d'obtenir une matrice de cooccurrence de taille également de 16x16.

Pour chaque bloc, on calcule donc les valeurs de 4 matrices pour une distance de 1 et des valeurs d'angles de 0°, 45°, 90°, 135°.
Les matrices de cooccurrence correspondantes sont construites de façon symétrique, c'est-à-dire, que pour une orientation donnée, il faut comptabiliser la co-occurrence dans les deux directions opposées à la fois. En effet, considérer un sens à la fois, comme constaté chez certains auteurs, n'a aucun sens du point de vue textural, car dire que le pixel i est en cooccurrence avec le pixel j, est équivalent à dire que le pixel j est en cooccurrence avec le pixel i. Le contraire conduit à calculer, en plus, les matrices pour les 4 autres angles, soit 180, 225, 270 et 315 degrés. De plus, pour évaluer des matrices symétriques, il suffit d'évaluer invariablement, la matrice triangulaire supérieure ou la matrice triangulaire inférieure qui sont équivalentes. On peut également faire la remarque que les éléments de la diagonale sont toujours pairs.

Une étape de post traitement est également opérée sur les GLCMs dans le but de stabiliser les valeurs des caractéristiques à extraire contre des manipulations spécifiques, à savoir, les transformations géométriques et l'ajout de bruit gaussien.

Les attributs texturaux sont, par définition, invariants aux translations et changements d'échelle, ce qui nous a tout d'abord incité à les choisir. Pour les rendre également invariants aux rotations, nous avons entrepris de sommer les différentes matrices à travers les quatre directions, comme suggéré par Haralick [Har73], pour obtenir une matrice unique, ce qui réduit par la même occasion l'espace de stockage ainsi que les temps de calculs.

Finalement on normalise les coefficients de la matrice de manière à ce que la somme des coefficients de la matrice vaille 1, représentant les probabilités jointes des pixels deux à deux.
Cette normalisation est obtenue en divisant chaque valeur de la matrice par un facteur égal à la somme de toutes les valeurs distinctes de zéro.

1	2	1	1	3	2	1	4
2	2	1	2	4	3	3	1
2	1	2	2	4	5	5	1
3	2	3	2	2	1	5	3
3	2	1	2	2	1	1	1
2	3	3	1	1	3	3	1
2	1	2	1	2	3	4	2
1	1	2	1	3	4	3	1

Région de 8x8 pixels quantifiée sur 8 niveaux

0							
0	10/67						
0	15/67	8/67					
0	7/67	7/67	6/67				
0	1/67	3/67	4/67	0			
0	2/67	0	1/67	1/67	2/67		
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0

Matrice triangulaire normalisée correspondante (d=1, θ=0)

Figure 5.8 Exemple de normalisation de la matrice

A partir de cette matrice finale, on calcule les indices de second ordre précités.
Cependant un problème peut apparaître. Les éléments du vecteur de description ne sont pas du même ordre de grandeur, ce qui va poser des problèmes. Lors du calcul des distances, certaines dimensions risquent d'être systématiquement privilégiées. C'est pourquoi il a été décidé de les diviser systématiquement par la moyenne de chaque élément de ce vecteur. Ainsi tous les éléments du vecteur ont été ramenés à un ordre de grandeur semblable, à savoir 1.

L'organigramme des traitements peut-être ainsi résumé :

- Partitionnement de l'image en blocs de taille réduite ;
- Pour chaque bloc faire :
  - Quantification du bloc sur 16 niveaux de gris ;
  - Calcul des matrices de cooccurrence ;
  - Sommation des matrices à travers les quatre directions ;
  - Normalisation de la matrice obtenue ;
  - Evaluation des paramètres statistiques du bloc.

IV.2 Génération de la signature :

L'image est d'abord partitionnée en blocs disjoints de taille réduite. Une signature est alors extraite de chaque bloc. Numériquement, cette signature se modélise comme un vecteur de réels (vecteur de caractéristiques statistiques). La taille des blocs choisie est un compromis entre efficacité de caractérisation texturale et temps de calculs. En effet, choisis trop petits, les blocs peuvent préserver plus de détails de texture mais augmentent les temps de calcul. Inversement, augmenter la taille des blocs peut réduire les temps de calculs mais diminuer l'information texturale en augmentant la granularité. Après expérimentations, la taille des blocs retenue a été fixée à 16x16 pixels.

Chaque vecteur de caractéristiques contient les indices statistiques précités (premier ordre ou deuxième ordre) qui sont normalisés de façon à être représentés sur 4x4 octets. Le codeur visite les blocs un à un, évalue les caractéristiques statistiques de chacun, et insère les caractéristiques évaluées au niveau du bloc précédent. Deux méthodes de marquage réversibles et basées sur l'expansion de la différence ont été tour à tour expérimentées : Wu et Tsai [WT03] et Tian [Tia02], présentées en détails dans le chapitre précédent. Ces méthodes ont été adaptées de façon à permettre une insertion bloc par bloc pour rendre possible la localisation d'attaques. Les blocs sont visités dans un ordre pseudo-aléatoire établi par un générateur de nombres pseudo-aléatoires paramétré par une clé secrète. Ainsi, les indices statistiques correspondant à un bloc donné sont insérés dans un autre bloc distant en guise de marque. De cette manière, toute attaque visant le moindre bloc se répercute sur un autre bloc situé plus loin et fait échouer tout le processus d'authentification. Cette façon de procéder permet de localiser les blocs attaqués et empêche les attaques du type couper/coller.

D'un autre côté, tous les vecteurs de caractéristiques sont au fur et à mesure concaténés dans une chaîne unique pour être finalement hachés à l'aide d'une fonction de hachage à sens unique, générant ainsi une signature numérique globale pouvant être utilisée dans un contexte à clé publique si l'infrastructure est disponible. Cette manière de procéder permet de personnaliser le système, suivant les besoins de l'utilisateur, et rendre certaines étapes optionnelles: on peut par exemple se contenter de faire une authentification globale et seulement en cas d'échec demander ou non à faire une authentification locale pour localiser les attaques.

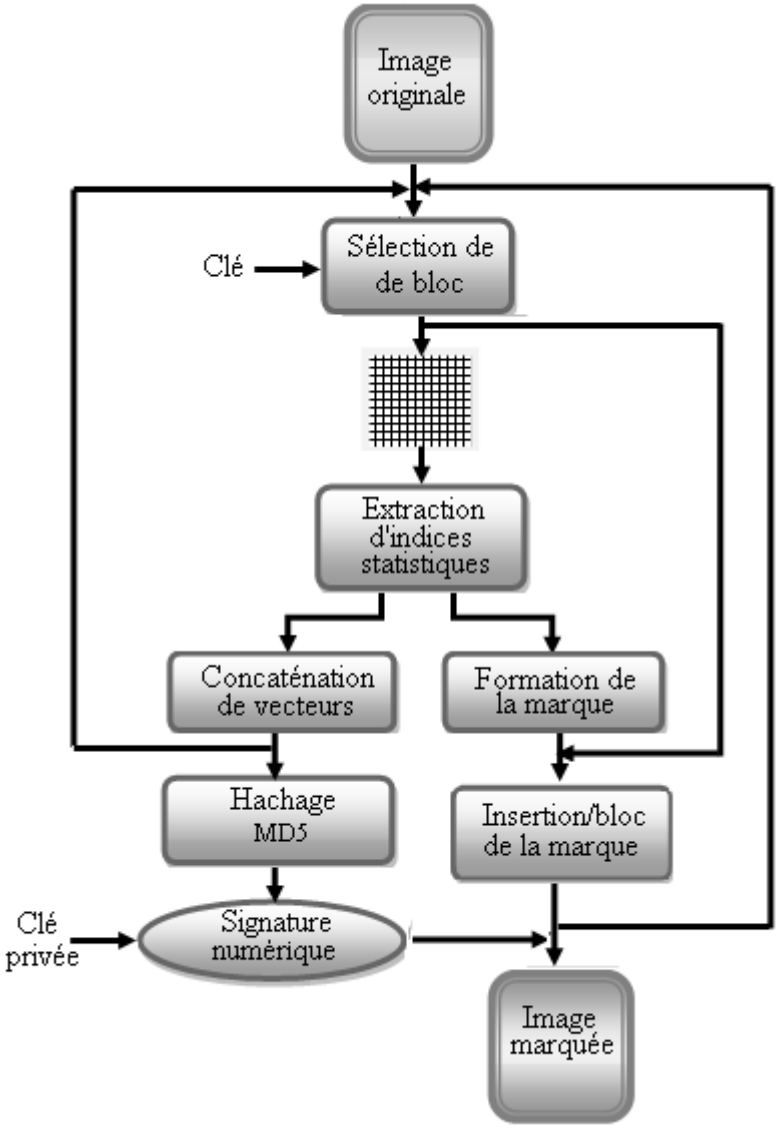


Figure 5.9 Processus de génération de la signature

Pour la fonction de hachage, notre choix s'est fixé sur MD5 [Web1] dont le code source est librement disponible sur la toile, mais d'autres fonctions telles que SHA-1 ou RIPE-MD (voir chapitre 2) auraient très bien pu convenir aussi.

IV. 3 Vérification de la signature et de localisation d'attaques

Remarquons tout d'abord que le système d'authentification opère en mode aveugle : Point n'est besoin de l'image originale pour engager l'authentification d'une image donnée.

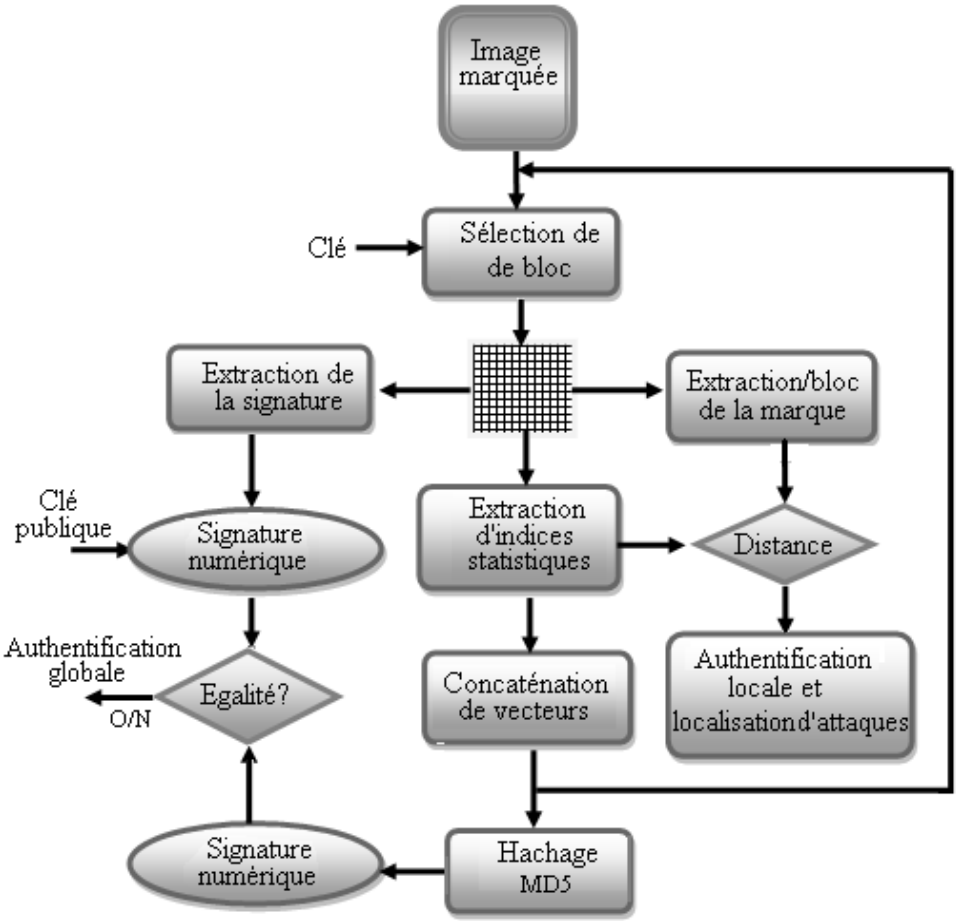


Figure 5.10 Processus de vérification de la signature et de localisation d'attaques

IV. 4 Insertion par la méthode de Wu et Tsai

Bien que la méthode de Wu et Tsai offre théoriquement une grande capacité d'insertion (3 à 7 bits par paire de pixels), dans notre cas, un seul bit sera inséré par paire pour minimiser au mieux la distorsion de l'image. Etant donné que la même quantité d'information sera toujours insérée, indépendamment de l'intervalle courant de la table des différences, cette dernière ne devient plus nécessaire, ce qui constitue une simplification de l'algorithme.
Quatre coefficients réels double longueur (soit du premier ordre soit du second ordre) seront calculés par bloc puis insérés dans un autre bloc aléatoirement sélectionné.
Considérons l'exemple d'une image 512X512 pixels. L'image est partitionnée en 1024 blocs disjoints contenant chacun 128 paires de pixels. Dans chaque différence nous pourrons insérer un bit du vecteur de caractéristiques de premier ou second ordre. Le traitement au niveau de chacun des blocs se fait dans l'ordre décrit dans la Figure 5.11 pour équilibrer au mieux la charge sur les différences.

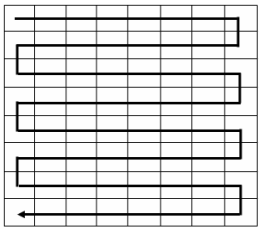


Figure 5.11 Ordre de traitement des blocs

IV. 5 Insertion par la méthode de Tian

La deuxième méthode d'insertion utilisée, également basée sur l'expansion de la différence est aussi réversible et de grande capacité (voir chapitre 4). Cependant, le problème qui se pose avec cette méthode est celui de la possibilité de débordement (overflow/underflow) vers des différences soit négatives soit supérieures à 255 au moment de l'expansion de la différence pour inclure l'information secrète. En effet, la nouvelle différence, après insertion, doit toujours satisfaire :

$$0 \le l + \left\lfloor \frac{h+1}{2} \right\rfloor \le 255 \quad \text{et} \quad 0 \le l - \left\lfloor \frac{h}{2} \right\rfloor \le 255$$

Pour permettre une extraction correcte de l'information secrète, on ne doit marquer que les paires de pixels qui ne souffrent pas de cet inconvénient. Nous avons donc besoin d'une information supplémentaire nous renseignant si la paire de pixels courante est une position d'insertion ou non.
Pour résoudre ce problème, Alattar [Ala04] proposa d'insérer, à son tour, cette information additionnelle, qu'il nomma "location map", dans l'image marquée après l'avoir compressée. Il faudrait noter que cette manière de procéder va diminuer la capacité de la technique tout en augmentant les temps de calcul, à cause du processus de compression.
Nous proposons à notre tour de résoudre le problème d'overflow/underflow d'une manière plus efficace, sans recourir à la location map ou à la compression. Dans la phase d'expansion de la différence, nous représentons la valeur de différence h dans le code binaire refléchi de Gray (CBRG). Ce dernier, est, rappelons-le, un code qui affecte à chacun d'un ensemble d'entiers consécutifs, des mots de codes qui diffèrent à chaque fois par un seul bit. Ceci a pour conséquence d'obtenir une distance de Hamming de 1 entre deux mots de codes adjacents, ce qui est très pratique dans beaucoup d'applications électroniques. Une propriété fondamentale de ce code est que les bits les moins significatifs des mots de codes adjacents sont alternativement 11 00 11 00 11 00 …contrairement au binaire pur pour lequel les mots de codes se terminent alternativement par 0 ou 1.
Cette alternance est exploitée ici pour élargir la valeur de différence sans induire de débordement aux pixels modifiés. Dans notre cas, la phase d'expansion de la différence devient :

- Après avoir calculé h, obtenir son code binaire réfléchi que l'on note h_G;
- Si le bit à insérer w_i = LSB de h_G, alors h' :=h_G

Sinon, si h_G est pair alors h' := h_G + 1 sinon (h_G est impair) h' := h_G – 1.

$$x' = l - h' \quad , \quad y' = l + h'$$

De cette manière, le LSB de h' est « forcé » pour contenir w_i tout en demeurant dans l'intervalle [0-255], contrairement à la méthode initiale ou h était multiplié par deux pour pouvoir insérer wi dans le bit généré par la multiplication, ce qui pouvait provoquer des débordements.
Dans la phase d'extraction, le processus inverse est opéré. On recalcule l' et h', puis on transforme h' dans le code BRCG et on extrait le LSB de h'(représentant wi). Si wi = 0 alors h := h'-1 sinon h := h'+1.
Le reste de la procédure est sensiblement le même.
On peut également noter que la capacité de la méthode va augmenter pour atteindre un bit par paire de pixels.

V. Calcul de distance

Pour mesurer la similarité entre les vecteurs de caractéristiques des blocs à authentifier et la marque qui y est insérée, nous avons utilisé deux mesures de distance couramment utilisées en classification:

- La première est la distance de Hamming normalisée définie par :

$$d_1(v, v') = \frac{1}{k} \sum_{i=0}^{k} \left| (v(i) - v'(i)) \right|$$

où k est le nombre de paramètres (4 dans notre cas) et qui peut également être donnée en termes de comparaison de symboles binaires où $\oplus$ désigne le ou exclusif :

$$d_2(v, v') = \frac{1}{k} \sum_{i=0}^{k} (v(i) \oplus v'(i))$$

Cette mesure donne une valeur tendant vers zéro pour des vecteurs similaires et vers 0.5 pour des vecteurs très dissimilaires. Pour un système d'authentification idéal, plus la manipulation au niveau du bloc est importante, et plus cette valeur va augmenter.

- La deuxième, et la plus classiquement utilisée, est la distance euclidienne, donnée par :

$$d(v_1, v_2) = \sqrt{((v_1(1) + v_2(1))^2 + ((v_1(2) + v_2(2))^2 + ((v_1(3) + v_2(3))^2 + ((v_1(4) + v_2(4))^2}$$

$$d(v_1, v_2) = \sqrt{((v_1(1) + v_2(1))^2 + ((v_1(2) + v_2(2))^2 + ((v_1(3) + v_2(3))^2 + ((v_1(4) + v_2(4))^2}$$

Elle sera également normalisée afin de ramener sa valeur à un même ordre de grandeur que celle de d_1, à savoir, une valeur entre 0 et 0,5.

Un score de similarité $S = \left(\frac{d_1 + d_2}{2} \right) \times 100$ est calculé puis comparé à un seuil.

Si le seuil est inférieur à moins de 5%, le bloc est déclaré avoir passé l'authentification, sinon, l'index de ce bloc servira à la localisation de l'attaque.

VI- Critères de qualité et mesure de distorsion d'une image

Lorsqu'il s'agit d'évaluer la qualité d'une image, deux mesures universelles sont généralement utilisées. Il s'agit des métriques *MSE* et *PSNR*. Ces deux formules sont mathématiquement liées et sont employées couramment de par leur facilité d'implémentation.

Ces deux méthodes sont des mesures pixel à pixel. La métrique *MSE* correspond à la différence quadratique moyenne entre la luminance d'une image *I* et celle d'une autre I' :

$$MSE = \frac{\sum\sum [I(i,j) - I'(i,j)]^2}{N^2}$$

La différence moyenne par pixel est alors donnée par la racine carrée de l'erreur quadratique moyenne ou *RMSE*.

Le PSNR est une mesure donnée en décibels, et se calcule à partir de la mesure MSE de la manière suivante :

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)$$

La métrique *MSE* mesure la différence entre deux images, alors que le PSNR mesure la fidélité entre deux images.

Les figures 5.12 à 5.14 montrent l'impact du marquage respectivement en termes d'aspect visuel, d'histogrammes et de variations du PSNR pour une vingtaine d'images.

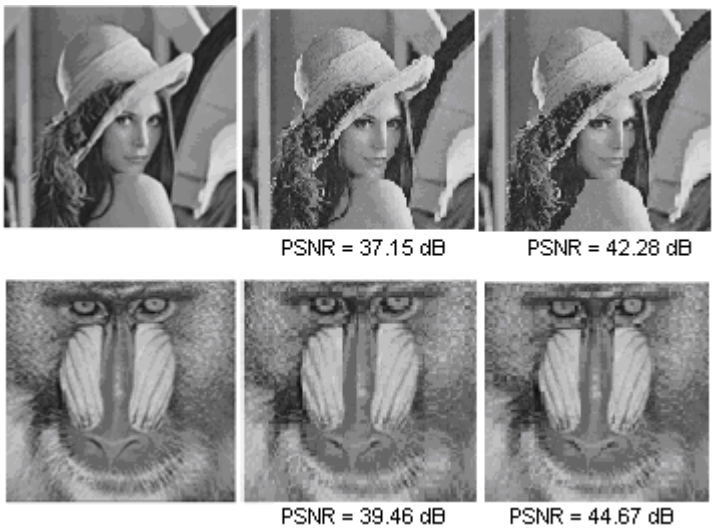


Figure 5.12 Images originales et images marquées respectivement par Wu et Tsai et Tian

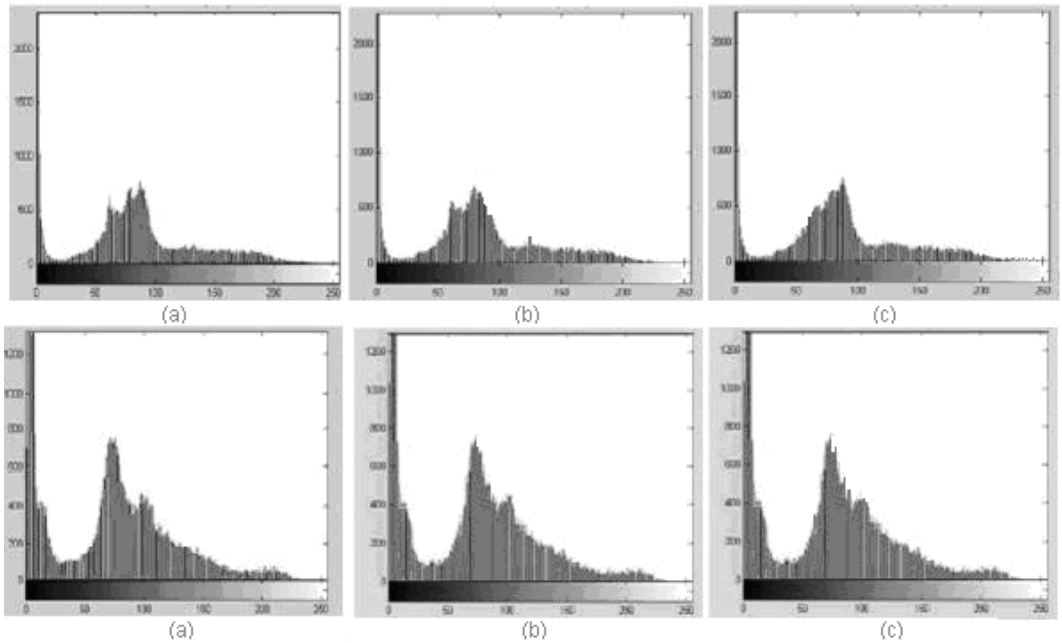


Figure 5.13 Histogrammes de Lena et Baboon (a) : avant marquage, (b) : marquage Wu et Tsai, (c) : marquage Tian

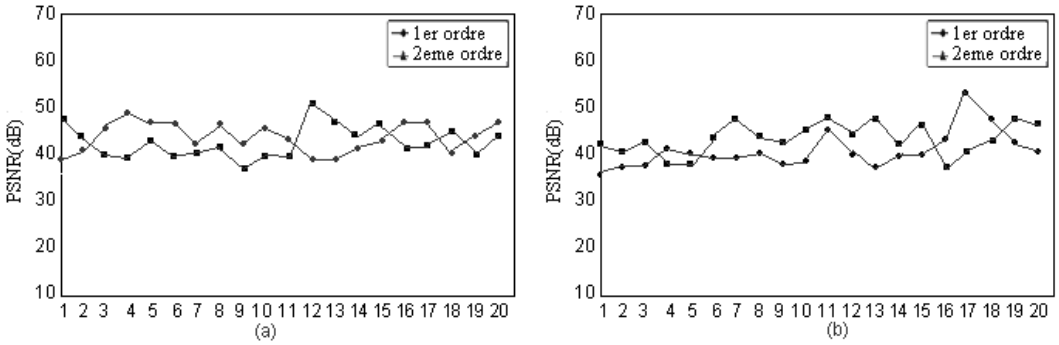


Figure 5.14 Variations du PSNR pour des images marquées par (a) Wu et Tsai, (b) : Tian

Table 5.1 Echantillons d'attributs de premier ordre

Attributs	vecteur1	Vecteur2	Vecteur3	Vecteur4	Vecteur5
Moyenne	59.011	62.755	70.351	58.168	63.455
Ecart type	16.296	16.988	15.410	16.355	16.763
Skewness	3965.59	4661.49	3683.58	4224.99	4562.51
Kurtosis	3.816	4.165	3.6523	3.9878	4.2163

Table 5.2 Distance de Hamming entre vecteurs originaux et vecteurs attaqués (1er ordre)

Attaque	vecteur1	vecteur 2	vecteur 3	vecteur 4	vecteur 5
Echelle 50%	0.0013	0.0016	0.0010	0.0022	0.0019
Echelle 70%	0.0035	0.0280	0.0037	0.0216	0.0084
Echelle 90%	0.0139	0.0617	0.0142	0.0918	0.0224
Rotation 10^o	0.1145	0.1220	0.1275	0.1110	0.0109
Rotation 20^o	0.1410	0.1591	0.1410	0.1591	0.0137
Rotation 30^o	0.2731	0.1591	0.2363	0.2982	0.1704
Translation 10%	0.0111	0.0451	0.0133	0.0505	0.0432
Translation 20%	0.0723	0.0883	0.0825	0.0921	0.0784
Translation 30%	0.0918	0.1103	0.0933	0.1045	0.0853

Tableau 5.3 Echantillons d'attributs de second ordre normalisés

Attributs	vecteur1	Vecteur2	Vecteur3	Vecteur4	Vecteur5
Energie	0.02775	0.003418	0.002849	0.002658	0.004515
Inertie	140.0115	206.7558	171.5411	115.2174	221.9841
Entropie	10.4162	8.2237	7.0154	8.9513	9.6852
Homogenéité	0.5031	0.3826	0.1446	0.3251	0.4222

Tableau 5.4 Distance de Hamming entre vecteurs originaux et vecteurs attaqués (2^e ordre)

Attaque	vecteur1	vecteur 2	vecteur 3	vecteur 4	vecteur 5
Echelle 50%	0.00117	0.00106	0.00163	0.00185	0.00109
Echelle 70%	0.00312	0.00318	0.00256	0.00421	0.00215
Echelle 90%	0.00693	0.00568	0.00456	0.00581	0.00854
Rotation 10^o	0.1045	0.0929	0.0987	0.0752	0.01123
Rotation 20^o	0.1113	0.1092	0.1087	0.0874	0.1186
Rotation 30^o	0.1821	0.1554	0.1162	0.2241	0.1654
Translation 10%	0.0086	0.0159	0.0184	0.0227	0.01321
Translation 20%	0.0722	0.0683	0.0524	0.0832	0.0544
Translation 30%	0.0913	0.0744	0.1003	0.0857	0.0995

Les distances de Hamming normalisées de quelques échantillons sont présentées dans le tableau 5.2 pour les indices statistiques de premier ordre et dans le tableau 5.4 pour les indices statistiques de deuxième ordre.

Plusieurs manipulations malicieuses ont été tour à tour simulées en ajoutant, ôtant ou remplaçant des objets/portions de l'image. Ces corruptions ont toutes conduit à un changement significatif des attributs et par conséquent, à la localisation des attaques. La figure 5.15 montre deux exemples de détection/localisation d'attaque.

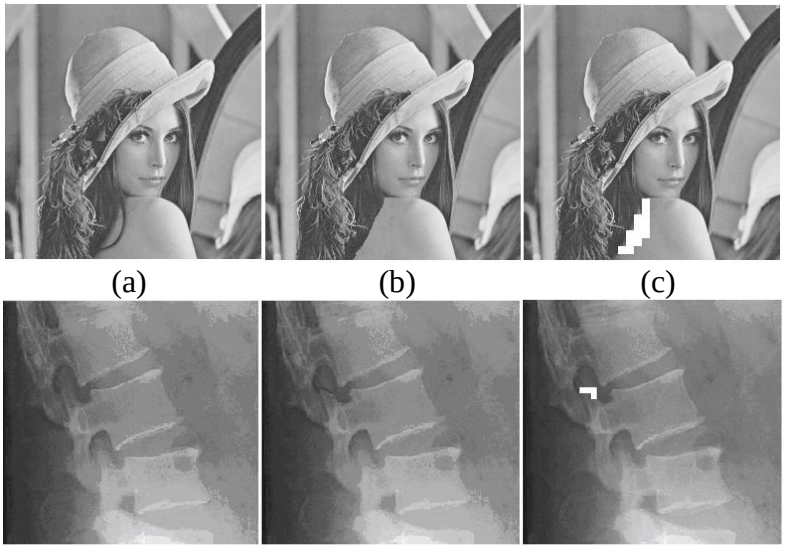


Figure 5.15 (a) Image originale, (b) Image marquée puis attaquée (c) Image vérifiée

La performance du système d'authentification a également été évaluée en termes de Taux de Fausses Acceptations (TFA) et de Taux de Faux Rejets (TFR) suivant les définitions suivantes :

TFA = Nombre de manipulations non détectées / Nombre de manipulations

TFR = Nombre d'images non manipulées détectées non authentiques.

Ces derniers ont étés calculés à travers une collection de 20 images sur lesquelles zéro, une ou plusieurs manipulations de différentes natures sont perpétrées.
La Figure 5.16 montre les variations du taux de fausses acceptations et du taux de faux rejets. On remarque une nette supériorité de l'incidence des statistiques du second ordre sur la performance du système par rapport aux statistiques du premier ordre. De même, on remarque une nette infériorité du taux de faux rejets par rapport au taux de fausses acceptations. Ceci est sans doute dû à la tolérance du système envers les attaques non malicieuses.

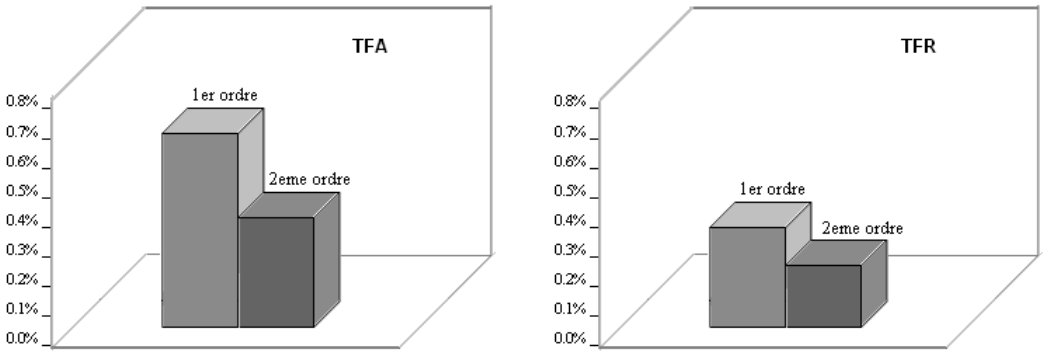


Figure 5.16 Variations du TFA et TFR pour les deux ordres de statistiques

En divisant l'image en blocs de taille réduite et appliquant la vérification sur chaque bloc séparément, on arrive à localiser précisément les attaques.
Il faudrait cependant noter que l'envergure de la manipulation par rapport à la taille des blocs considérés va certainement affecter la performance des résultats. La taille des blocs étudiés ici (16x16) est intentionnellement choisie suffisamment petite pour capturer les manipulations sur une faible à moyenne granularité.

Conclusion

Nous venons de proposer un système d'authentification souple basé sur le contenu sémantique des images. Ce système est capable de détecter les altérations portant sur la texture et d'ignorer les transformations géométriques bénignes tout en maintenant une bonne qualité visuelle de l'image. L'insertion/extraction des données est réalisée bloc par bloc pour permettre la localisation des régions altérées.

A travers une étude sur les méthodes d'analyse d'image par texture, nous avons pu constater la robustesse de certaines d'entre elles aux transformations géométriques. Parmi les méthodes testées, la cooccurrence se démarque par de très bons résultats. Le principal inconvénient de cette méthode est le temps de calcul nécessaire. Du fait qu'elle calcule plusieurs matrices de cooccurrence, assez longues à obtenir, le temps de calcul de l'ensemble de l'algorithme prend des proportions importantes. Cependant l'utilisation d'un ensemble de prétraitements nous a permis de réduire considérablement ce temps d'exécution.
Au terme de cette étude, nous sommes en mesure de proposer une méthode d'authentification souple, capable de localisation, exploitable dans des temps très réalistes. Néanmoins, d'autres tests restent nécessaires pour valider notre algorithme.
Comme champs d'application à notre étude, nous avons pensé à l'imagerie médicale qui nécessite de grandes précautions de sécurité. Un ensemble de solutions seront proposées en ce sens par la suite. Dans ce contexte, nous avons également mis en œuvre un système d'authentification réversible basé compression. Toujours, c'est la simplicité de mise en œuvre et la rapidité des calculs que nous avons privilégié. Nous le présenterons également dans le prochain chapitre.

Chapitre 6

Protection des images médicales

Introduction

Depuis quelques années, nous assistons à un développement massif des techniques d'imagerie digitale dans le domaine de la médecine. La proportion d'images obtenues sous forme digitale croit d'année en année et occupe une place de plus en plus importante dans le diagnostic clinique. L'acquisition des images sous forme numérique, jusque là réservée aux techniques de tomodensitométrie axiale, résonance magnétique et scintigraphie, s'applique aujourd'hui aussi à des techniques traditionnellement analogiques, telles que l'angiographie et l'échographie.

Cette numérisation intensive, qui peut être directe ou indirecte, permet à l'image médicale d'être traitée, formatée, travaillée sur une console informatique, et ensuite d'être remise au patient ou au médecin demandeur, soit sous forme de films, soit sous forme numérique. Avec la numérisation, la télé-médecine ou la transmission des images et des dossiers médicaux entre professionnels de santé et organisations de soins à travers les réseaux, est devenue possible. Dans ces applications, l'image joue un rôle vital, que ce soit pour permettre un diagnostic, faciliter une intervention chirurgicale ou approfondir les connaissances d'une manière générale. Ainsi, aujourd'hui, il est possible, dans certains centres occidentaux que le médecin demandeur reçoive à distance l'examen radiologique de son patient en temps réel à travers les réseaux connectant différents types de systèmes d'information médicaux (SIM). Les infrastructures les plus courantes sont les *Hospital Information Systems* (HIS), *les Radiology Information Systems* (RIS), et les *Picture Archiving and Communication Systems* (PACS). Toutes ces applications ont pour objectifs de faire circuler plus vite, en plus grand nombre les

flux d'informations entre les différents services de l'infrastructure et particulièrement les dossiers médicaux, souvent complexes, des patients. En effet, l'image médicale est rarement transmise seule. Elle doit souvent être mise en perspective avec les autres éléments plus conventionnels du dossier médical : l'historique du malade, ses antécédents, sa pathologie et ses traitements actuels, sans oublier les informations administratives permettant d'identifier de façon univoque patients et données.

Ce mode de circulation d'informations soulève de sérieuses questions de sécurité relatives aux dossiers médicaux, particulièrement face aux exigences des aspects éthiques et légaux propres au domaine médical. Les problématiques soulevées par la généralisation de la numérisation des images médicales sont encore nombreuses et ne peuvent être considérées comme totalement résolues à l'heure actuelle.

Les principaux aspects de sécurité concernés par les dossiers médicaux sont :

- La confidentialité, qui assure que les informations médicales transmises à travers le réseau ou stockées pour archivage ne sont accessibles que par les parties concernées. Les atteintes à la confidentialité sont alors la divulgation d'informations secrète et le re- routage par des parties non- autorisées.
- L'authentification sert à prouver que les dossiers reçus ont bien été émis par la partie déclarée et qu'ils n'ont pas été altérés en cours de route. Une partie non-autorisée peut avoir insérée une fausse image médicale ou un faux document dans le système d'information
- La non-répudiation qui garantit que les partenaires d'un échange de dossier médical ne puissent nier avoir envoyé ou reçu le dossier en question.
- L'intégrité assure que les informations médicales n'ont pas été altérées accidentellement ou frauduleusement pendant leur transfert sur le canal de communication.
- Enfin, la disponibilité nécessite que l'accès aux dossiers médicaux soit toujours possible pour les parties autorisées.

Plusieurs solutions cryptographiques existent pour assurer ces services de sécurité, notamment les techniques de contrôle d'accès qui ont atteint des degrés de maturité très acceptables, mais cette sécurité reste insuffisante devant des tentatives inlassables des attaquants pour accéder aux informations les plus sensibles. Les méthodes de cryptographie seules sont devenues insuffisantes pour couvrir tous les aspects de sécurité concernant le traitement et le transfert numérisé des images et dossiers médicaux. Le marquage numérique peut avantageusement compléter les outils cryptographiques classiques pour protéger les images médicales.

Dans ce chapitre nous allons proposer deux solutions de sécurité pour les images médicales au format numérique, basées sur les concepts avancés dans les chapitres précédents. Ces deux solutions ont pour but de fournir l'authentification de l'image en question, tout en assurant la confidentialité des données patient qui l'accompagnent en les insérant sous une forme chiffrée au sein de l'image en même temps que l'information d'authentification (l'empreinte). Ceci permettra

en plus de faire transiter l'image seule, sans avoir besoin de l'accompagner d'un fichier textuel.
Pour mettre en évidence l'utilité de telles solutions nous avons imaginé un scénario d'utilisation à travers une application de partage d'images médicales sur le Net. Nous la décrirons en détails dans la dernière partie de ce chapitre.
Mais auparavant, nous nous devons d'exposer, même succinctement, les principales modalités d'imagerie médicale existantes, ainsi que les standards les régissant.

I- Les différentes modalités d'imagerie médicale

L'imagerie médicale est le procédé par lequel un médecin peut examiner l'intérieur du corps d'un patient sans l'opérer. L'imagerie médicale peut être utilisée à des fins cliniques, à la recherche d'un diagnostic ou pour le traitement d'un grand nombre de pathologies. L'identification précise de la lésion facilite le recours à la chirurgie, souvent seule solution thérapeutique pour certains malades. De telles techniques permettent également de mieux comprendre le fonctionnement de certains organes encore mystérieux, comme le cerveau.
Les méthodes d'imagerie médicale sont nombreuses et utilisent plusieurs types de procédés physiques dont nous citerons les principales, résumées à partir de [Web19] :

➢ La radiographie

Découverte il y a plus d'un siècle, la radiographie présente un intérêt diagnostique de premier plan dans beaucoup de domaines de la médecine. Elle utilise les rayons X, qui permettent d'imprégner une plaque photographique et ont la faculté de traverser le corps. Plus la densité du corps sera importante, moins le rayon pourra passer au travers, c'est grâce à ce phénomène que l'image obtenue apparaîtra plus ou moins noire. En effet, lors de la radiographie du corps humain, les rayons vont rencontrer soit des tissus, soit des muscles ou encore des os. Les rayons vont aisément passer à travers les tissus qui auront donc une apparence très sombre. A l'inverse, lorsqu'ils rencontreront des os, ceux-ci vont être totalement arrêtés, il n'y aura donc aucune impression sur la plaque et celle-ci restera blanche.

➢ L'échographie

L'échographie est une technique d'exploration de l'intérieur du corps basée sur les ultra-sons. Une sonde envoie un faisceau d'ultrasons de fréquence appropriée (de 3,5 à 10 MHz pour le diagnostic) dans la zone du corps à explorer. Selon la nature des tissus, ces ondes sonores sont réfléchies avec plus ou moins de puissance. Le traitement de ces échos permet une visualisation des organes observés. Lors du passage des ultrasons à travers les tissus, deux facteurs importants conditionnent la formation de l'image: l'atténuation et la réflexion. L'atténuation est causée par la perte d'énergie du système par suite de l'absorption, de la réflexion, de la réfraction et de la divergence du faisceau. Plus l'atténuation est forte et plus le signal de l'écho récupéré sera faible. C'est la réflexion des ondes ultrasonores en direction de la sonde émettrice-réceptrice qui produit l'image dont la texture ou « échostructure » traduit les différences d'indépendance acoustique des différents tissus examinés.

L'échographie permet l'analyse de nombreux organes superficiels (parotide, thyroïde, muscles et tendons, articulations, ganglions, vaisseaux , etc.) ou profonds (foie, vésicule, reins, rate, pancréas, ovaires, utérus, prostate, etc.)

- **Le scanner X**

Le Scanner appelé aussi tomodensitométrie est un examen qui utilise les rayons X pour visualiser un organe par coupes. Comme la radiographie classique, le scanner s'appuie sur l'absorption plus ou moins importante des rayons X selon le milieu traversé, mais au lieu d'être fixe, le tube de rayons X va tourner autour du corps. Le scanner permet de visualiser l'objet par tranches successives de quelques millimètres d'épaisseur chacune, alors qu'une radiographie ordinaire n'offre "qu'une vue en projection" du volume irradié. L'appareil balaye la section examinée avec un faisceau étroit de rayons X et enregistre, pour chaque position du faisceau, l'intensité transmise. Pour recueillir suffisamment d'informations, le balayage du plan doit se faire en plusieurs fois, sous des angles différents. En répétant l'opération sur plusieurs coupes successives, on en construit une image X tridimensionnelle.
Dans la plupart des cas, un produit de contraste à base d'iode est utilisé pour améliorer la qualité de l'image. Cet examen présente l'avantage de donner des informations très précises sur les organes étudiés.

- **Le Doppler-échotomographie**

Le Doppler-échotomographie, renseigne sur la morphologie des vaisseaux sanguins mettant en évidence occlusions ou rétrécissements de calibre; il n'offre pas un enregistrement sonore ou graphique, mais une image : Une sonde à ultrasons est placée en regard du vaisseau à examiner, et les ultrasons émis sont renvoyés et transformés en sons qui varient selon la vitesse du sang. Les ultrasons se réfléchissent sur les différents tissus qu'ils rencontrent ; les échos ainsi renvoyés sont transformés en points lumineux dont l'éclat est proportionnel à l'énergie réfléchie;

- **Imagerie par Résonance Magnétique (IRM)**

L'imagerie par résonance magnétique permet de visualiser de manière très précise des détails invisibles sur les radiographies standard, l'échographie ou le scanner, et est utilisée pour analyser à distance des organes tels que le cerveau, la colonne vertébrale, les articulations et les tissus mous. Cette technique, très complexe, fait appel aux propriétés magnétiques des noyaux des atomes, en particulier, l'hydrogène. Placés dans un puissant champ magnétique, tous les atomes d'hydrogène s'orientent dans la même direction : ils sont alors excités par des ondes radio durant une très courte période (ils sont mis en résonance). A l'arrêt de cette stimulation, les atomes restituent l'énergie accumulée en produisant des signaux qui sont enregistrés et traités par un système informatique. Un système de codage spatial permet de faire une cartographie de ces signaux, et les mesures recueillies permettent de constituer un plan de coupe sélectionné de la zone étudiée, qui peut être restituée en deux ou trois dimensions.

- **La scintigraphie**

Une scintigraphie est un examen de médecine nucléaire permettant de faire des images du corps humain par injection dans une veine d'un produit légèrement radioactif. Le produit peut mettre un certain temps à se fixer suivant l'organe à observer. L'appareil, appelé gamma caméra, capte les signaux émis par le produit, fixé de façon différentielle dans le corps.

- **La tomographie à émission de positons**

La tomographie à émission de positons (TEP, ou PET en anglais) est un examen difficile à pratiquer et très coûteux. Elle repose sur la disponibilité d'un accélérateur de particules, permettant de préparer et de rendre immédiatement disponibles les produits radioactifs nécessaires. Des isotopes radioactifs injectés en quantité minime par voie artérielle émettent des rayonnements captés par des caméras spécifiques permettant la reconstruction d'images par ordinateur. L'intérêt principal de cette méthode est que les isotopes peuvent être liés à des composés chimiques naturels dans le corps humain et que l'on peut observer dans une certaine mesure la répartition de ces composés ce qui donne une image du fonctionnement du corps. Cet examen utilise une grande variété de marqueurs, qui permettent d'étudier différentes fonctions cérébrales et différents métabolismes.

Le tableau 6.1 suivant, forcément incomplet, résume les principales méthodes d'imagerie, le principe physique employé, le type d'images ainsi que les parties du corps que l'on explore.

Tableau 6.1 Principales modalités d'imagerie médicale [Web19]

	Nom	Procédé	Images	Exploration
Radiographie	Radiographie	Rayons X	Projection	Poumons, abdomen, squelette, seins
	Radiographie numérique	Rayons X	Projection	Poumons, abdomen, squelette, seins
Ultrasonographie	Échographie	Ultrasons	Coupes	Abdomen, cœur, seins, muscles et tendons
	Döppler	Ultrasons	Coupes	Vaisseaux sanguins
Scanner ou tomodensitométrie	Scanner	Rayons X	Coupes	Toutes
	Scanner hélicoïdal	Rayons X	Coupes	Toutes
	Scanner "multicoupe"	Rayons X	Volume	Toutes
Imagerie par résonance magnétique	IRM	RMN	Coupes	Toutes ou presque
	Angio-RM	RMN	3D	Vaisseaux sanguins
Imagerie vasculaire	Artériographie	Rayons X	Projection	Vaisseaux sanguins
	Angiographie	Rayons X	Projection	
	Angiographie numérisée	Rayons X	Projection	
Médecine Nuclaire	Scintigraphie	Émission de rayonnement	Projection	Toutes ou presque
	Tomographie par émission de Positons	Positons	Coupes	
	SPECT		Coupes	Cerveau

Afin permettre l'interopérabilité entre les systèmes d'informations médicaux, plusieurs formats standards ont été pensés, permettant notamment de représenter et d'enregistrer les images médicales sur support numérique ainsi que toutes les informations textuelles associées. Parmi ces standards, la norme DICOM (Digital Image Communications in Medicine) [web20] a été développée en 1992 par l'ACR (American College of Radiology) et NEMA (National Electrical Manufacturers Association) afin de faciliter l'interconnexion des systèmes d'imagerie médicale aux réseaux.
L'ACR a également fixé quelques recommandations techniques pour les systèmes de télémédecine, visant à améliorer l'affichage de l'image médicale [Web18]:
Parmi ces recommandations on trouve :

- Images de petit format – CT, IRM, ultrasons, et médecine nucléaire:
  - Acquisition ou numérisation : au moins 500 x 500 pixels, définition sur 8 bits,
  - Affichage : au moins 500 x 480 pixels, définition sur 8 bits.
- Images de grand format - films radiographiques :
  - Acquisition ou numérisation : une résolution spatiale minimum de 2,5 paires de lignes/mm et une acquisition en 1024 niveaux de gris (10 bits),
  - Affichage : une résolution spatiale minimum de 2,5 paires de lignes/mm et une acquisition en 256 niveaux de gris (8 bits).

II. CC-MARK: Un système de compression-chiffrement-marquage d'images [BLB 06]

La première solution de sécurité que l'on propose va allier chiffrement, marquage et compression de l'image dans un même algorithme. Pour que les informations médicales ne soient accessibles qu'aux parties autorisées il faudrait les rendre illisibles, donc chiffrées avant d'être transférées. De plus, pendant le transfert des données, il ne faut absolument pas qu'une image soit dissociée du nom du patient concerné pour éviter toute confusion d'appartenance à la réception de celle-ci. Le marquage numérique peut apporter une solution à ce problème en enfouissant les données patients au sein même de l'image. Cependant, l'incrustation de la marque peut entraîner une distorsion plus ou moins importante de l'image. Si cette distorsion est souvent minime, elle reste inacceptable dans le cas des images médicales ou la moindre modification risque de mener à des diagnostiques erronés. Les méthodes de marquage qualifiées de réversibles seront donc préférentiellement utilisées.

D'autre part, les images médicales numérisées, posent par leur taille importante, de nombreux problèmes quant à leur transmission ou à leur stockage. Pour gagner aussi bien en vitesse qu'en place, il est il est souvent nécessaire de faire une compression de l'image pour pouvoir l'utiliser dans une application quelconque de télémédecine.

On se propose donc de réaliser une application de traitement d'images médicales qui assure ces trois fonctions, à savoir, la sécurité de l'image, la confidentialité des données patient et la vitesse de transfert de ces dernières. L'application en question retourne des images compressées par la méthode RLE, marquées par des informations concernant le patient en utilisant la méthode LSB réversible et chiffrées par une clé privée en même temps pour les transmettre via le réseau (éventuellement Internet).

Le problème qui se pose est comment réaliser toutes ces opérations tout en restant performant afin d'être réellement utilisable ?

L'ordre logique des différentes taches dans cette application, consiste d'abord à compresser l'image pour réduire la quantité d'informations, la marquer ensuite par une marque spéciale, puis la brouiller par un algorithme de chiffrement **:** c'est un algorithme séquentiel à trois passes, et il est évident qu'il est peu performant.

Pour améliorer les performances de l'application, on se propose de remplacer les traitements consécutifs de compression, marquage et chiffrement par l'utilisation d'un seul et même algorithme réalisant les 3 taches en une seule passe : il s'agit de l'algorithme "**CC-MARK**". Nous démontrerons qu'il est plus avantageux d'utiliser cet algorithme à travers de nombreux résultats expérimentaux.

Les exigences de notre application sont la vitesse d'exécution, la sécurité, la réversibilité et la transparence de la marque. De plus, l'extracteur devra opérer en mode aveugle.

Les 3 algorithmes retenus pour les différentes tâches sont intentionnellement choisis simples et de faible coût puisque le but est de démontrer la supériorité des performances en temps de calculs de l'algorithme combiné sur celles des trois

algorithmes appliqués consécutivement plutôt que de démontrer l'efficacité de chacun des algorithmes. De plus, des algorithmes à faible coût sont hautement recommandés dans toute application distribuée.

II.1 Méthode de compression :

Notre choix s'est porté sur la méthode de compression réversible de codage de plages ou RLE (Run Length Encoding). Cette méthode agit en remplaçant les chaînes de répétition d'un même symbole par une seule instance de ce symbole suivie par un compteur du nombre de répétitions. Elle est particulièrement efficace pour les fichiers binaires qui alternent de longues suites de zéros et de uns. Les taux de compression reportés dans la littérature sont de 40 à50 %. Ici, il est attendu que cette méthode exploite la redondance naturelle contenue dans les images, particulièrement celles contenant de larges zones d'un même niveau de gris.
Dans notre cas, nous avons adapté la méthode de la manière suivante :

L'image est parcourue par blocs de 8 pixels consécutifs: les pixels dans le bloc courant peuvent être identiques (I-block) ou différents (D-block).
- Si le bloc courant est un D-block les pixels sont codés séparément sur 8 octets précédés par un octet signalant qu'il s'agit d'un D-block.
- Si au contraire c'est un I-block, un autre bloc de 8 pixels est accepté, et ainsi de suite jusqu'à la rencontre d'un D-block. Tous les I-blocks consécutifs sont finalement codés sur 3 octets :
    - Le premier est un indicateur de un ou plusieurs I-blocks;
    - Le second indique le nombre de blocs identiques consécutifs;
    - Le dernier indique la valeur du niveau de gris courant.

Il est vrai que le processus de codage va augmenter la taille du fichier quand un D-block est traité, (9 octets au lieu de 8), mais on obtient une compression significative quand plusieurs I-blocks consécutifs sont codés dans un bloc de 3 octets.

II.2 Méthode de chiffrement :

Pour cette phase, nous proposons une méthode de chiffrement simple et efficace, qui s'effectue sur le modèle du masque jetable, opérant en mode CBC (Cipher Block Chaining). Le modèle du masque jetable est réputé pour être inconditionnellement sûr (voir chapitre 2), alors que le mode CBC assure que des blocs en clair identiques donneront des blocs chiffrés différents, ce qui empêche une cryptanalyse éventuelle sur la base d'une analyse statistique sur le texte chiffré. Nous utiliserons dans ce qui suit le terme « xorer » pour désigner une opération de ou exclusif sur les chaines de bits.
Le chiffrement s'opère donc par blocs : un bloc de données de taille fixe est transformé en un autre bloc de données par utilisation d'une fonction de transformation et d'une clé. Un bloc de texte chiffré s'obtient d'abord en "xorant" le bloc en clair avec le bloc chiffré précédent puis en chiffrant le résultat. Seul le premier bloc sera xoré avec un vecteur d'initialisation (VI), qui représente une valeur secrète partagée par l'émetteur et le récepteur. Les pixels seront codés par

blocs de trois : d'abord une chaine de 24 bits est aléatoirement générée comme vecteur d'initialisation de même qu'un masque également de 24 bits.
Le vecteur d'initialisation aussi bien que les trois pixels à chiffrer sont convertis en trois nombres ASCII et sont additionnés deux à deux. Le résultat modulo 2 est xoré avec le masque et le résultat est retenu comme entrée pour chiffrer les trois prochains pixels. Le processus est répété jusqu'à traiter les trois derniers pixels complétant par des zéros si nécessaire. La table 6.2 montre un exemple de chiffrement d'un bloc de trois pixels.

Tableau 6.2 Phases de chiffrement d'un bloc de 3 pixels

3 pixels	01110011	11000011	01100010
ASCII	115	195	98
VI	65	108	76
Add	180	303	174
Mod 256	180	47	174
Binaire	10110100	00101111	10101110
Masque	00011101	1100001100	10101011
Xor	10101001	11100011	00000101

Dans la phase de déchiffrement, le récepteur commence par xorer les trois valeurs ASCII consécutives avec le masque généré par la même clé secrète que celle de l'émetteur. Ensuite, il substitue de ce résultat les trois codes ASCII du vecteur d'initialisation si c'est le premier bloc, ou bien le mot de code précédent pour les autres blocs, en ajoutant 256 si la valeur obtenue est négative.

II.3 Méthode de marquage

Pour enfouir les données patient à l'intérieur de l'image correspondante nous avons choisi le domaine spatial et la technique LSB (Least Significant Bit), d'abord pour sa simplicité donc rapidité, ensuite pour sa grande capacité et la faible distorsion induite (voir Chapitre 3). En effet, les bits les moins significatifs de la valeur des pixels sont généralement considérés comme étant du bruit généré par le système d'acquisition et peuvent par conséquent être changés sans incidence sur la qualité de l'image. Pour le côté sécurité, la substitution peut être effectuée sur n'importe lequel des LSB, choisi au moyen d'une clé secrète. L'inconvénient principal de cette méthode et sa sensibilité à toute compression avec perte telle la compression JPEG. Nous comptons détourner ces inconvénients à notre avantage, d'abord en utilisant une méthode de compression sans perte (RLE), ensuite en ne modifiant même pas les LSB des pixels de l'image, mais seulement les LSB des indicateurs de type des blocs ce qui empêchera une éventuelle tentative de stérilisation.

II.4 Processus combiné

L'ordre naturel dans lequel les trois opérations devraient être réalisées est de d'abord compresser l'image, de la marquer ensuite avec les informations patient, puis finalement la chiffrer. En effet, le fait de commencer par la compression, va diminuer la taille des données à marquer et chiffrer et, par conséquent, aboutir à un gain en temps. De plus, si on procède au chiffrement avant la compression, les propriétés statistiques de l'image chiffrée vont limiter la compressibilité de l'image. Nous maintiendrons donc cet ordre de traitement dans le processus combiné et placerons les opérations de l'insertion et chiffrement à la fin de la boucle de compression.

Chaque étape de compression produit des blocs de taille 3 ou 9 octets selon que le bloc courant est un I-block ou un D-block. Chaque fois qu' un bloc est compressé nous allons remplacer le bit le moins significatif (BLMS) de l'indicateur du type de bloc (I ou D) par un bit de l'information patient. En effet, puisque cet octet est juste un indicateur binaire, seul le bit le plus significatif (BLPS) a de l'importance, le reste peut être donc être modifié sans avoir de conséquence sur le processus de décompression.

L'étape de chiffrement est appliqué sur le I/D-block obtenu lors de l'étape de compression de la façon suivante : S'il s'agit d'un I-block, ses trois octets seront chiffrés de la même manière que dans le processus séquentiel. Si, au contraire, il s'agit d'un D-block, le chiffrement est appliqué 3 fois de suite sur les 9 octets de ce D-block. Le processus de chiffrement est ainsi itérativement poursuivi jusqu'aux 8 derniers pixels.

L'expérimentation porte sur un ensemble d'images médicales (en grande partie des radiographies) collectées principalement à partir du web. Pour illustration nous présenterons quelques résultats sur les images échantillons montrées sur la

Figure 6.1. La première est une image IRM de 128x128 pixels, la seconde est une image CT-scan de 256 x 256 pixels et la troisième une image rayons X de 512x512 pixels.

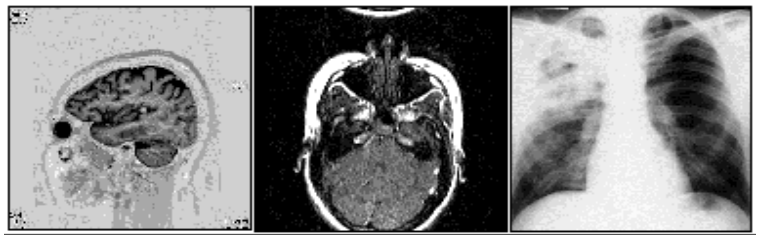


Figure 6.1 Images échantillons originales

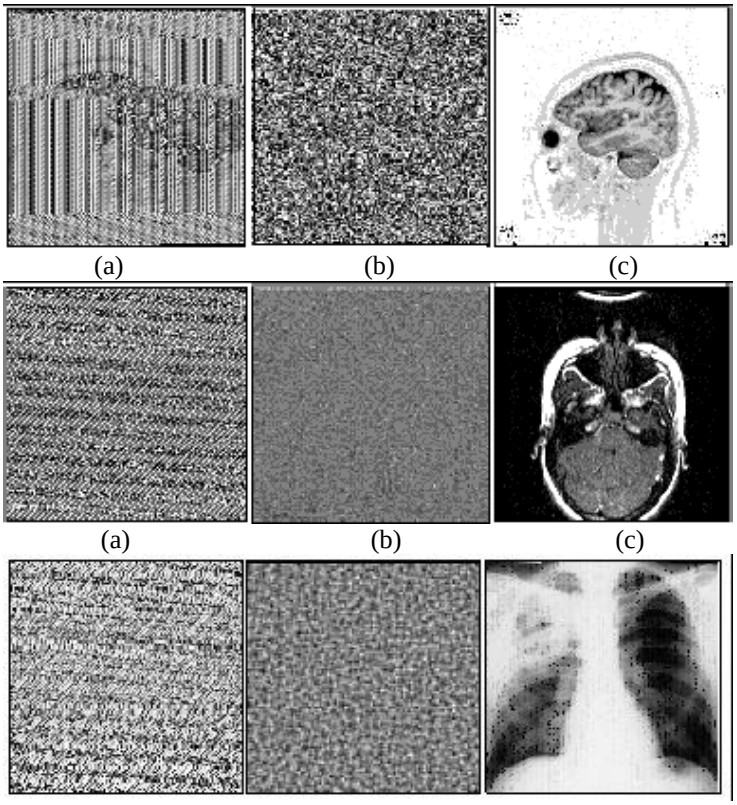


Figure 6.2 Résultats sur les images échantillons traitées par (a) l'algorithme séquentiel, (b) l'algorithme imbriqué, (c) images décodées

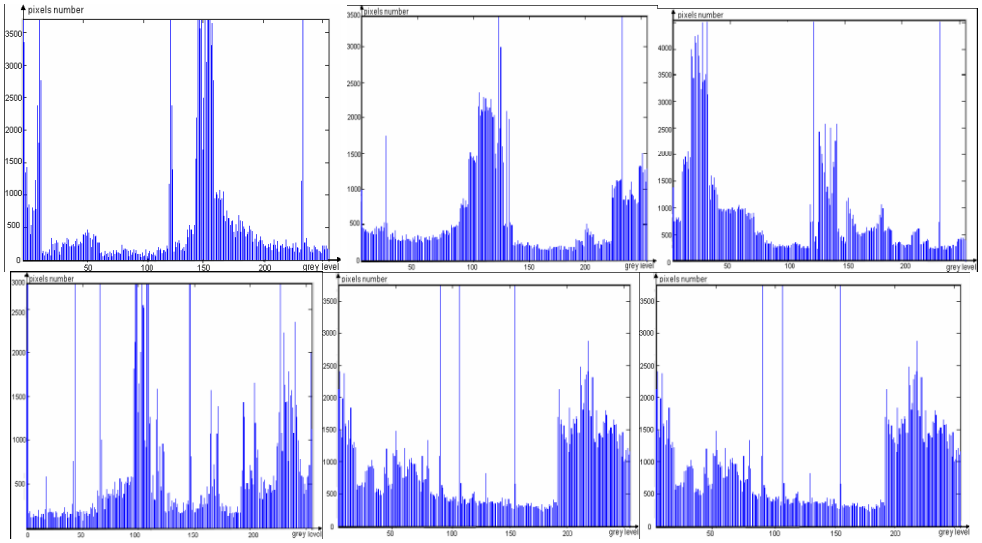


Figure 6.3 Histogrammes des 3 images avant et après codage

Pour évaluer les performances de notre algorithme nous avons mesuré et comparé les consommations en temps de l'algorithme séquentiel et de l'algorithme imbriqué, par utilisation de la procédure tic-toc de Matlab. Trois mesures consécutives sont réalisées sur chacune des images. Les résultats de ces mesures sont reportés dans le tableau 6.**3.** Comme attendu, les performances de l'algorithme combiné sont au dessus de celles de l'algorithme séquentiel. Nous avons également étudié les variations du temps en fonction de la taille des images à travers un certain nombre d'images et notre conclusion est que, comme on pouvait s'y attendre, le processus de codage/décodage dépend aussi de la taille des images. La Figure 6.4 montre cette dépendance pour le codage comme pour le décodage.

Tableau 6.3 Consommation en temps pour les deux algorithmes (en secondes)

	Processus séquentiel		Processus combiné	
	Codage	Décodage	Codage	Décodage
Image (a)	0.7410	0.8110	0.5310	0.3900
	0.7410	0.8010	0.5110	0.3910
	0.7420	0.8010	0.5310	0.3800
Image (b)	3.0570	3.0540	1.4220	1.3020
	3.0240	3.0240	1.4630	1.2920
	3.040	3.0250	1.4420	1.3120
Image (c)	10.9860	8.4220	4.4890	3.3760
	10.9950	8.4130	4.4980	3.3760
	10.9950	8.4150	4.5080	3.3660

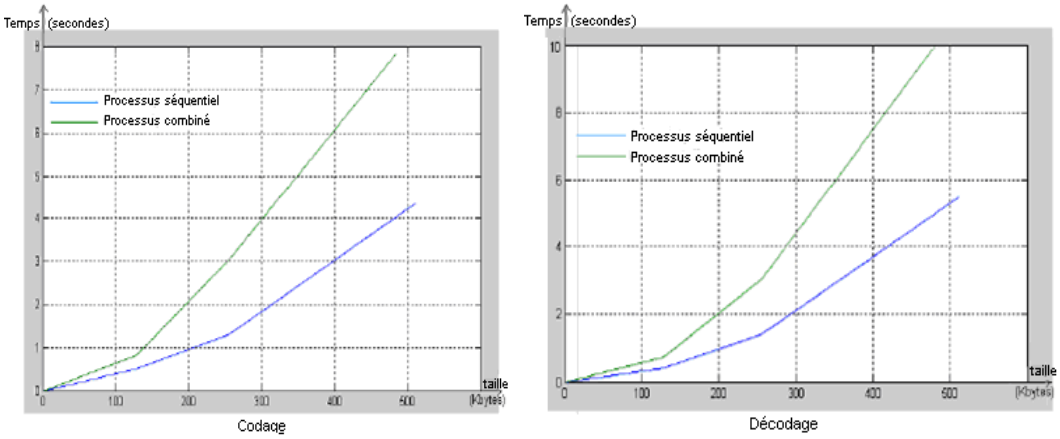


Figure 6.4 Variations du temps en fonction de la taille des images

En dehors de la vitesse d'exécution, nous avons évalué la qualité des images produites par les deux processus. On peut déjà remarquer sur la Figure 6.2 que l'effet de bloc est réduit par application du processus combiné. Les Figures 6.3 montrent également les histogrammes de ces mêmes 3 images et on peut remarquer une meilleure distribution des niveaux de gris sur les images produites par le processus combiné, ce qui démontre un meilleur brouillage. De même nous avons évalué l'entropie sur chacune des images, qui s'est avérée supérieure dans le cas du processus combiné.

Tableau 6.4 Mesures de l'entropie (en bits)

Original image	Processus séquentiel		Processus combiné
Image (a)	5.8247	6.7033	7.4230
Image (b)	7.0814	7.2644	7.5911
Image (c)	6.4857	7.2595	7.5810

III- Un système d'authentification à clé secrète basé compression [BB05]

La deuxième solution proposée se manifeste dans une application entre deux tiers communicants, leur permettant de s'émettre mutuellement des dossiers médicaux à travers un réseau, tout en assurant l'intégrité de l'image échangée ainsi que la confidentialité des informations l'accompagnant. Ces fonctionnalités sont obtenues à travers le mariage d'un mécanisme de code d'authentification de message (MAC) à clé secrète et d'un algorithme de marquage de type LSB rendu réversible grâce à une méthode de compression sans perte.

Dans la solution précédente, l'image était reçue chiffrée et le récepteur se devait de la déchiffrer d'abord pour pouvoir l'exploiter, ainsi que les données patient. Par contre, ici l'image est reçue simplement marquée, et le récepteur peut choisir de l'exploiter telle quelle, ou extraire les données patient et effectuer une vérification d'intégrité, ou encore la sauvegarder au format marqué, pour une utilisation ultérieure tout en la gardant protégée.

Les objectifs attendus de cette procédure sont de pouvoir :

- Insérer la marque de manière invisible dans l'image.
- Authentifier l'image sans ambiguïté et en mode aveugle.
- Obtenir une image marquée qui ne soit pas supérieure en taille à l'image originale.
- Récupérer l'image originale dans son intégralité à partir de l'image marquée.

Pour minimiser la visibilité, nous utiliserons un seul LSB par pixel au niveau du marquage. Le MAC servant à authentifier l'image sera obtenu par chiffrement de l'empreinte MD5 avec l'algorithme de Vigénère (voir chapitre 2). Il sera inséré dans les LSB après les avoir compressé par RLE, pour « lui faire de la place ».

Du côté émetteur, les étapes suivantes sont effectuées :

- Calculer l'empreinte de l'image sur les 7 bits de poids forts (MSB) de chacun des pixels.
- Concaténer l'empreinte de l'image avec les données du patient et chiffrer le résultat avec l'algorithme de Vigénère
- Sélectionner les LSBs de tous les pixels de l'image originale et leur appliquer une compression sans perte RLE.
- Concaténer le résultat de la compression avec les données chiffrées puis les réinsérer dans l'emplacement des LSBs.

La figure 6.5 montre les différentes étapes de formation et d'insertion de la marque.

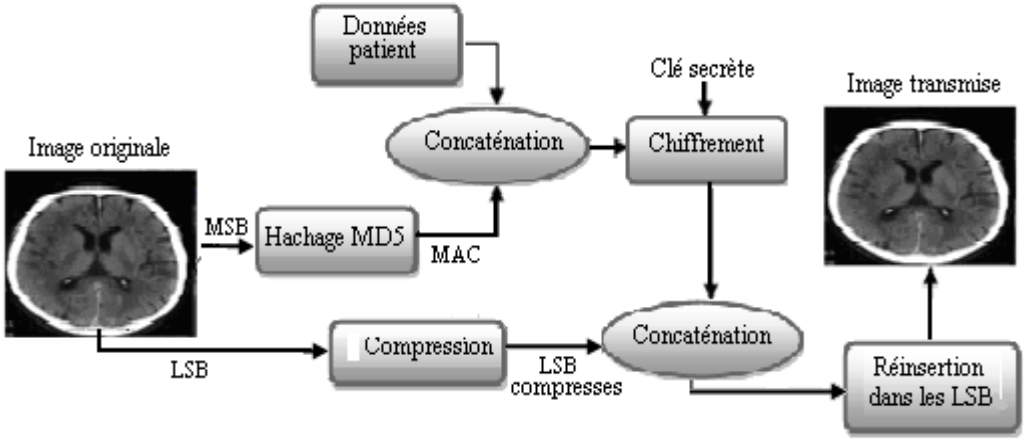


Figure 6.5 Schéma de formation et d'insertion de la marque

Pour extraire les données patient et effectuer un contrôle d'intégrité sur l'image reçue, le récepteur effectue de son côté les étapes suivantes :

- Extraire les LSBs de l'image reçue.
- Séparer ces LSBs en deux chaînes de caractères l'une représentant les LSBs originaux compressés et l'autre la concaténation de l'empreinte avec les données patient.
- Décompresser les LSBs et les remettre à leur emplacement pour récupérer l'image originale.
- Récupérer les données du patient et le MAC de l'image en appliquant un décodage de Vigenère.
- Le contrôle de l'intégrité de l'image se fait en recalculant le MAC de l'image et en le comparant avec le MAC extraite de l'image.

La figure 6.6 montre les différentes étapes d'extraction et de vérification de la marque.

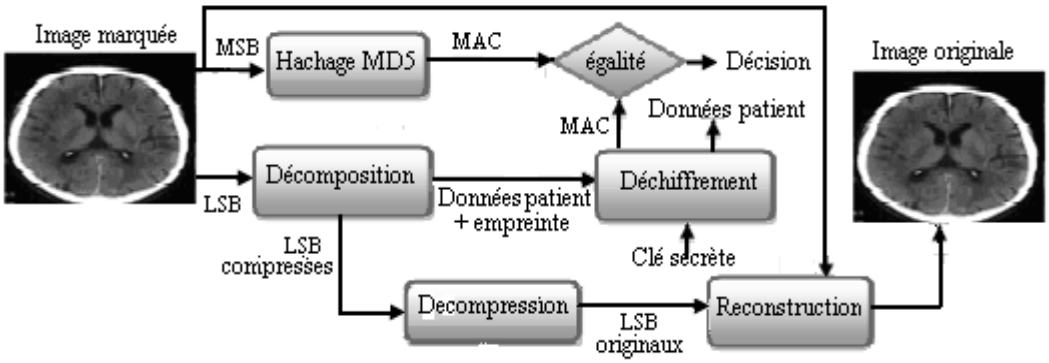


Figure 6.6 Schéma d'extraction et de vérification de la marque

Ce procédé de marquage se caractérise par une grande capacité d'insertion. Par exemple pour une petite image 256x256, elle permettra d'insérer jusqu'à 8 Ko de données patient incluant les 128 bits du MAC, ce qui est largement au dessus des besoins de la plupart des applications. De plus, cette capacité va encore augmenter avec la taille de l'image. On peut alors choisir de ne marquer que certains pixels spécifiques choisis de manière secrète, ce qui va renforcer la sécurité de la méthode. Bien-sûr, le pas d'insertion va dépendre aussi de la taille de l'image. Le tableau 6.4 montre le format des données (un total de 96 caractères) qui seront insérées dans l'image en plus des 32 caractères de l'empreinte. La figure 6.7 montre l'effet du marquage sur 3 images échographiques, de différentes tailles, en termes de PSNR. Les figures 6.8 à 6.11 montrent l'effet du marquage sur différentes images, en termes d'images différences et de leurs histogrammes.

Tableau 6.5 Format des données patient

Données	Nb de caractères	Données	Nb de caractères
Nom Patient	16	Nom-Medecin	16
Age	3	Résultats	16
Sexe	1	Diagnostic	16
Date d'acquisition	8	Traitement	20

256x256	400x268	512x512
39.72 dB	52.27 dB	41.33 dB

Figure 6.7 Impact du marquage sur trois images test en termes de PSNR

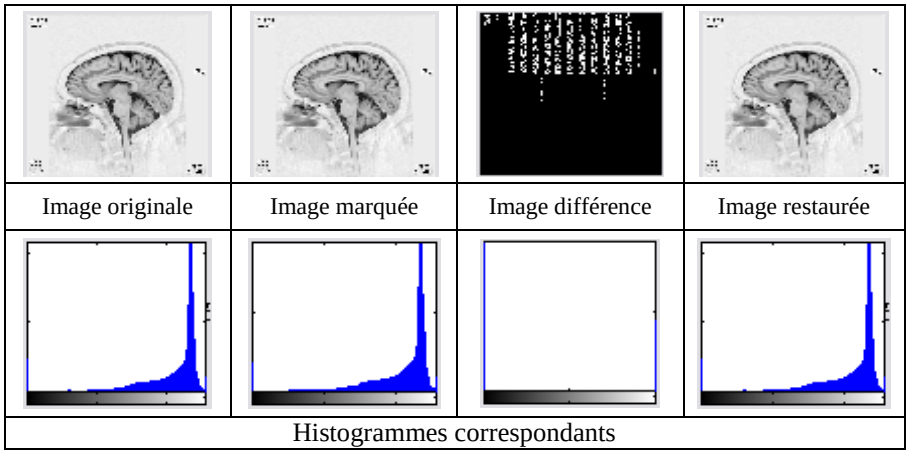


Figure 6.8 Exemple d'image IRM (512*512 pixels) avec pas d'insertion 1 pixel

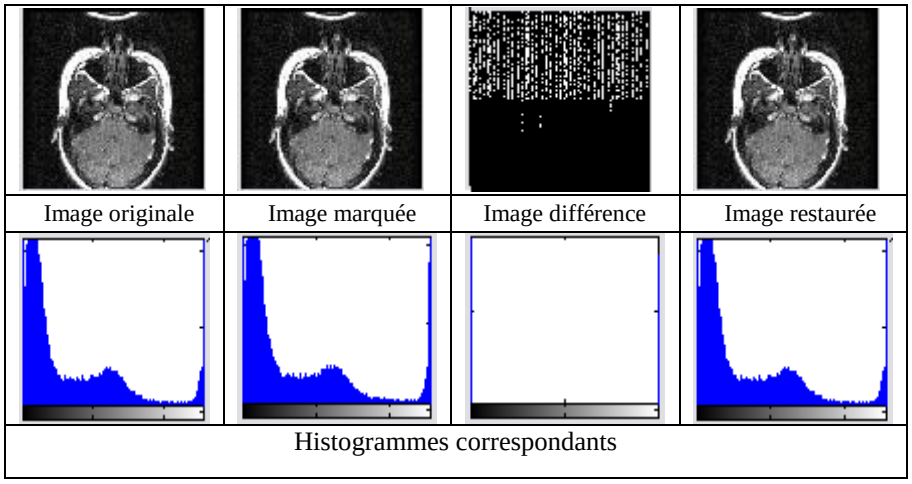


Figure 6.9 Exemple d'Image IRM (256*256 pixels) avec pas d'insertion 2 pixels

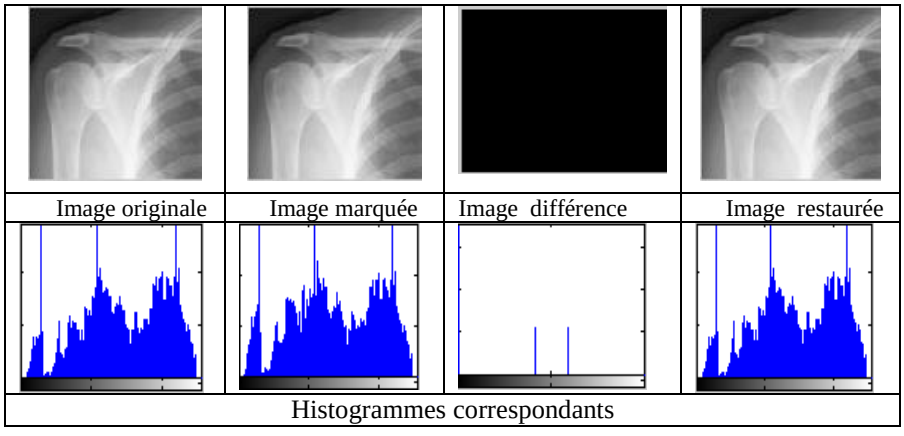


Figure 6.10 : Exemple d'image Rayon X (121*104 pixels) avec pas d'insertion 1 pixel

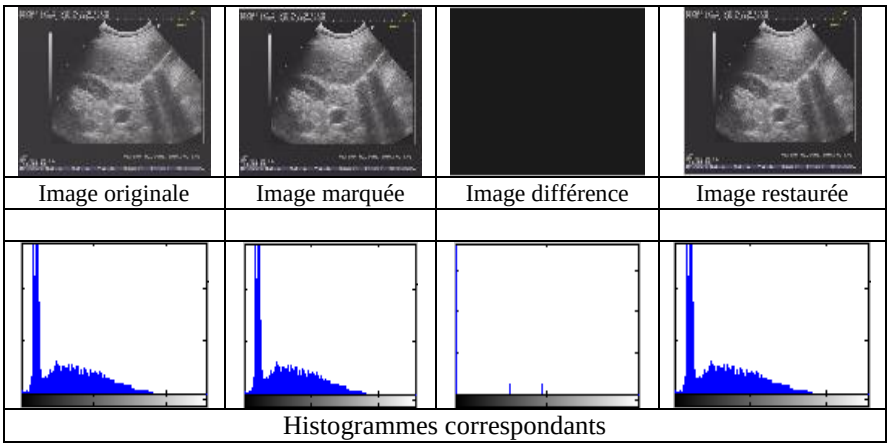


Figure 6.11 : Exemple d'image d'échographie (800*600 pixels) avec pas d'insertion 25 pixels

Plusieurs tests concernant l'intégrité ont été effectués. La détection des données patient et de l'empreinte se fait avec succès pour les images marquées n'ayant subi aucune attaque, mais échoue pour les images ayant été attaquées.

La Figure 6.12, montre une image (a) et l'image (a') marquée correspondante ainsi que l'image (b) originale, l'image (b') marquée puis l'image (b') marquée et attaquée par une retouche d'image simulant une fracture. Le tableau 6.6 montre le résultat du contrôle d'intégrité de ces images.

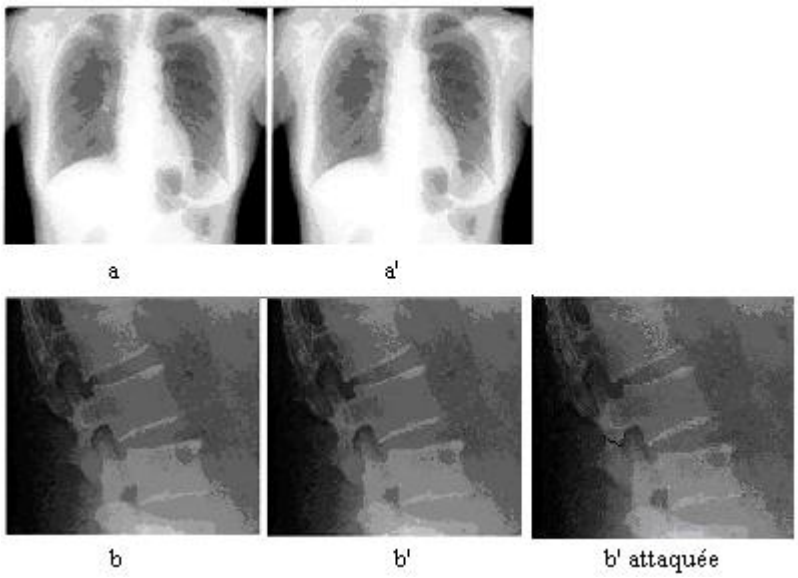


Figure 6.12 Echantillon d'images a : originale et marquée
b : originale, marquée puis attaquée

Tableau 6.6 Vérification de l'intégrité en comparant l'empreinte calculée avec l'empreinte extraite

Valeur de l'empreinte	Image a : 8C12ED177E82F3D99CD4B66BEE1B6DA7
Valeur de l'empreinte extraite des LSB	Image a' : 8C12ED177E82F3D99CD4B66BEE1B6DA7
Re-calcul de l'empreinte	Image a' : 8C12ED177E82F3D99CD4B66BEE1B6DA7

Valeur de l'empreinte	Image b : F3826EB03048F9BCD66F74DCE3E08B85
Valeur de l'empreinte extraite des LSB	Image b': F3826EB03048F9BCD66F74DCE3E08B85 attaquée
Re-calcul de l'empreinte	Image b': A1FB76CCEBE8C7835815C106B3285C6 attaquée

IV- MEDIMAGE : Un service web pour le partage sécurisé d'images médicales [BBZ05][BB06]

Dans cette section nous allons proposer un cadre applicatif possible pour nos contributions qui va se manifester sous la forme d'un service web permettant à une communauté déterminée d'utilisateurs (par exemple des radiologistes) de partager en toute sécurité une collection d'images médicales à travers le Net. En effet, les réseaux informatiques étant encore souvent absents de nos structures hospitalières, nous avons opté pour l'échange d'images à travers l'Internet qui, lui, est largement disponible en Algérie. Ce service va permettre aux médecins inscrits de constituer, à terme, une banque d'images médicales des cas les plus intéressants qu'ils rencontrent dans leur vie professionnelle, pour instaurer une recherche collaborative en toute sécurité, et sans faillir aux exigences éthiques et légales propres au domaine. La base construite pourra également être utilisée par des étudiants qui auront le seul privilège de consulter ou de télécharger une version marquée des images. Pour permettre une utilisation fiable de ce service, nous mettrons en œuvre les mécanismes d'authentification et de confidentialité présentés précédemment, à travers une architecture client-serveur dans *laquelle* les droits d'accès sont gérés par le serveur.

L'utilisation pratique de ce système présente de nombreux avantages :

- Les patients n'aiment généralement pas voir leurs renseignements privés ouverts au public sur un site web, surtout s'il s'agit d'une maladie grave. De ce point de vue, on peut dire que les images médicales constituent une propriété privée dont le copyright est détenu par les patients concernés.

- Quand un médecin archive une image pour une longue période et un autre médecin désire s'y référer (par exemple pour une étude comparative avec des

cas similaires), ce dernier peut vouloir s'assurer de son intégrité avant de l'utiliser.

- Pour des médecins en mobilité, par exemple lors d'une conférence, l'Internet est souvent le seul canal de communication accessible. Ce service pourra s'avérer très utile dans diverses interventions.

- Le fait d'insérer las données patient au sein de l'image correspondante empêche la perte de lien qui peut arriver dans une sauvegarde séparée.

- Grâce à un marquage réversible, l'image peut être authentifiée de manière non ambiguë et peut être restaurée dans son intégralité après extraction de la marque, pour un diagnostic fiable. Si elle est altérée pour quelque raison que ce soit, l'utilisateur en est informé.

- Le marquage/vérification ainsi que toutes les opérations de sécurisation sont effectués du côté serveur, ce qui décharge les médecins d'avoir à s'occuper de ce côté ou d'avoir à installer sur leur machines des plugs-in spécifiques qu'ils ne sauront pas spécialement faire fonctionner. De plus, le serveur pourra être considéré comme un TTP (Trusted Third Party) dans un scénario de gestion des conflits, dans un contexte médico-légal.

IV-1 Fonctionnalités du système

Différentes fonctionnalités sont offertes par le service au client lui permettant de:

- S'enregistrer pour devenir un utilisateur autorisé;
- S'enregistrer pour devenir un utilisateur privilégié;
- Consulter la collection d'images par catégories;
- Ajouter une image dans la base après qu'elle soit marquée;
- Télécharger une image marquée;
- Vérifier l'intégrité d'une image téléchargée;
- Extraire les informations patient pour les utilisateurs privilégiés;

Toute image publiée sera marquée par les informations patient correspondantes envoyées par le propriétaire, en plus d'une empreinte spécifique calculée par le serveur et destinée à son authentification en temps voulu.
Pour éviter le coût de l'établissement d'une infrastructure à clé publique (PKI), le rôle de tiers de confiance est délégué au serveur dans lequel est implémentée une autorité de certification (AC) gérant les clés privées et publiques et distribuant les certificats aux utilisateurs privilégiés. Pour la génération des clés de sessions le protocole de Diffie-Hellman (voir chapitre 2) est utilisé. Le dialogue entre les machines clientes et le serveur se fait par le biais de formulaires et de rapports HTML.

La structure des certificats utilisée est une version simplifiée du standard X.509 [Sta03] contenant les informations suivantes :

- Le numéro de série du certificat;
- Le nom (DN, pour *Distinguished Name*) de l'autorité de certification;
- La date de début de validité du certificat;
- La date de fin de validité du certificat;
- Les clés publique et privée du propriétaire du certificat;
- La signature de l'autorité de certification;

L'ensemble des informations clients (informations + clés du demandeur) est signé par l'autorité de certification et constitue sa signature. Cela signifie qu'une fonction de hachage est utilisée pour produire une empreinte de ces informations, puis ce condensé est chiffré à l'aide de la clé privée de l'autorité de certification;

Au premier démarrage du serveur, l'autorité de certification (AC) génère les clés privée et publique du serveur.

Avant de pouvoir utiliser le service, tout utilisateur doit enregistrer ses informations personnelles en choisissant un nom d'utilisateur et un mot de passe. Quand l'enregistrement d'un utilisateur privilégié réussit, l'AC génère les clés publique et privée de cet utilisateur. Une entrée dans la base sera créée pour cet utilisateur contenant ses informations personnelles ainsi que sa clé publique. Une clé de session est conjointement crée par les deux parties à travers une procédure type Diffie-Hellman. Finalement, l'utilisateur privilégié obtient un certificat contenant sa paire de clés et la clé publique du serveur le tout chiffré à l'aide de IDEA (voir chapitre 2) et de la clé de session fraîchement générée. Le certificat sera signé par l'autorité de certification à l'aide de sa clé privée et de l'algorithme RSA.

Pour accéder au service, tout utilisateur doit d'abord s'identifier à travers son nom utilisateur et son mot de passe. Si le processus d'identification réussit, l'échange peut alors commencer.

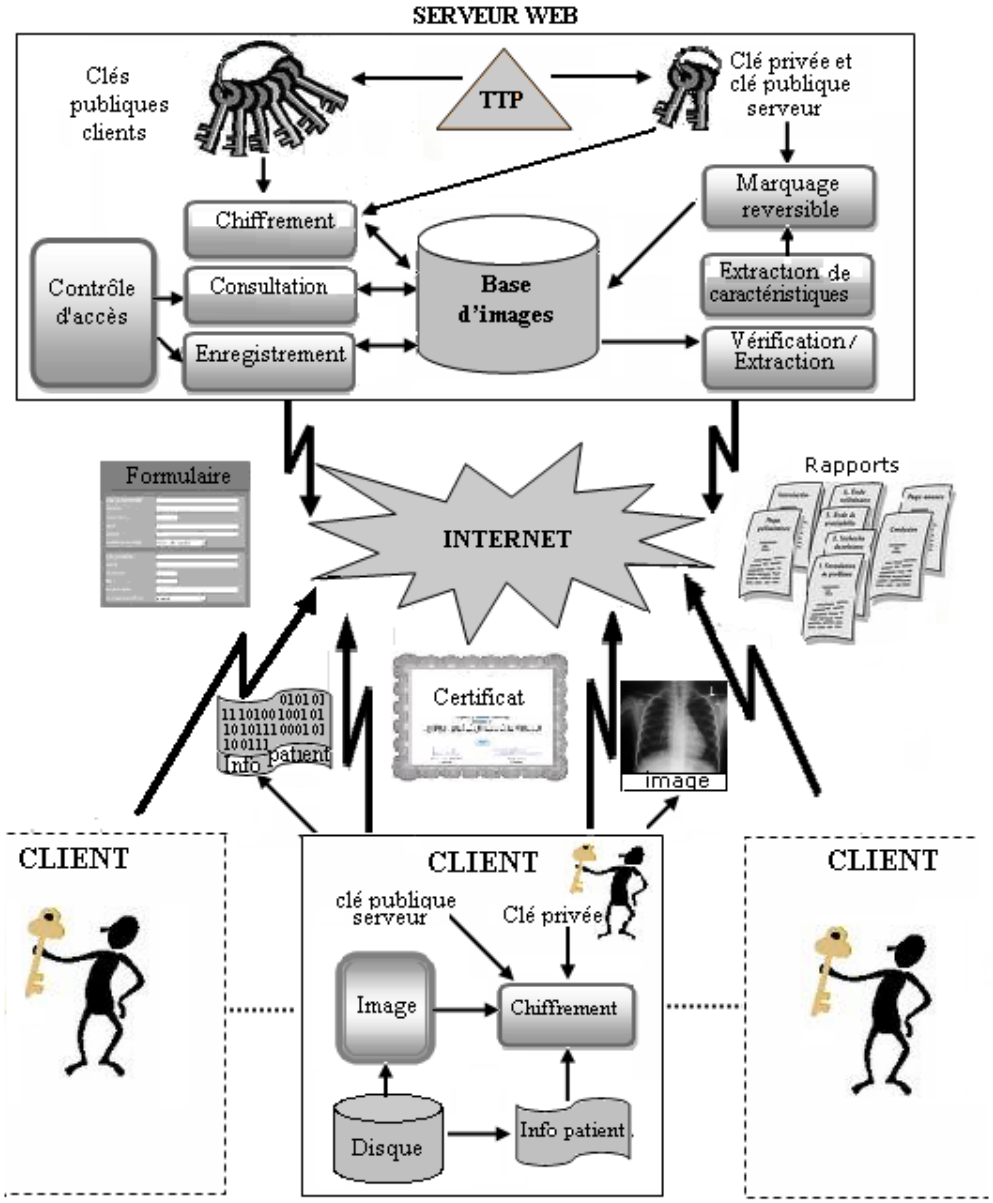


Figure 6.13 Architecture du système

IV- 2 Règles d'utilisation :

- Les images sont stockées au niveau du serveur uniquement sous forme marquée, ce qui va limiter les possibilités de falsification. Elles seront reçues sous forme chiffrée, et dés qu'elles seront déchiffrées elles seront directement marquées et rangées dans la base.

- Seuls les utilisateurs privilégiés peuvent ajouter des images dans la base; En effet, les utilisateurs autorisés mais non privilégiés forment un plus grand groupe non nécessairement dignes de confiance, qui peuvent être, par exemple, des étudiants.

- L'inscription d'un nouvel utilisateur privilégié nécessite l'aval de deux utilisateurs privilégiés déjà inscrits. Pour ceci, ce nouvel utilisateur doit présenter dans sa requête au serveur l'identité de ses deux parrains ainsi que le numéro de série de leurs certificats respectifs avec la signature de l'AC. La signature numérique permet d'assurer que le certificat de ce parrain a bien été établi par l'AC et lui donne un caractère officiel.

- C'est à l'utilisateur qui ajoute l'image dans la base qu'incombe le choix de la catégorie de sauvegarde dans la base; Le serveur n'est pas censé pouvoir discerner les catégories d'images qu'il reçoit et qu'il doit seulement marquer et ranger dans la base.

- Le serveur n'a pas accès aux données patient, puisqu'elles sont reçues sous une forme chiffrée; Elles resteront toujours uniquement accessibles au propriétaire initial de l'image car chiffrées avec sa clé secrète.

- Les utilisateurs non privilégiés peuvent uniquement voir ou télécharger une version marquée de l'image;

- Tous les utilisateurs enregistrés peuvent vérifier l'intégrité d'une image avant de la télécharger;

- Les utilisateurs privilégiés peuvent demander à supprimer la marque et obtenir l'image originale authentifiée; En effet, certaines utilisations de l'image téléchargée nécessitent une version très précise de l'image pour qu'elle soit exploitable.

- Seul l'utilisateur ayant fourni l'image pourra déchiffrer puis exploiter les données patient grâce au mécanisme de clé privée/clé publique instauré lors du chiffrement (étant le seul à connaître la clé de chiffrement.)

IV- 3 Méthode de conception

Pour la conception de notre application, nous avons choisi une méthode simple et générique qui se situe à mi-chemin entre UP (Unifed Process), un cadre général très complet de processus de développement, et XP (eXtreme Programming), une approche centrée sur le code. Elle repose essentiellement sur celle présentée par Pascal Rocques dans [Roc03]. Elle comporte 3 étapes à savoir :

- L'étape de l'identification des besoins qui consiste à :
  - Identifier et représenter les besoins à l'aide des diagrammes de cas d'utilisation.
  - Spécifier les besoins d'une manière détaillée en utilisant le diagramme de séquence système.
  - Concevoir une maquette de l'IHM de l'application.

- L'étape d'analyse, et qui contient :
  - Analyse du domaine en utilisant le modèle du domaine (classe métiers).
  - Diagramme de classes participantes.
  - Diagramme d'activités de navigation.

- L'étape de conception, qui consiste à établir :
  - Le diagramme d'interaction.
  - Le diagramme de conception.

A titre d'illustration, nous présentons dans ce qui suit le diagramme de cas d'utilisation et quelques diagrammes de classes et diagrammes de séquence les plus représentatifs.

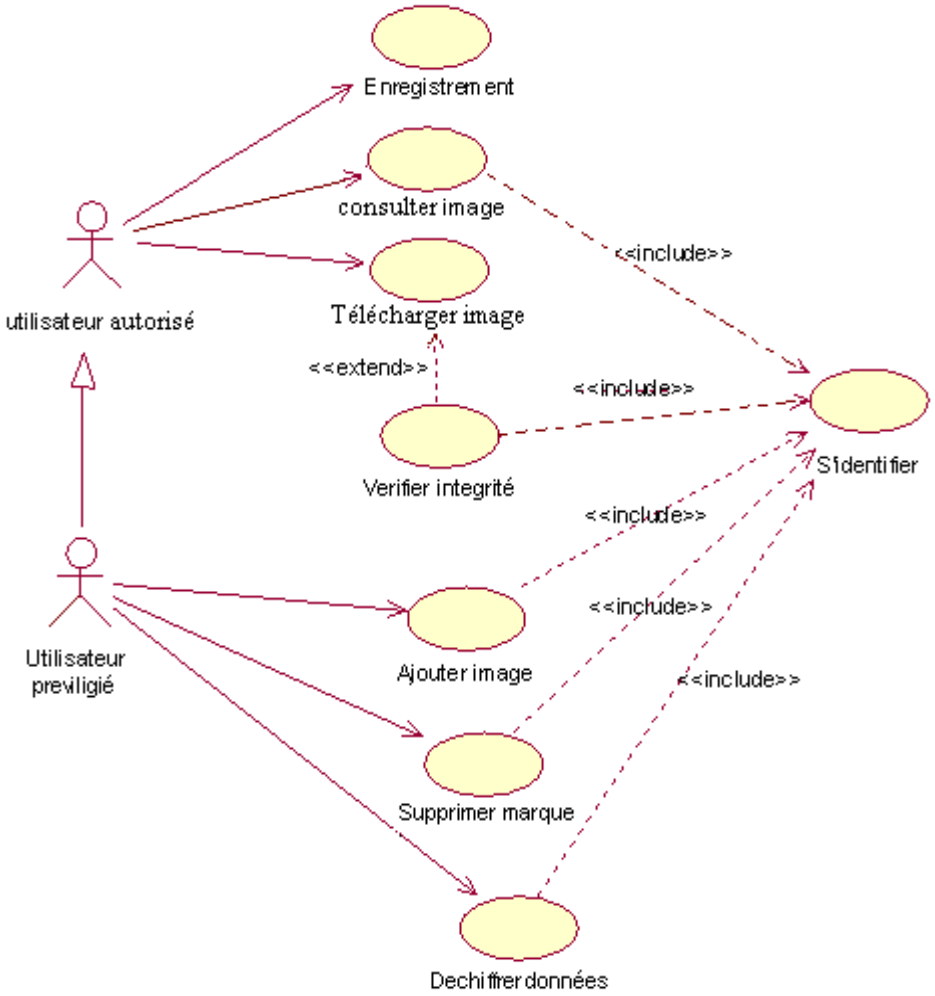


Figure 6.14 Diagramme de cas d'utilisation

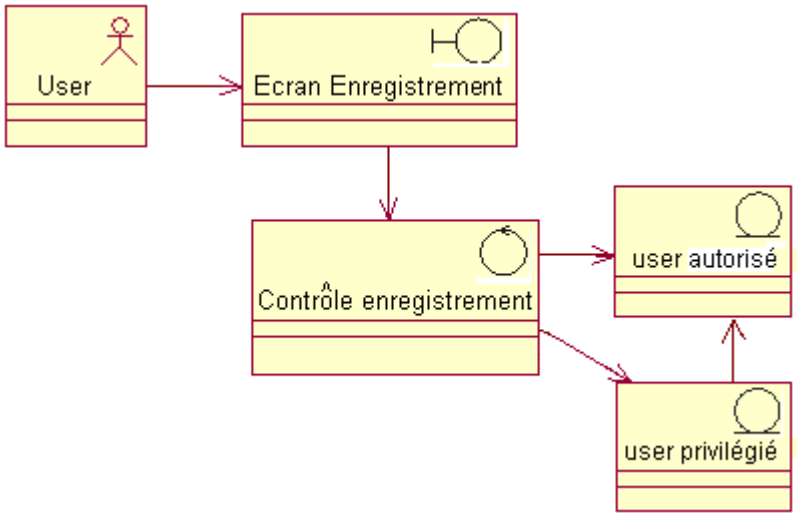


Figure 6.15 Diagramme des classes participantes de "Enregistrement"

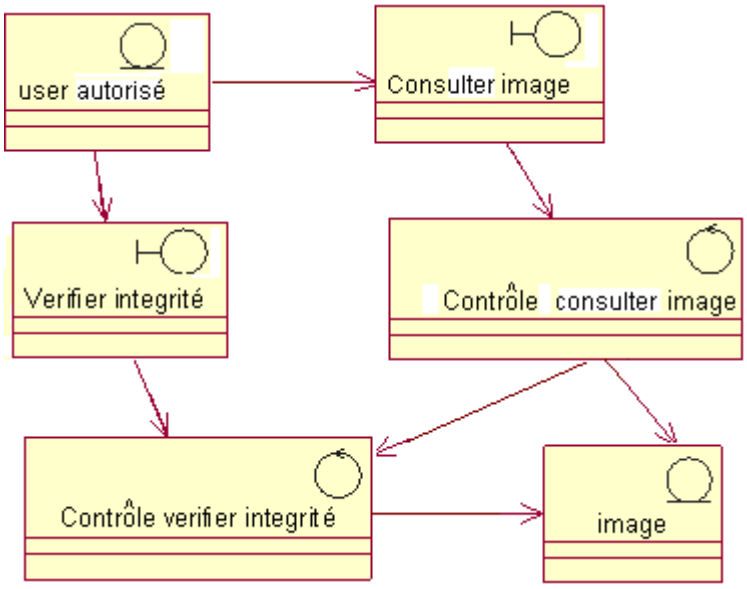


Figure 6.16 Diagramme des classes participantes de "Consulter Image"

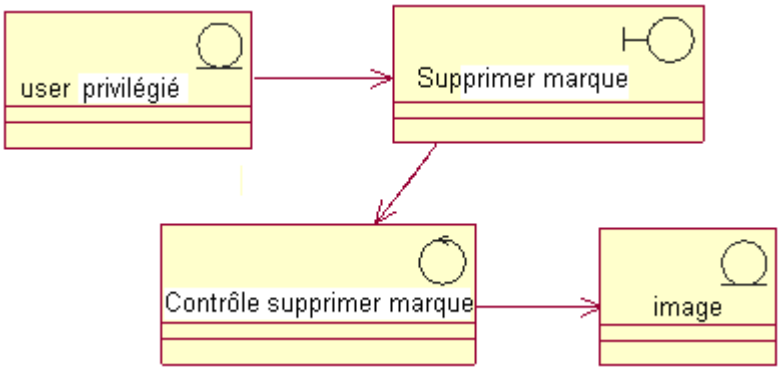


Figure 6.17 Diagramme de classes participantes de " Supprimer marque"

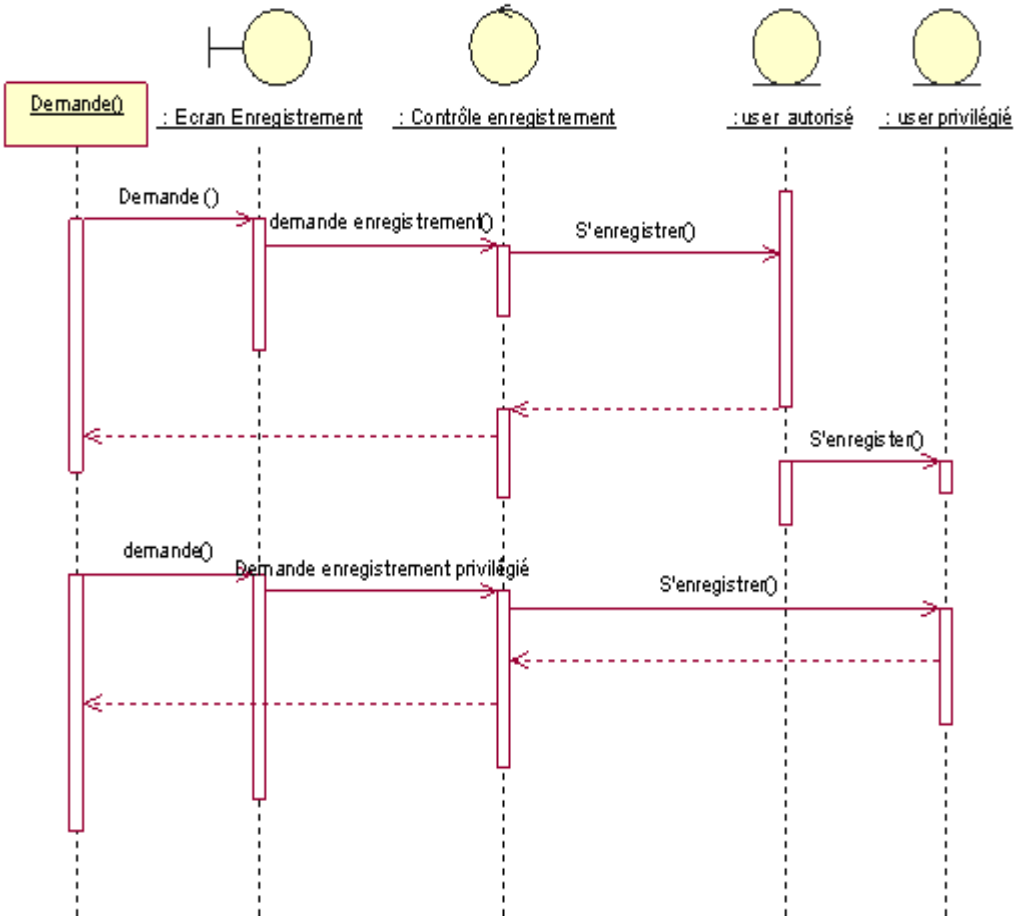


Figure 6.18 Diagramme de séquence de " Enregistrement"

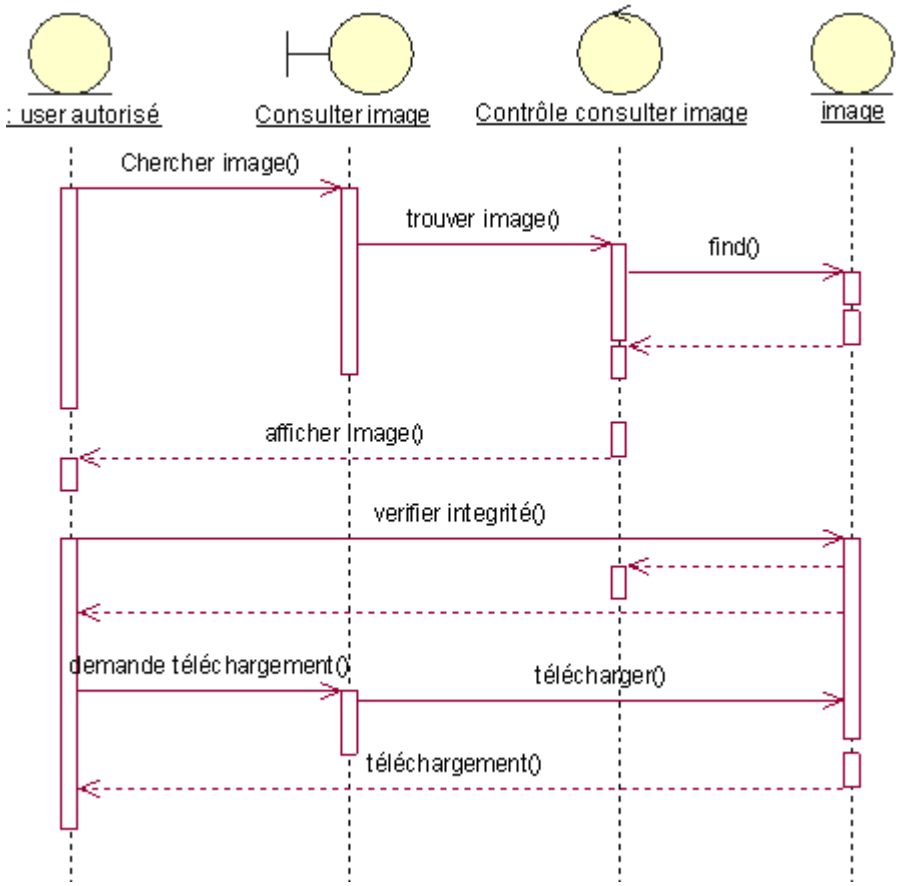


Figure 6.19 Diagramme de séquence de "Consulter image"

IV.4 Réalisation du service

L'architecture du système est basée sur un serveur web donnant accès à une base d'images à travers un site web publiquement accessible. L'application côté serveur utilise le tandem PhpMySql sous le serveur web Apache, qui est bien adapté au traitement des formulaires. MySql est un petit SGBD compact, idéal pour des applications de petite envergure comme la nôtre. De plus, Apache fournit la possibilité de la configuration d'accès aux fichiers qui permettront au serveur d'associer des noms d'utilisateurs et des mots de passe en vue de gérer l'accès aux différentes pages du site.

Le côté client utilise, quand a lui, un simple navigateur avec l'URL appropriée.

Pour chaque image à inclure dans la base, une clé unique est générée (Image-Key), puis chiffrée à l'aide de la clé privée du serveur. Dés que l'image ainsi que les données patient chiffrées sont obtenues, le serveur invoque la procédure de marquage réversible telle que décrite dans le chapitre précédent. Il renvoie ensuite au propriétaire Image-Id et sauvegarde dans la table des images Image-Id, Image-Key et User-Id en même temps que l'image.
Quand la procédure de vérification est invoquée, le serveur contrôle d'abord le User-Id et le mot de passe, et si le client s'avère être un utilisateur autorisé, il récolte l'image dénotée par Image-Id, et utilise la clé correspondante pour effectuer la vérification d'intégrité.

Il génère ensuite un rapport de vérification en HTML renseignant si l'image est authentique ou si elle est altérée, auquel cas il renvoie au client un message de mise en garde.

Pour finir, nous présentons quelques écrans de saisie de l'application.

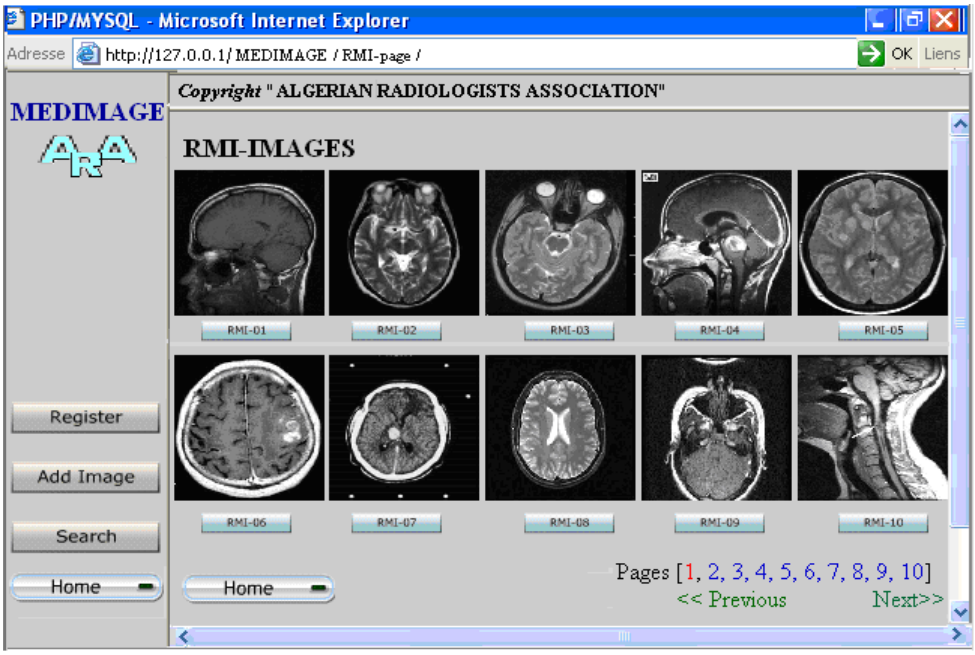


Figure 6.20 Consultation de la base

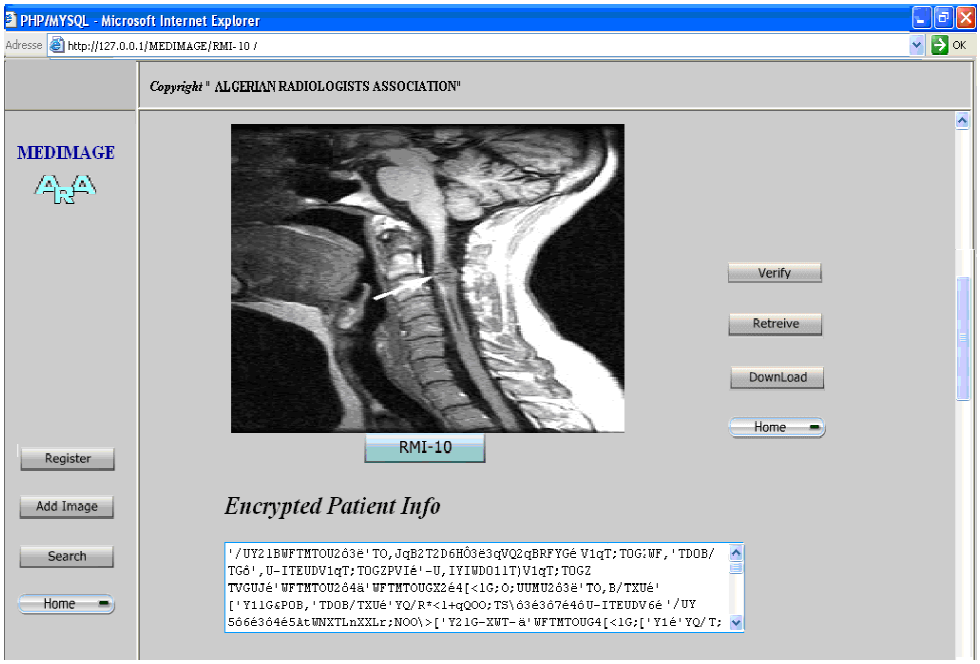


Figure 6.21 Vérification d'une image de la base

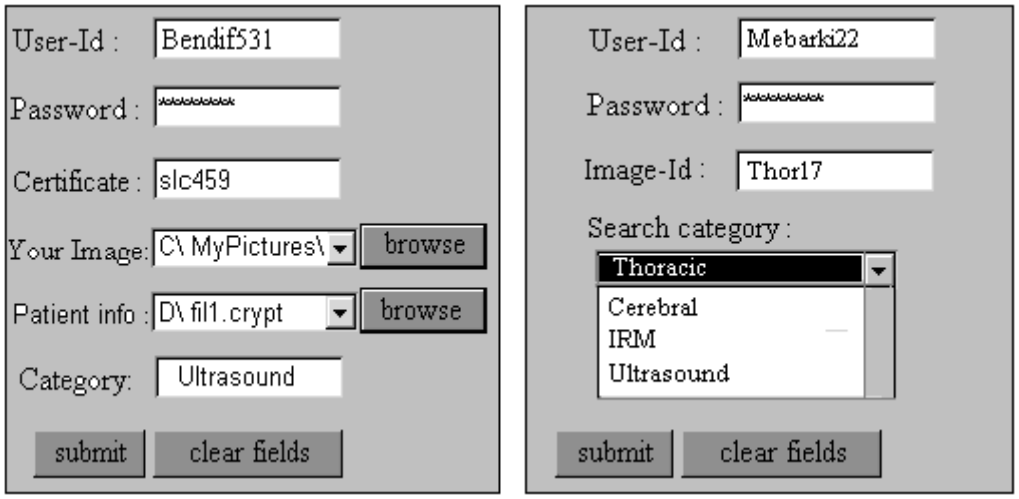


Figure 6.22 Exemples de formulaires

Conclusion

Dans ce chapitre, plusieurs solutions ont été proposées pour l'application du marquage numérique dans le domaine de l'imagerie médicale numérisée, dans un contexte distribué. Ces propositions tentent de prendre en compte les spécificités de l'image médicale et adoptent différentes stratégies d'insertion en faisant appel à divers outils tels que, la compression sans perte, la cryptographie et des procédés de marquage de nature réversible.

L'hybridation de ces méthodes a conduit à couvrir des aspects clés de la sécurisation des applications de télémédecine tels que la vérification de l'origine des images échangées, leur intégrité, ainsi que la confidentialité des données patient l'accompagnant, tout en ménageant la qualité visuelle des images transmises.

Enfin, les solutions de sécurité proposées reposent uniquement sur des outils logiciels.

Conclusion générale

Au terme de cette thèse nous tentons de tirer des conclusions des travaux présentés et de discuter les perspectives d'amélioration de nos propositions.

L'objectif principal de cette étude était de proposer une méthode d'authentification souple basée sur des caractéristiques d'ordre sémantique de l'image et sur des concepts cryptographiques. La caractéristique majeure de cette application est qu'elle est s'adapte au contenu de l'image.

Afin de mener cette étude, nous avons investigué le domaine de l'analyse d'images et nous avons finalement retenu la méthode d'analyse par texture afin de caractériser une image par des paramètres pertinents. Ces paramètres serviront à la construction d'une signature basée sur le contenu, destinée à être insérée dans l'image hôte par des techniques de marquage numérique.

En nous inspirant de la littérature scientifique sur la recherche d'images par le contenu, sur la classification et sur la segmentation, nous avons finalement retenu les approches par les statistiques de premier et de second ordre portant sur le calcul de la matrice de cooccurrence et des paramètres d'Haralick. Cette dernière méthode, bien qu'elle ait été introduite il y a assez longtemps, apparaît encore incontournable et fait souvent référence dans les domaines sus cités.
L'approche statistique modélise les notions qualitatives usuelles de texture, à savoir, granularité, contraste, homogénéité, répétitivité, fragmentation, orientation, etc. Elle est utilisée pour caractériser des structures fines, sans régularité apparente, c'est ce qui nous a d'abord incité à les utiliser étant donné qu'on se situe dans un contexte d'imagerie médicale, généralement caractérisée par des tissus mous ayant des structures tout à fait aléatoires et le plus souvent non homogènes.

Après une étude bibliographique des techniques d'authentification par marquage existantes et aussi de la théorie de l'analyse d'image par texture, nous avons mis au point une méthode semi-fragile basée sur le contenu textural de l'image. De nombreux tests sur des échantillons d'images ont permis d'évaluer les performances de cette technique. Ils ont mis en évidence quelques un de ses avantages telle que sa capacité de localisation des attaques et sa robustesse face

aux transformations géométriques et ont souligné les limites de ce type d'approche en termes de fausses alarmes et faux rejets.

Dans ce document, nous avons proposé différents états de l'art qui dépassent parfois le cadre applicatif nous ayant servi à élaborer nos algorithmes. Cependant, pour des raisons de perspectives et pour une meilleure compréhension du contexte général de cette thèse, il nous est apparu nécessaire de les présenter dans leur globalité. Afin d'élaborer un algorithme de protection globale pour les images médicales, nous avons privilégié des méthodes de faible complexité opérant dans le domaine spatial. En effet, cela nous a permis d'envisager de combiner une solution de chiffrement, une solution de compression et une solution de marquage utilisant le même support. En marge du marquage, nous avons donc étudié les principes de cryptographie afin de proposer une solution complète de protection des images médicales, alliant une protection a priori et une protection a posteriori.

Enfin, nous pensons que cette étude a juste permis de poser les bases d'une méthode d'authentification innovante, mais des améliorations et des évolutions sont envisageables sur différents plans. Plus précisément nous pensons aux perspectives suivantes :

- L'inconvénient majeur de l'évaluation des attributs texturaux par des méthodes statistiques de second ordre est la manipulation des matrices de cooccurence coûteuse en temps. Bien que nous ayons essayé de minimiser au mieux ces temps par des opérations de prétraitement, il serait intéressant de réfléchir à des moyens plus rapides de les utiliser. Par exemple, une idée possible est de ne pas utiliser chaque fois toute la matrice de texture, coûteuse dans la manipulation, mais juste certaines grandeurs caractéristiques sur la matrice de co-occurrence. On pourrait utiliser des transformations linéaires sur les matrices en lieu et place des 14 paramètres de haralick, évitant du même coup de statuer sur le choix de l'un ou l'autre de ces paramètres. Les théorèmes de l'algèbre linéaire nous disent que deux matrices semblables ont le même polynôme caractéristique, les mêmes valeurs propres, la même trace et le même déterminant. Autrement dit, ce sont des invariants de similitude. On pourrait alors, représenter la signature texturale du bloc courant, par exemple par le polynôme caractéristique et comparer avec le nouveau polynôme lors de la phase de vérification. Le même emploi peut être fait avec les valeurs propres, la trace ou le déterminant, ou encore une combinaison de ces grandeurs. Bien sûr il faudrait s'assurer au préalable de la pertinence de ces grandeurs par diverses simulations. A creuser…

    - Toujours dans le même ordre d'idée, au lieu d'utiliser la représentation directe des matrices de cooccurence on pourrait utiliser la représentation chainée avancée par Clausi & Jernigan [Cla01]. Dans cette méthode, chaque nœud d'une liste GLCLL (Grey Level Cooccurence Linked List) contient les paires de pixels (i,j) et leur probabilités jointes p(i,j). Seules les probabilités non nulles sont maintenues dans la liste, qui doit être

gardée toujours triée. Si la paire courante est déjà représentée dans la liste sa probabilité est mise à jour, sinon, un nœud est crée en position adéquate et initialisé. Cette structuration permet l'accès direct aux différents noeuds par des pointeurs et conduit à un calcul rapide des différentes statistiques, en évitant les doubles sommations à travers toute la matrice de cooccurence, qui sont remplacées par des sommations simples sur la longueur de la liste chaînée. Selon ses auteurs cette méthode ne consomme que 18% du temps habituellement pris par les MCNG. Le seul inconvénient est le traitement supplémentaire découlant de devoir maintenir la liste triée. Bien que peu utilisée à ce jour, cette méthode mérite d'être testée dans le cadre de notre étude.

- La caractérisation de texture à l'aide de paramètres pertinents permet également la restauration d'une partie dégradée ou manquante dans une image en la remplaçant par une version synthétique générée a partir du modèle textural élaboré. Dans le domaine de l'infographie ou de l'audiovisuel, la synthèse de texture découle naturellement de l'analyse et conduit à son utilisation pour le réalisme, l'art, le design....Il serait intéressant d'utiliser cette propriété pour tenter de reconstruire les blocs de l'image qui échouent l'authentification. Dans ce cas, la taille des blocs devient un paramètre très important et des simulations très pointues devraient être engagées auparavant afin de déterminer la taille permettant de capter l'information texturale dans son intégralité. De cette manière, on pourra ajouter une brique supplémentaire à l'édifice de protection globale à travers une fonctionnalité de reconstruction.

Nous concluons ce document sur deux thèmes. Le premier concerne la recherche de simplicité dans les solutions de sécurité proposées à côté de leur efficacité. En effet, beaucoup des techniques d'authentification étudiées à travers l'état de l'art, et bien que parfois très élaborées, ne constituent pas des solutions viables dans des systèmes réels, à cause de leur difficulté de mise en œuvre. Ceci se reflète aussi dans la recherche d'une solution globale de sécurité et les pistes que nous avons suivies pour atteindre ce but.

Le deuxième concerne le cadre applicatif proposé pour héberger nos solutions. En l'absence d'infrastructures distribuées dans nos hôpitaux, nous avons proposé l'Internet, conscients que cette solution n'est qu'une alternative temporaire, en attendant que se développent de véritables SIM à même de résoudre les nombreux problèmes posés par la gestion des dossiers médicaux en Algérie.

Références bibliographiques

[AABN01] Acharya, R., Anand, D., Bhat, S. & Niranjan, U.C. (2001). Compact storage of medical images with patient information, IEEE Transactions on Information technology in Biomedicine, vol. 5, pages 320-323.

[Ala04] Alattar, A. M. (2004). Reversible watermark using difference expansion of quads, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, vol. 3, pages 377–80.

[AN98] Anand, D. & Niranjan, U. (1998). Watermarking Medical Images with Patient Information, Proceedings of the 20th Annual International Conference of the IEEE/ Engineering in Medicine and Biology Society (EMBS) , Hong Kong, China, volume 2, pages 703-706.

[Arv03] Arvis, V. (2003). Vers une caractérisation de milieu par analyse de texture. Application à l'étude de l'ensilage de maïs. Thèse de PhD, Université Blaise Pascal, Clermont II.

[BKR02] Barreto, P., Kim, H. & Rijmen, V.(2002) . Toward a secure public-key blockwise fragile authentication watermarking. IEE Proceedings on Vision, Image and Signal Processing, volume 149, pages 57-62.

[Bas00] BAS, P. (2000). Méthode de tatouage d'image fondé sur le contenu, Thèse de Doctorat, Institut National Polytechnique de Grenoble.

[BCM02] Bas, P., Chassery, J. M. & Macq, B. (2002). Geometrically invariant watermarking using features points, IEEE Transactions on Image Processing, vol. 11, pages 1014-1028.

[BK98] Bhattacharjee, S. & Kutter, M. (1998). Compression Tolerant Image Authentication. IEEE Proceedings of the International Conference on Image Processing (ICIP'98), Chicago, USA, pages 435-439.

[BB04] Boucherkha, S. & Benmohamed, M. (2004). A lossless watermarking based authentication system for medical images, Transactions on Engineering, Computing and Technology, Vol 1, no.1, pages 240-243.

[BBZ05] Boucherkha, S., Boursas, B. & Zellit, S., (2005). A www service for sharing digital medical images between a limited set of users. Proceedings of International Symposium on Electromagnetism, Satellites and Crypography, ISESC'05, (IEEE - France section), Jijel (Algeria), pages 195-199.

[BB06] Boucherkha, S. & Benmohamed, M. (2006). A multi-tier architecture to safely share digital medical images, Proceedings of IEEE Conference on Dependability of Computer Systems DepCoS-RELCOMEX, Szklarska Poreba, Poland, 2006, pages 319 – 326.

[BLB06] Boucherkha, S., Laboudi, Z. & Benmohamed, M. (2006). A Low Cost Multipurpose Algorithm for Secure Transfer of Medical Images , International Review on Computers and Software (I.RE.CO.S.), ISSN 1828-6003, Vol. 1, issue 3, pages 217-223.

[BDB07] Boucherkha, S., Dreibine, L., & Benmohamed, M. (2007). A Statistical Model for Medical Image Authentication and Transfer", Proceedings of IEEE International Conference on e-Medical Systems e-Medisys'07, Fez, Morocco, pages 230-238.

[BB07] Boucherkha, S. & Benmohamed, M. (2007). A Texture Based Image Signature Using Second Order Statistics Characterisation", Proceedings of Information Security Workshop (IS'07) at OTM federated conferences, LNCS 4805, Springer Verlag Berlin-Heidelberg, Vilamoura (Portugal), pages 44-45.

[CFF05] Cayre, F., Fontaine, C. & Furon. T. (2005). Watermarking Security: Theory and Practice, IEEE Transactions on Signal Processing, Volume 53, Number 10, pages 3976-3987.

[CKLS97] Cox, I. J., Kilian, J. Leighton, T. & Shamoon, T. (1997). Secure Spread Spectrum Watermarking for Multimedia", IEEE Transactions on Image Processing, 6, 12, pages 1673-1687.

[Cla02] Clausi, D.A. (2002). An analysis of co-occurrence texture statistics as a function of grey level quantization. Canadian Journal of Remote Sensing, volume 28 (1), pages 45–62.

[CMS01] Coatrieux, G., Maitre, H. & Sankur, B. (2001). Strict Integrity Control of Biomedical Images, Proceedings of Security and Watermarking of Multimedia Contents III, SPIE Vol. 4314, pages 229-240.

[CSST02] Celik, A.M., Sharma, G., Saber, E. & Tekalp A. (2002). Hierarchical watermarking for secure image authentication with improved localization and security. IEEE Transactions on Image Processing, volume 11(6), pages 585-595.

[CSST05] Celik, A.M., Sharma, G., Tekalp, MG. & Saber, E.(2005).Lossless generalized-lsb data embedding, IEEE Transactions on Image Processing, vol. 14, no. 2, pages 253–266.

[CSTS02] Celik, A.M., Sharma, G., Tekalp, A. & Saber, E. (2002). Reversible data hiding, IEEE Proceeding of the International Conference on Image Processing (ICIP'02), volume II, pages 157–160.

[CZ02] Clausi, D.A. & Zhao, Y. (2002). Rapid co-occurrence texture feature extraction using a hybrid data structure. Computers & Geosciences volume 28 (6), pages 763–774.

[DR99] Dugelay J.-L., & Roche S. (1999). Introduction au marquage d'images, Annales des Télécommunications, volume 54, no 9-10, pages 427-437.

[Fri98] Fridrich, J. (1998). Image Watermarking for Tamper Detection. IEEE Proceeding of the International Conference on Image Processing (ICIP'98), Chicago, USA, pages 404-408.

[Fri99] Fridrich, J. (1999). Robust Bit Extraction From Images. Proceeding of the International Conference on Multimedia and Computing and Systems ICMCS'99, Florence, Italy, volume 2, pages 536 – 540.

[Fri02] Fridrich, J. (2002). Security of fragile authentication watermarks with localization. Proceedings of the IS&T/SPIE International Symposium on Electronic Imaging'02, volume 4675, pages 691-700.

[FG99] Fridrich, J. & Goljan, M. (1999). Protection of Digital Images using Self Embedding. Proceeding of The Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, pages146-152.

[FG00] Fridrich, J. & Goljan, M. (2000). "Robust hash functions for digital watermarking," Proceedings of IEEE International Conference on Information Technology: Coding and Computing, pages 178–183.

[FGB00] Fridrich, J., Goljan, M., & Baldoza, A. (2000). New fragile authentication watermark for images. Proceedings of the IEEE International Conference on Image Processing, ICIP'00, pp 446-449.

[FGD01] Fridrich, J., Goljan, M. & Du, R. (2001). Invertible authentication watermark for JPEG images. Proceeding of the IEEE International Conference on Information Technology, ICIT'01, pages 223–227.

[FGD02] Fridrich, J., Goljan, M. & Du, R. (2002). "Lossless data embedding - new paradigm in digital watermarking,"EURASIP Journal on Applied Signal Processing, vol. 02, no. 2, pages 185–196.

[FGM00] Fridrich, J., Goljan, M., & Memon, N. (2000). Further attack on Yeung-Mintzer watermarking scheme. Proceedings of the SPIE Conference on Security and Watermarking of Multimedia Contents II, volume 3971, pages 428-437.

[FL06] Fauzi, M. F. A. & Lewis, P. H. (2006). Automatic texture segmentation for content-based image retrieval application. Pattern Analysis & Applications, volume 9 (4), pages 307-323.

[GK90] Gotlieb, C.C. & Kreyszig, H.E. (1990). Texture descriptors based on co-occurrence matrices. Computer Vision, Graphics and Image Processing volume 51, pages 70-86.

[Har79] Haralick, R. (1979). Statistical and structural approches to textures. Proceedings of the IEEE, volume 67 (5), pages 786–804.

[HL04] Hsu, C.-Y. & Lu, C.-S. (2004). Geometric distortion-resilient image hashing system and its application to scalability. Proceedings of the Workshop on multimedia and security'04, ACM Press, pages 81-92.

[HM00] Holliman, M. & Memon, N. (2000). Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. IEEE Transactions on Image Processing, 9(3), pages 432-441.

[HR04] Howarth, P. & Rüger, S. (2004). Evaluation of Texture Features for Content-based Image Retrieval. Book chapter of Image and Video Retrieval, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, ISSN 0302-9743, volume 3115, pages 978-1039.

[HSL02] Horng, M.-H., Sun, Y.-N., & Lin, X.-Z. (2002). Texture feature coding method for classification of liver sonography. Computerized Medical Imaging and Graphics, vol. 26 pages 33–42.

[JBAN04] Jagadish, N., Bhat, P. S., Acharya, R. & Niranjan, U. C. (2004). Simultaneous storage of medical images in the spatial and frequency domain : a comparative study, Biomedical Engineering Online, Vol 3, no. 1.

[JDM00] Jain, A. K., Duin, R. P. & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22(1), pages 4–37.

[KF01] Kong, X. & Feng, R. (2001). Watermarking Medical Signals for Telemedicine, IEEE Transactions on Information Technology in Biomedicine, volume 5, no. 3, pages 195-201.

[Kim05] Kim, H. (2005). A new public-key authentication watermarking for binary document images resistant to parity attacks. Proceedings of the IEEE International Conference on Image Processing ICIP'05, pages 254-261.

[KH99] Kundur, D. & Hatzinkos, D. (1999). Digital watermarking for telltale tamper proofing and authentication. Proceedings of IEEE, 87(7), pages 1167-1180.

[KNO03] Kailasanathan, C., Naini, R. & Ogunbona, P. (2003). Compression tolerant DCT based image hash. Proceedings of the IEEE Distributed computing systems workshops'03, pages 562-567.

[KVH00] Kutter, M., Voloshynocskiy, S. & Herrigel, A. (2000). The Watermark Copy Attack. Proceedings of SPIE Security and Watermarking of Multimedia Content, San Jose, USA, P. W. Wong, E. J. Delp Editors, vol. 3971, pages 371-380.

[KZ95] Koch, E. Zhao, J. (1995). Towards robust and hidden image copyright labeling, Proccedings IEEE Workshop on Nonlinear Signal and Image Processing, pages 123-129.

[LC98] Lin, C.-Y., & Chang, S.-F. (1998). Generating Robust Digital Signature for Image/Video Authentication. Proceedings of Multimedia and Security Workshop at ACM Multimedia' 98, Bristol, UK, pages 677-680.

[LC00] Lin, C.-Y., & Chang, S.-F. (2000). Semi-Fragile Watermarking for Authenticating JPEG Visual Content. SPIE International Conference on Security and Watermarking of Multimedia Contents, San Jose, USA, vol. 3971, No 13, pages 140–151.

[LC01] Lin, C.-Y., & Chang, S.-F., (2001). A robust image authentication method distinguishing JPEG Compression from malicious manipulation. IEEE Transactions on Circuits and Systems of Video Technology, 11(2), pages 153-168.

[LL00] Lou, D.-C., & Liu, J.-L. (2000). Fault resilient and compression tolerant digital signature for image authentication. IEEE Transactions on Consumer Electronics, 46(1), pages 31-39.

[LL03] Lu, C.-S. & Liao, H.-Y. (2003). Structural Digital Signature for Image Authentication: an Incidental Distortion Resistant Scheme. IEEE Transactions on Multimedia, Vol. 5, pages 161-173.

[LLC00] Li, C.-T., Lou, D.-C., & Chen, T.-H. (2000). Image authentication via content-based watermarks and a public key cryptosystem. Proceedings of the IEEE International Conference on Image Processing, III, pages 694-697.

[LML02] Lefèbvre, F., Macq, B., Legat, J.D. (2002). RASH: Radon Soft Hash algorithm, Proceeding of 11th European Signal Processing Conference, Toulouse, France, pages.

[LZ04] Liu, Y. & Zhou, X. (2004). Automatic texture segmentation for texture-based image retrieval. Proceedings of the International Conference on Multimedia Modelling, Kent Ridge, Singapore, pages 285–290.

[MBSH04] McCarthy, E., Balado, F., Silvestre, G. & Hurley, N. (2004). A framework for soft hashing and its application to robust image hashing. Proceedings of the IEEE International Conference on Image Processing, ICIP'04, volume 1, pp 397-400.

[ME04] Monga, V. & Evans, B. (2004). Robust perceptual image hashing using distributed coding. Proceedings of the IEEE International Digital Signal Processing Workshop - DSPWS 2004, volume 1, pages 677-680.

[MHDG90] Marceau, D.J., Howarth, P.J., Dubois, J.-M.M. & Gratton, D.J. (1990). Evaluation of grey level co-occurrence matrix method for land-cover classification using SPOT imagery. IEEE Transactions on Geoscience and Remote Sensing volume 28 (4), pages 513–519.

[MM96] Manjunath, B. & Ma, W. (1996). Texture features for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 18, pages 837-842.

[MSC96] Manjunath, B.S., Shekhar, C. & Chellappa, R.(1996) A new approch to image feature detection with applications. Pattern Recognition, 29(4), pages 627-640.

[MV01] Mihçak, M. K. & Venkatesan, R. (2001). "New iterative geometric methods for robust perceptual image hashing," Proceedings of ACM Workshop on Security and Privacy in Digital Rights Management, Philadelphia, USA, pages 324-331.

[MWNS00] Manjunath, B., Wu, P., Newsam, S. & Shin, H. (2000). A texture descriptor for browsing and similarity retrieval. Journal of Signal Processing: Image Communication volume16, pages 33-43.

[OP97] Ó Ruanaidh, J.J.K. & Pun, T.(1997). Rotation, Scale and Translation Invariant Digital Image Watermarking. Proceedings of the IEEE International Conference on Image Processing ICIP'97, Santa Barbara, California, vol. 1, pages 536-539.

[Pla02] PLANTE, D. (2002). Tatouage d'image par quantification, Thèse de Doctorat, Université de La Rochelle.

[PJ98] Puate, J. & Jordan, F. (1998). Using fractal compression scheme to embeded a digital signature into an image. *Proceedings. of SPIE Photonics East Symposium*, Boston, USA, vol. 1, pages 58-64.

[PS00] Portilla, S. & Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients, International Journal of Computer Vision, volume 40(1), pages 49–71.

[Que98] Queluz, M. P. (1998). Towards Robust, Content Based Techniques for Image Authentication., Proceedings of the IEEE Second Workshop on Multimedia Signal Processing, Redondo Beach, CA, USA, pages 297-302.

[Que01] Queluz, M. P. (2001). Authentication of digital images and video: generic models and a new contribution. Signal Processing Journal : Image Communication, Vol. 16, pages 461-475.

[QC05]. Qibin, S. & Chang, S.-F. (2005). A secure and robust digital signature scheme for JPEG2000 image authentication, IEEE Transactions on Multimedia, 7(3)/480-494, 2005.

[RD00] Rey, C. & Dugelay, J.-L.(2000). Blind Detection of Malicious Alterations On Still Images Using Robust Watermarks. IEE Secure Images and Image Authentication colloquium, London, UK.

[RD02] Rey, C. & Dugelay, J.L. (2002), 'A survey of watermarking algorithms for image authentication', EURASIP Journal on Applied Signal Processing, Vol 02, Issue 6, pp. 613–621.

[RSA77] Rivest, R. L., Shamir, A. & Adelman, L. (1997). On Digital Signatures and Public Key Cryptosystems. MIT Laboratory for Computer Science Technical Memorandum 82.

[SMW04] Swaminathan, A., Mao, Y. & Wu, M. (2004). Image hashing resilient to geometric and filtering operations, Proceedings of IEEE Workshop on Multimedia Signal processing , pages 57-63.

[Sta03] Stallings, W. (2003). Cryptography and Network Security: Principles and Practices (3rd edition), Prentice Hall.

[Sti03] Stinson, D. (2003). Cryptographie, théorie et pratique, Edition Vuibert, Paris.

[Tia02] Tian, J. (2002) "Reversible watermarking by difference expansion," in Proc. of Workshop on Multimedia and Security,J. Dittmann, J. Fridrich, and P. Wohlmacher, Eds.,Dec. 2002, pp. 19–22.

[TSS05] Thiemert, S., Sahbi, H. & Steinebach, M. (2005). Applying interest operators in semi-fragile video watermarking. In IS&T/SPIE International Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII, volume 5681, pages 353-362.

[Tsu92]Tsudik, G.(1992). Message Authentication with One-Way Hash Functions, Proceedings of IEEE Infocom'92. Volume 22(5), pages 29-38.

[VDM01] De Vleeschouwer, C., Delaigle, J. F. & Macq, B. (2001). Circular interpretation of histogram for reversible watermarking, Proc. of IEEE 4th Workshop on Multimedia Signal Processing.

[VDM03] De Vleeschouwer, C., Delaigle, J. F. & Macq, B. (2003). Circular interpretation of bijective transformations in lossless watermarking for media asset management," IEEE Transactions on Multimedia, vol. 5, no. 1, pages 97–105.

[VKJM00] Venkatesan, R., Koon, S.M., Jakubowski, M.H. & Moulin, P. (2000). Robust image hashing. Proceedings of the IEEE International Conference on Image Processing, ICIP '00, Vancouver, Canada, pages 68-83.

[VTO98] Van Schyndel, R.G., Tirkel, A. Z. and Osborne, C. F.(1998). A Digital Watermark. Proceedings IEEE International Conference on Image Processing (ICIP'98), Austin, Texas, Vol. 2, pages 86-90.

[VVB03] Van Leest, A., Van der Veen, M., & Bruekers, F. (2003). Reversible image watermarking. Proceedings of the IEEE International Conference on Image Processing, II.

[Wak02] Wakatani, A. (2002) Digital watermarking for ROI medical images by using compressed signature image, Proceedings of Annual Hawaii International Conference on System Sciences, Hawaï, USA, pages 2043– 2048.

[Wal95] Walton, S. (1995). Information Authentification for a Slippery New Age. *Dr. Dobbs Journal*, vol. 20, No. 4, pages 18-26.

[Wan94] Wang, L. (1994). Vector Choice in the Texture Spectrum Approach. International Journal of Remote Sensing, Vol. 15, no. 18, pages 3823-3829.

[WBSS04] Wang, Z., Bovik, A.C., Sheikh, H.R. & Simoncelli, E.P. (2004). Image Quality Assessment: From Error Measurement to Structural Similarity IEEE Transactions On Image Processing, vol.13, no.1, pages 48-54.

[WD99] Wolfgang, R.B., & Delp, E. J.(1999). Fragile Watermarking Using the VW2D Watermark. SPIE International Conference on Security and Watermarking of Multimedia Contents, vol. 3657, No. 22, San Jose, USA, pages 658-663.

[WM00] Wong, P.-W. & Memon, N. (2000). Secret and public key authentication watermarking schemes that resist vector quantization attack. Proceeding of the SPIE conference on Security and Watermarking of Multimedia Contents II. pages

[WM01] Wong, P.-W. & Memon, N. (2001). Secret and public key image watermarking schemes for image authentication and ownership verification. IEEE Transactions on Image Processing, 10(10):1593-1601.

[Won98] Wong, P.-W. (1998). A public key watermark for image verification and authentication. Proceeding of the IEEE International Conference on Image Processing, pages 455-459.

[Wu02] Wu, C. (2002). On the design of content-based multimedia authentication systems. IEEE Transactions on Multimedia, volume 4(3) pages 385-393.

[WL98] Wu, M. & Liu, B. (1998). Watermarking for image authentication. Proceeding of the IEEE International Conference on Image Processing, II, 437-441.

[WT03] Wu, D. C., Tsai, W.H. (2003). A Steganographic Method for Images by Pixel-Value Differencing, Pattern Recognition Letters, Vol.24, pages 1613-1626.

[XA01] Xie, L. & Arce, G. R. (2001). A class of authentication digital watermarks for secure multimedia communication. IEEE Transactions on Image Processing, 10(11), 1754-1764.

[XZCSNS02] Xuan, G., Zhu, J., Chen, J., Shi, Y. Q., Ni, Z. & Su, W. (2002). Circular interpretation of bijective transformations in lossless watermarking for media asset management, IEE Electronics Letters, vol. 38, no. 25, pages 1646–1648.

[YC05] Yang, S. & Chen, C. (2005). Robust image hashing based on SPIHT. In Proc IEEE Information Technology: Research and Education – ITRE'05, pages 110-114.

[YM97] Yeung, M. & Minzter, F. (1997). An Invisible Watermarking technique for image verification. Proceeding of the IEEE International Conference on Image Processing, I, 680-683.

[ZSQIT06] Zaim, A., Sawalha, A., Quweider, M., Iglesias, J., Tang, R. (2006). A New Method for Iris Recognition using Gray-Level Coccurence Matrix, Proceedings of IEEE International Conference on Electronic information Technology'06, Pages 350 – 353.

[ZT02] Zhang, J. & Tan, T. (2002). Brief review of invariant texture analysis methods. Pattern Recognition, volume 35, pages 735–747.

Webographie

[Web1] The MD5 Message-Digest Algorithm, DDN Network Information Center, http://www.ietf.org/rfc/ rfc1321.txt.

[Web2] SHA-1, Secure Hash Standard (SHS), spécification (FIPS 180-1), http://www.itl.nist.gov/fipspubs/fip180-1.htm

[Web3] IST – CERTIMARK: a benchmark suite for watermarking of visual content and a certification process for watermarking algorithms. http://www.certimark.org

[Web4] E. Roskis. Images truquées. Le Monde Diplomatique, Jan. 1998 http://www.monde-diplomatique.fr/.

[Web5] Nishio, M., Kawashima, Y., Nakamuar, S. & Tsukamoto, N. (2002). Development of a digital watermark method suitable for medical images with error correction, RSNA'02, ArchiveSite: http://archive.rsna.org/index.cfm.

[Web6] Understanding and Intergrating KODAK Picture Authentication Cameras. http://www.kodak.com/US/en/digital/software/imageAuthentication/

[Web7] Signum Technologies. http://www.signumtech.com/

[Web 8] U.S. Department of Commerce, National Institute of Standards and Technology (FIPS PUB 186-1). http://www.itl.nist.gov/fipspubs/bynum.htm

[Web 9] Mediasec. http://www.mediasec.com

[Web10] Agence Reuters. http://www.reuters.com

[Web11] ARTUS Project. http://www.icp.inpg.fr/~elisei/ARTUS/Modeles.html

[Web12] http://www.distributed.net/pressroom/news

[Web 13] RSA Public Key Cryptography. http://www.rsa.com/rsalabs/

[Web 14] Open PGP Alliance. http://www.openpgp.org

[Web 15] RACE ,Integrity Primitives Evaluation. http://www.esat.kuleuven.be/ripemd160.html

[Web 16] Interconnexion de systèmes ouverts -- Modèle de référence de base. http://www.iso.org/iso/fr/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=14256

[Web 17] Krawczyk, H. Bellare, M. Canetti, R. Request for Comments : 2104. http://www.ietf.org/rfc/rfc2104.txt