

**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**



**Université Cheikh Larbi Tbessi –Tébessa**  
**Faculté des Sciences de l'Ingénieur**  
**Département Informatique**  
Ecole doctorale en Informatique de l'Est IA & GL  
(EDI Est IA & GL)  
Pôle Annaba



N° D'ORDRE : .....  
SERIE : .....

Détection des Informations Cachées  
dans les Images Numériques  
Basée loi de Zipf

MEMOIRE PRESENTE PAR

**Laimeche Lakhdar**

Pour l'obtention du diplôme de Magistère en informatique

**Option:** Intelligence Artificielle

**Soutenu le 10/01/2010**

**Devant le Jury composé de:**

<b>Président</b>	:	Dr BOUHADADA Tahar	(Université d'Annaba)
<b>Rapporteur</b>	:	Dr MEROUANI Hayet Farida	(Université d'Annaba)
<b>Examineur</b>	:	Dr GHOUALMI Nacira	(Université d'Annaba)
		Dr TLILI Yamina	(Université d'Annaba)

# ABSTRACT

Steganalysis is the art of studying a document to decide if it contains a hidden message; it consists to attacking the steganographic methods by detection, destruction, retrieval or modification of data.

Detection methods can be classified in two types of steganalysis: specific or universal. Specific steganalysis can specifically answer to the question, "the medium does it be marked with the algorithm A?". The universal Steganalysis can answer to the question " does the medium marked ?" .In other words, the measures used for detection are independent of algorithms that we are trying to detect.

In this work, we propose a universal method of Steganalysis based on statistical deviations. The statistical characteristics that we define for the detection of hidden information are based on the use of Zipf's law. The application of Zipf's law is based on statistical distribution of patterns in the image. This law makes it possible to extract certain number of parameters which can characterize the structure of an image.

The whole of these parameters which translates to some degree the specific mark to each image represents a tool of a great interest to carry out discrimination between images. Fisher linear discriminant (FLD) is then used to find a threshold that separates stego-images from cover-images.

**Keywords:** Hidden information, Steganalysis, Specific Steganalysis, Universal Steganalysis, steganography, statistical deviations, Zipf's law, FLD

## RESUME

La stéganalyse est la contre partie de la stéganographie, elle consiste à l'étude d'un document afin de décider si ce dernier contient un message cachée. Les méthodes de stéganalyse peuvent être répertoriées selon le type de la stéganalyse: spécifique ou universelle.

La stéganalyse spécifique permet de répondre à la question « *le médium a t-il été stéganographié avec l'algorithme A?* », elle est dédiée a un algorithme que nous essayons de détecter. La stéganalyse universelle permet de répondre à la question « *le médium est-il stéganographié ?* », autrement dit, les mesures utilisées pour la détection sont indépendantes des algorithmes que nous essayons de détecter.

Dans ce travaille, nous proposons une méthode de stéganalyse universelle basée sur les déviations statistiques. Les caractéristiques statistiques que nous avons définit pour la détection des informations cachées sont basées sur l'utilisation de la loi de Zipf. L'application de la loi de Zipf consiste à l'étude des changements statistiques des motifs présents dans l'image (motifs des pixels, coefficients DCT). Cette loi permet d'extraire un certain nombre de paramètres qui peuvent caractériser la structure d'une image.

La méthode de stéganalyse proposée se fait en deux étapes principales: d'une part l'extraction des caractéristiques des images pour former l'espace de caractéristiques, ou l'espace de représentation, et d'autre part l'utilisation d'un discriminant linéaire de Fischer (DLF) permettant de classer une nouvelle image à l'une des deux classes d'images (propres, stéganographiées).

**Mots clés:** dissimulation d'information, stéganalyse, stéganalyse spécifique, stéganalyse universelle, stéganographie, déviations statistiques, loi de Zipf, discriminant linéaire de Fischer (DLF).

## REMERCIEMENTS

Tout d'abord, je voudrais remercier Mme **Hayet Farida MEROUANI**. Elle a toujours pris sur son temps précieux pour m'écouter amicalement et de m'aider à trouver le bon chemin dans ce travaille. Je voudrais aussi la remercier pour ses conseils qui m'ont permis de prendre du recul pour prendre les bons choix. Ce travaille n'aurait pas vu le jour sans la confiance qu'elle a placé en moi. Je dis "Merci" en espérant que ce simple mot peut exprimer ce que je ne peu pas.

Je voudrais ensuite remercier **Mr BOUHADADA TAHAR**, Docteur d'état à l'université Badji Mokhtar- Annaba. Je voudrais le remercie pour l'honneur qu'il me fait en président ce jury. Je ne sais pas comment je pourrai exprimer mes remerciements et mes reconnaissances devant ses bienfaits.

Je tiens également à exprimer ma gratitude et mes remerciements à **Dr Ghoualmi Nacira**, Docteur d'état à l'université Badji Mokhtar-Annaba, et **Dr Tlili Yamina**, Docteur d'état à l'université Badji Mokhtar-Annaba, de l'honneur qu'elles mon faites pour participer à mon jury de Magistère.

Je ne pouvais bien évidemment pas conclure mes remerciements sans saluer la compréhension de ma femme, mes filles et ma famille qui ont accepté mon manque de disponibilité et mes retards devant mes obligations.

# DEDICACE

*Je dédie ce travail à mon père, ma mère, mon  
épouse, ma petite famille, mes  
frères et à mes amis.*

## LISTE DES TABLEAUX

<b>Tableau 1.1</b> : Schéma générale en dissimulation.....	17
<b>Tableau 3.1</b> : Correspondance entre indice et élément.....	48
<b>Tableau 3.2</b> : Occurrences de chaque élément.....	48
<b>Tableau 3.3</b> : Résultat de la fonction d'homogénéité sur l'image de base.....	51
<b>Tableau 3.4</b> : Résultat de la fonction d'homogénéité et classification.....	52
<b>Tableau 3.5</b> : Récapitulation des résultats obtenus.....	53
<b>Tableau 3.6</b> : Modifications des différents ensembles sous insertion LSB.....	54
<b>Tableau 4.1</b> : Vecteur de caractéristiques extrait des images propres et stégo-images.....	76
<b>Tableau 4.2</b> : Calcule des paramètres du classifieur.....	84
<b>Tableau 4.3</b> : Matrice de confusion pour un taux stéganographique de 10%.....	85
<b>Tableau 4.4</b> : Matrice de confusion pour un taux stéganographique de 15%.....	85
<b>Tableau 4.5</b> : Probabilités de succès de la stéganalyse basée Zipf et la Stéganalyse basée mesure de similarité binaire.....	86

# LISTE DES FIGURES

<b>Figure 1.1 :</b>	Classification des techniques de dissimulation d'information.....	6
<b>Figure 1.2 :</b>	Exemple d'une attaque mosaïque.....	11
<b>Figure 1.3 :</b>	Détection des régions manipulées.....	12
<b>Figure 1.4 :</b>	Exemple d'un tatouage visible.....	13
<b>Figure 1.5 :</b>	Exemple d'un tatouage invisible.....	13
<b>Figure 1.6 :</b>	Etape de dissimulation pour une image fixe.....	14
<b>Figure 1.7 :</b>	Etape d'extraction pour une image fixe.....	15
<b>Figure 1.8 :</b>	Compromis entre Imperceptibilité, Capacité et la Robustesse.....	16
<b>Figure 1.9 :</b>	Diagramme représentant la dissimulation d'information vs Cryptographie.....	18
<b>Figure 1.10 :</b>	Principe de chiffrement asymétrique .....	19
<b>Figure 1.11 :</b>	Complexité de la distribution de clés symétriques .....	20
<b>Figure 1.12 :</b>	Mécanisme asymétrique .....	21
<b>Figure 2.1 :</b>	Une image en noir et blanc (a), niveaux de gris (b) et en couleurs(c) ...	25
<b>Figure 2.2:</b>	Diagramme représentant quelques formats d'images .....	26
<b>Figure 2.3 :</b>	Fragment d'un fichier image.bmp en hexadécimal .....	27
<b>Figure 2.4 :</b>	Schéma de compression/décompression JPEG .....	28
<b>Figure 2.5 :</b>	Découpage d'une image en blocs de $8 \times 8$ valeurs .....	29
<b>Figure 2.6 :</b>	Décomposition d'une image suivant les composantes RVB et YCbCr...	30
<b>Figure 2.7 :</b>	Parcours en Zigzag d'un bloc $8 \times 8$ .....	31
<b>Figure 2.8 :</b>	Exemple d'ajout en fin de fichier JPEG .....	33
<b>Figure 2.9 :</b>	Insertion dans les bits de poids faible .....	35
<b>Figure 2.10 :</b>	255ème rouge (a), 254ème rouge (b) .....	35
<b>Figure 2.11 :</b>	image Rouge original (a), stéganographiée avec Invisible secret (b)...	35
<b>Figure 2.12 :</b>	Fragment du fichier Rouge.bmp en Hexadécimal.....	36
<b>Figure 2.13 :</b>	Fragment du fichier Rouge stéganographié.bmp en Hexadécimal.....	36
<b>Figure 2.14 :</b>	Différentes couleurs rouge utilisées pour la dissimulation. ....	37
<b>Figure 2.15 :</b>	Matrice de quantification .....	39
<b>Figure 3.1 :</b>	Image originale.....	44
<b>Figure 3.2 :</b>	Dernier plan de bit avant et après insertion de l'image dégradée.....	45
<b>Figure 3.3 :</b>	Image Lena originale.....	45
<b>Figure 3.4 :</b>	Dernier plan de bit avant et après insertion de l'image Lena.....	45
<b>Figure 3.5 :</b>	Partie d'un histogramme d'une image avant et après insertion.....	46
<b>Figure 3.6 :</b>	Image notwindow (a) et son histogramme (b) .....	56
<b>Figure 4.1 :</b>	Motif original (a), motif codé avec la méthode des rangs généraux (b) .	61
<b>Figure 4.2 :</b>	Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Lena.bmp.....	62
<b>Figure 4.3 :</b>	Courbe de Zipf obtenue avec le codage des rangs généraux à partir De l'image Masuda non stéganographiée.....	64
<b>Figure 4.4:</b>	Courbe de Zipf obtenue avec le codage des rangs généraux à partir De l'image Masuda stéganographiée.....	65
<b>Figure 4.5 :</b>	Différence entre les courbes de Zipf associées à l'image Masuda originale et l'image Masuda Stéganographiée.....	65
<b>Figure 4.6 :</b>	Courbe de Zipf obtenue avec le codage des rangs généraux à partir	67

	de l'image Tiffany non stéganographiée.....	
<b>Figure 4.7 :</b>	Logiciel de stéganographie Caméléon 1.0.....	68
<b>Figure 4.8 :</b>	Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Tiffany stéganographiée par Cameleon1.0.....	68
<b>Figure 4.9:</b>	Différents plans de bit de l'image Lena.....	69
<b>Figure 4.10 :</b>	Images Eau, Lena, Arbre et singe.....	71
<b>Figure 4.11:</b>	Conversion des images Arbre, Lena, Singe et Eau.....	72
<b>Figure 4.12:</b>	Logiciel de stéganographie Cameleon 1.0.....	73
<b>Figure 4.13:</b>	Logiciel de stéganographie Invisible secrets 4.0.....	73
<b>Figure 4.14:</b>	Logiciel de stéganographie JPHS pour windows 5.0.....	74
<b>Figure 4.15:</b>	Extraction des caractéristiques.....	74
<b>Figure 4.16:</b>	Dissimulation d'information dans les images Lena.JPEG, Singe.BMP et Eau.GIF.....	75
<b>Figure 4.17:</b>	Représentation des nuages pour $p = 3$ et $k = 2$ .....	79
<b>Figure 4.18:</b>	Images extraites de la base Philip.....	83





## TABLE DES MATIERES

ملخص .....	I
ABSTRACT.....	II
RESUME.....	III
DEDICACES.....	IV
REMERCEMENTS.....	V
LISTE DES TABLEAUX.....	VI
LISTE DES FIGURES.....	VII
TABLE DES MATIERES.....	VIII
INTRODUCTION GENERALE.....	1
<b>CHAPITRE 1 : LA DISSIMULATION D'INFORMATION</b>	
1.1. Introduction.....	5
1.2. Terminologie.....	6
1.3. Classification de la dissimulation d'information.....	6
1.3.1 La stéganographie.....	7
1.3.1.1 Qu'est ce que la stéganographie.....	7
1.3.1.2 Pour quoi la stéganographie .....	7
1.3.1.3 Formes possible de la stéganographie .....	7
1.3.1.3.1 La stéganographie linguistique.....	7
1.3.1.3.2 La stéganographie technique.....	8
1.3.1.4 Types des supports.....	9
1.3.1.4.1 Dissimuler de l'information dans du texte.....	9
1.3.1.4.2 Dissimuler de l'information dans les images.....	10
1.3.1.4.3 Dissimuler de l'information dans du son .....	10
1.3.1.4.4 Dissimuler de l'information dans la vidéo.....	10
1.3.1.4.5 Autres supports.....	10
1.3.2. Le tatouage.....	11
1.3.2.1 Applications du Tatouage.....	12
1.3.2.2 Types du tatouage.....	12
1.3.3. Le filigrane.....	14
1.4. Processus de la dissimulation.....	14
1.4.1. Insertion.....	14
1.4.2. Extraction.....	15
1.5. Critères de dissimulation d'information.....	15
1.5.1 La capacité.....	15
1.5.2 L'imperceptibilité.....	16
1.5.3 La robustesse.....	16
1.6. Comparaison entre les techniques de la dissimulation.....	17
1.7. Stéganographie VS Cryptographie.....	18
1.7.1 Notions de la cryptographie.....	18
1.7.2 Types de chiffrement.....	18
1.7.2.1 Algorithmes de cryptographie symétrique.....	18
1.7.2.2 Algorithmes de cryptographie asymétrique.....	20
1.8. Conclusion.....	21

## CHAPITRE 2 : LA STEGANOGRAPHIE ADAPTEE AUX IMAGES FIXES

2.1	Introduction.....	23
2.2	Les images numériques.....	24
2.2.1	Types des images.....	24
2.2.1.1	Les images de type Bitmap.....	24
2.2.1.2	Les images de type vectoriel.....	24
2.2.2	Le codage d'une image numérique.....	24
2.2.3	Formats des images.....	26
2.2.3.1	Image BMP.....	26
2.2.3.2	Image GIF.....	27
2.2.3.3	Image JPEG.....	28
2.2.4	Compression JPEG.....	28
2.3	Technique de la stéganographie.....	32
2.3.1	Stéganographie basée sur la structure du fichier.....	33
2.3.2	Stéganographie dans le domaine spatial.....	34
2.3.2.1	Domaine spatial.....	34
2.3.2.2	Stéganographie dans des bits de poids faible (LSB).....	34
2.3.2.3	Stéganographie dans les images GIF.....	37
2.3.3	Stéganographie dans le domaine fréquentiel.....	38
2.3.3.1	Domaine fréquentiel.....	38
2.3.3.2	Stéganographie dans les coefficients DCT.....	38
2.4	Conclusion.....	40

## CHAPITRE 3 : LA STEGANALYSE DANS LE DOMAINE SPATIAL

3.1	Introduction.....	41
3.2	Description de la stéganalyse.....	42
3.3	Types de stéganalyse.....	43
3.3.1	Attaque active.....	43
3.3.2	Attaque passive.....	43
3.4	Les méthodes de la stéganalyse.....	43
3.4.1	Stéganalyse universelle.....	44
3.4.1.1	Attaque visuelle.....	44
3.4.1.2	Stéganalyse basée sur des paires de valeurs des pixels de l'image.....	46
3.4.1.2.1	L'analyse basée test $\chi^2$ .....	46
3.4.1.2.2	Stéganalyse RS (Régulier, Singulier).....	49
3.4.1.2.3	Stéganalyse basée sur les égalités statistiques.....	54
3.4.1.3	Méthodes de stéganalyse dans le domaine spatial.....	56
3.4.2	Stéganalyse spécifique.....	57
3.5	Conclusion.....	58

## CHAPITRE 4 : CONCEPTION ET REALISATION

4.1	Introduction.....	59
4.2	Principe de la loi de Zipf.....	60
4.2.1	Application aux images.....	60
4.2.2	Codage des motifs.....	61
4.2.3	Analyser une image avec la loi de Zipf.....	61
4.3	Justification de choix de la stéganalyse basée loi de Zipf.....	63
4.4	Principe de la stéganalyse proposée.....	63
4.4.1	Extraction des caractéristiques.....	64

4.4.1.1	La pente $\alpha$ .....	66
4.4.1.2	L'entropie.....	66
4.4.1.3	ordonné à l'origine du graphe de Zipf.....	67
4.4.1.4	la qualité Zipf ZQ (Zipf quality).....	69
4.4.1.5	Corrélation linéaire.....	70
4.4.1.6	Utilisation de l'écart type.....	71
4.4.2	Evaluation du vecteur de caractéristiques.....	71
4.4.3	Méthode de classification.....	77
4.4.3.1	Mesures de performance du classifieur.....	78
4.4.3.2	Justification de choix de classifieur.....	78
4.4.3.3	Discriminant Linéaire de Fischer.....	79
4.5	Testes et résultats.....	82
4.6	Utilisation du détecteur ZIPF.....	86
4.6.1	Fenêtre principale.....	86
4.6.2	Afficher une image dans le $i^{\text{ème}}$ plan de bit.....	87
4.6.3	Extraction des caractéristiques basée loi de Zipf.....	87
4.6.4	Détection des images stéganographiées.....	88
4.7	Conclusion.....	88
	CONCLUSION ET PERSPECTIVE.....	90
	REFERENCES BIBLIOGRAPHIQUES.....	92
	ANNEXE 1.....	97
	ANNEXE 2.....	98
	ANNEXE 3.....	99

## INTRODUCTION GENERALE

### 1. PROBLEMATIQUE

Le développement des réseaux de communication et des supports numériques a facilité le partage et le transfert des données numériques, introduisant ainsi de nouvelles formes de piratage de documents et de nouveaux défis de sécurité à relever.

La confidentialité des communications est le plus souvent assurée par la cryptographie : l'information subit un traitement particulier, appelé chiffrement, visant à la rendre incompréhensible par toute personne non autorisée.

Cette information, une fois chiffrée, peut être librement transmise au travers d'un canal susceptible d'être écouté : sa confidentialité n'est pas exposée puisque son sens a été complètement masqué.

Une donnée cryptée attire l'attention dans une masse de données en clair. Il est évident pour un pirate, que les données chiffrées sont les données les plus intéressantes.

De plus, le problème de la protection du contenu d'un support numérique multimédia ne connaît pas encore de solutions satisfaisantes. Il est devenu aisé de modifier ou de reproduire un média et même de revendiquer ses droits d'exploitation.

Afin de diminuer la copie des œuvres multimédias, assurer la confidentialité d'une transmission, protéger l'intégrité des documents et contribuer à la protection du copyright, des nouvelles méthodes ont été développées. Il s'agit des méthodes de dissimulation d'information.

La dissimulation d'information cherche à cacher une information de n'importe quel type dans un autre support qui peut être de type texte, image, audio ou vidéo. Les applications de la dissimulation se distinguent par leurs objectifs. En stéganographie, le but est de cacher un message dans un support numérique pour permettre à des partenaires de communiquer d'une façon secrète, le support n'a aucun lien avec le message à envoyer. Le tatouage numérique consiste à insérer une marque qui a un lien avec le support numérique. Il est utilisé pour la protection des droits d'auteurs, la protection des copies, l'indexation et la vérification de l'intégrité du document.

Si la marque insérer dans un support numérique est différentes pour toutes les copies du support de base, on parle alors du filigrane, le but principal de ce dernier est de tracer la source des copies illégales.

Malgré que les objectifs soient distincts, ces trois approches partagent des points communs : un support pour la dissimulation (son importance est liée à l'application), des informations à cachées (que se soit un message, marque ou une empreinte) et une clé pour l'insertion et l'extraction ou détection. La différence entre la stéganographie et le tatouage est qu'en tatouage on cherche à marquer le support (on se limite souvent à la dissimulation d'un bit : marquer/pas marquer) pour protéger les droit d'auteurs ou encore de démontrer l'intégrité du document. Une autre différence importante se situe au niveau des attaques. En stéganographie le pirate cherche à lire le message dissimulé dans le support, tandis qu'en tatouage, va chercher à laver le support de toute marque possible.

La stéganographie et la cryptographie sont souvent très proches, on peut dire que ces deux disciplines sont complémentaires. Dans le cas de la stéganographie, la communication n'est pas chiffrée. Elle ne peut pas être détectée par une tierce personne, ce dernier ne se doute pas que les parties communicantes échangent des messages. La cryptographie permet d'établir une liaison sécurisée entre deux parties communicantes en chiffrant la communication ce qui la rend incompréhensible pour une tierce personne.

## **2. MOTIVATION ET OBJECTIFS**

L'objectif principal de la stéganographie est de communiquer sans que cela ce voit. Autrement dit, dissimuler l'existence du message dans un support de caractère anodin. Mais le réel problème de la stéganographie est lorsque cette communication sert à dissimuler aux yeux de la justice des actions illégales.

L'utilisation de la stéganographie paraît bien adaptée au vol d'informations confidentielles, car les messages cachés sont difficilement détectables. De nombreuses personnes peuvent être intéressées par le vol de ce type d'information qui a très souvent une valeur marchande.

Les pirates informatiques peuvent aussi utiliser cette technique pour camoufler leurs attaques. Un hacker peut très bien dissimuler des codes fragmentés à travers des stégo-medium (ex: images) et procéder aux réassemblages du code malveillant directement sur l'ordinateur de la victime. Le hacker peut également dissimuler un cheval de Troie et prendre possession de la machine.

C'est la raison pour laquelle, il est nécessaire de prendre des mesures de sécurité liés à la mauvaise utilisation de la stéganographie.

Il existe des techniques permettant de découvrir les média stéganographiés: c'est le cas de la stéganalyse, appelé aussi l'analyse stéganographique.

La stéganalyse est la technique qui permet de déceler la stéganographie. Il existe deux types de stéganalyse. La stéganalyse dite passive, il s'agit simplement de détecter la présence de données dissimulées. On peut par exemple imaginer que ce type d'attaque passive soit utilisé pour voir si des données stéganographiées sortent d'une entreprise. La stéganalyse active, dans ce type d'attaque on souhaite non seulement détecter le message caché mais, en plus, on va chercher à extraire, modifier ou supprimer ces données.

Ce travail présente une méthode de stéganalyse utilisant un ensemble de caractéristiques statistiques pour détecter la présence d'un éventuel message caché dans un médium de type image BMP. Nous avons choisis ce format d'image car le domaine utilisé pour la représentation des images est le domaine spatial.

Les caractéristiques statistiques que nous avons défini pour la détection des informations cachées sont basées sur l'utilisation de la loi de Zipf. L'application de la loi de Zipf consiste à l'étude des changements statistiques des motifs présents dans l'image (motifs des pixels, coefficients DCT). Cette loi permet d'extraire un certain nombre de paramètres qui peuvent caractériser la structure d'une image.

La méthode de stéganalyse proposée se fait en deux étapes principales: d'une part l'extraction des caractéristiques des images pour former l'espace de caractéristiques, ou l'espace de représentation, et d'autre part l'utilisation d'un discriminant linéaire de Fischer (DLF) permettant de classer une nouvelle image à l'une des deux classes d'images (propres, stéganographiées).

### **3. ORGANISATION DU DOCUMENT**

Hormis l'introduction générale et la conclusion-perspective, ce document est composé de quatre chapitres qui se présentent comme suit:

- Dans le premier chapitre, nous présentons la terminologie et les objectifs de la dissimulation d'information. La stéganographie, le tatouage et le filigrane sont détaillés pour mieux comprendre la différence entre ces trois techniques de dissimulation d'information.

Nous décrivons par la suite le processus de la dissimulation d'information qui comporte généralement deux fonctions : la fonction d'insertion et la fonction d'extraction, ensuite nous définissons les critères de la dissimulation d'information.

Nous terminons ce chapitre par une comparaison entre les trois techniques de dissimulation (stéganographie, tatouage et filigrane) et entre les deux disciplines complémentaires : la stéganographie et la cryptographie.

- Le deuxième chapitre se divise en deux parties. Nous présentons dans la première partie tout d'abord les notions de bases sur les images fixes, les types d'images, les domaines de représentation des images (domaine spatial, domaine fréquentiel) et la compression JPEG. Cela va nous permettre, dans la deuxième partie, de mieux comprendre comment utiliser la stéganographie dans ce type de support.

Dans la deuxième partie, nous étudions le schéma d'insertion LSB des informations cachées dans le domaine spatial, le domaine fréquentielle et l'insertion basée sur la structure du fichier.

- Le troisième chapitre se veut être une description générale de la contre partie de la stéganographie connue par la stéganalyse ou l'analyse stéganographique. Les méthodes de stéganalyse peuvent être répertoriées soit par rapport aux domaines de représentations des images: spatial ou fréquentielle, soit par rapport à leurs type de stéganalyse : universelle ou spécifique.

Dans ce chapitre, seul des stéganalyses dans le domaine spatial seront détaillées. Pour ce qui est de la stéganalyse dans le domaine fréquentielle, une courte description sera donnée.

- Nous présentons dans le dernier chapitre une nouvelle méthode de stéganalyse dans le domaine spatial pour la détection des informations cachées dans les images fixes basée sur la loi de Zipf. Nous exposons, dans une première partie, le principe général de la loi de Zipf, ensuite nous expliquons comment cette loi peut être appliquée à l'analyse des images. Dans une deuxième partie, nous utilisons la loi de Zipf pour extraire le vecteur de caractéristique permettant la distinction entre les images propres et les stégo-images.

Un discriminant linéaire de Fischer (DLF) est utilisé pour la distinction entre les images propres (non stéganographiées) et les stégo-images.



# **La Stéganographie Adaptée aux Images fixes**

# CHAPITRE 1

## LA DISSIMULATION D'INFORMATION

### 1.1 INTRODUCTION

Depuis l'invention de l'écriture, le besoin de sécurité est motivé par les problèmes de confidentialité et de l'intégrité ; où l'on souhaite éventuellement que l'information écrite ne soit accessible qu'à certaines personnes et qu'elle ne soit pas modifiée volontairement dans un but de mystification.

Deux grandes tendances visent à protéger l'information ; la cryptographie et la dissimulation d'information.

- La cryptographie désigne l'art de chiffrement, dont l'objectif est de rendre l'information incompréhensible à une personne ne possédant pas les connaissances adéquates à son décodage mais ne cherche pas à dissimuler la transmission de l'information.
- La dissimulation d'information désigne l'ensemble des techniques permettant de cacher une information numérique dans un support appelé récipient, par exemple un fichier texte, une image, une audio, un vidéo, un système de fichier, ou un code source.

L'adjectif caché signifie que l'information n'est pas visible par l'oeil humain. Si elle est codée, on parle alors de cryptographie; ceci signifie que la présence de l'information n'est pas perceptible car elle est enfouie dans une autre information.

Les difficultés de la cryptographie résident d'une part, dans le partage de la clé, comment la transmet-on ? Peut-on aisément communiquer une clé sécurisée sur un réseau ? D'autre part, la lenteur des algorithmes de cryptage à clé public par rapport aux algorithmes à clés secrètes.

De plus, une donnée cryptée attire l'attention dans une masse de données en clair. Il est évident, pour un voleur d'information, que les données chiffrées sont les données les plus intéressantes. Pour cette raison certaines personnes se ramènent à la dissimulation d'information plutôt qu'à la cryptographie. Par exemple, vous pouvez user de la dissimulation d'information pour transmettre vos messages en toute liberté et ce, même dans des conditions de censures ou de surveillances. De plus, un message dissimulé attire moins l'attention qu'un message chiffré.

## 1.2 TERMINOLOGIE

Dans la suite de ce mémoire, nous conservons la terminologie définie ci-dessous, afin de distinguer les différents éléments qui interviennent en dissimulation d'information [Ray 02].

- **Cover-medium**: où bien le médium de couverture, il s'agit d'un support numérique dans lequel seront dissimulées les informations. Il peut s'agir d'un texte, d'une image, d'un son, d'une vidéo...
- **stégo-medium** : une fois les informations dissimulées, le cover-medium devient un stégo-medium.
- **Données** : ce sont les informations qui vont être cachées dans le médium de couverture. Il peut s'agir d'un message, une marque ou une empreinte.
- **Stégo-clè** : c'est une information secrète additive, mais essentielle dans tous le processus de dissimulation d'information.

## 1.3 CLASSIFICATION DE LA DISSIMULATION D'INFORMATION

Nous nous intéressons, dans le cadre de ce travail, à la dissimulation d'information et en particulier à la stéganographie. Cette discipline se subdivise en plusieurs catégories (figure 1.1) en fonction des objectifs recherchés [Den 05].

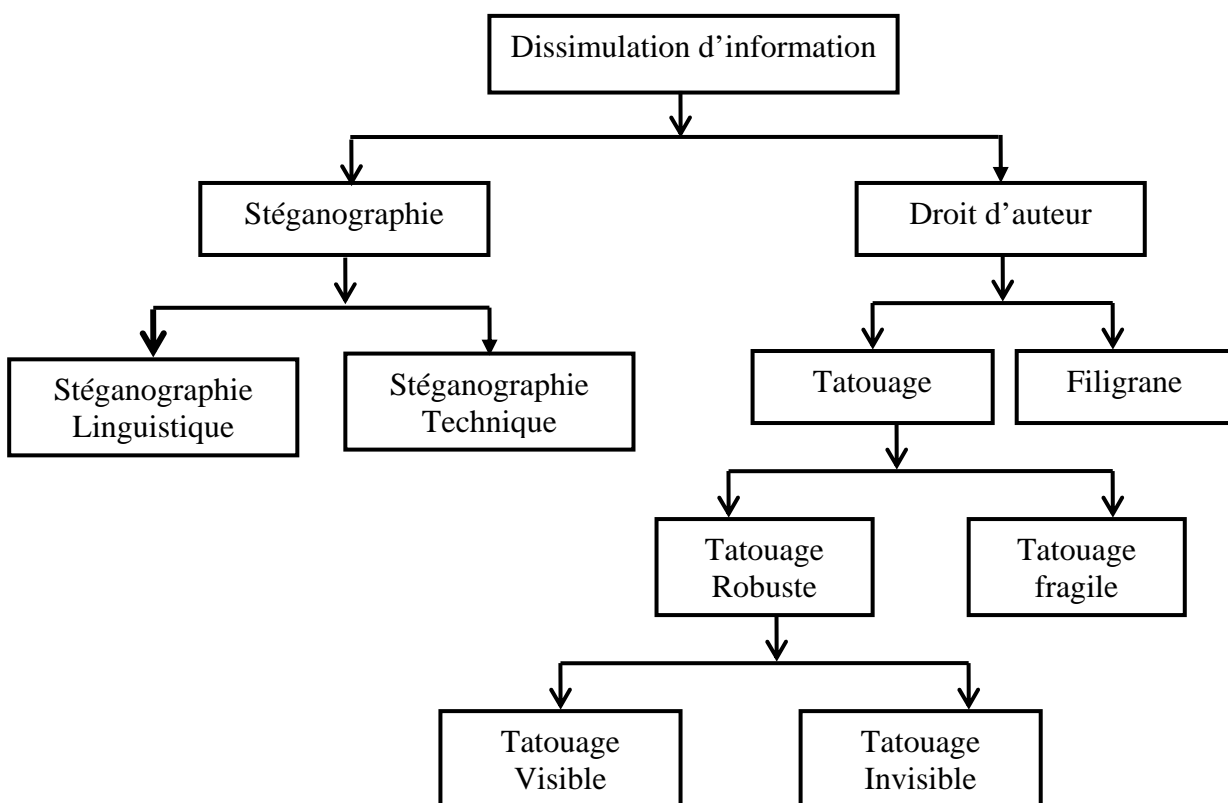


Figure 1.1. : Classification des techniques de dissimulation d'information

### 1.3.1 LA STEGANOGRAPHIE (STEGANOGRAPHY)

#### 1.3.1.1 QU'EST CE QUE LA STEGANOGRAPHIE

Le mot stéganographie vient du grec ‘ steganos ’ (caché ou secret) et ‘ graphy ’ (écriture ou dessin) et signifie latéralement écriture caché.

La stéganographie étudie les techniques pour permettre à des partenaires de communiquer de façon caché en établissant un véritable protocole de communication secrète au dessus d'autres protocoles anodins, c'est se qu'on appelle canal de communication secrète (*cover Channel*), le mot caché signifie que la présence de l'information n'est pas perceptible parce qu'elle vie dans un support d'un caractère anodin qui peut être de type image, vidéo, audio, ou un texte. Le message dissimulé n'a aucun lien avec le support chargé de transport.

#### 1.3.1.2 POUR QUOI LA STEGANOGRAPHIE ?

De nombreux usages peuvent exister dans des domaines très variés mais souvent sensibles :

- Communiquer en toute liberté même dans des conditions de censure et de surveillance.
- Protéger ses communications privées là ou l'utilisation de la cryptographie n'est pas normalement permise car elle soulèverait des suspicions.
- Publier des informations ouvertement mais à l'insu de tous, des informations qui pourront ensuite être révélées.

#### 1.3.1.3 FORMES POSSIBLE DE LA STEGANOGRAPHIE

La stéganographie peut être classée en deux catégories : la stéganographie linguistique et la stéganographie technique. La première comprend toutes formes de styles, jeux de langue ou utilisation de repère au niveau de caractères. La seconde regroupe les moyens de transmissions purement physiques.

##### 1.3.1.3.1 La stéganographie linguistique

La stéganographie linguistique se divise en deux types de camouflage [**web 1.1**]: le sémagramme et le code camouflé.

- **Sémagramme**

Le sémagramme représente une des deux grandes catégories de la stéganographie linguistique. Ce procédé permet de transmettre un message qui n'est pas composé de lettres ou de chiffres mais dont le sens est véhiculé par une combinaison d'objets, de signes ou de symboles. Le système stéganographique échappe totalement à l'observateur.

Alfred de Musset est l'utilisateur le plus connu de ce procédé puisqu'il a entretenu une relation secrète avec Georges Sand (entre 1833 et 1834) au travers de poèmes qu'il lui envoyait [**Gal 04**].

- **Le code camouflé : les nulles**

Le chiffrement par nulles est un genre de code camouflé. Cette méthode consiste à marquer d'un signe particulier sur certaines lettres d'un texte où seules ces quelques lettres sont porteuses de sens. Celles-ci sont dites « repérées ». Le reste des lettres encadrant les lettres repérées sont appelées les nulles car elles sont dépourvues de signification [Den 05].

*Ennés le tacticien* (historien de la Grèce antique) dans ses *mémoires sur la stratégie*, décrit ce type de camouflage, qui consiste, à marquer certaines lettres d'un texte anodin par de petits trous ou encore par la hauteur des lettres, dans le premier cas, le message cachée est obtenue en relevant les lettres au-dessous des quelles se trouve un petit trou ; dans le second cas, deux tailles de caractères sont utilisées, le message étant constitué des lettres soit de petites tailles, soit de grandes tailles selon la convention adoptée pour l'échange [Bar 07].

### 1.3.1.3.2 La stéganographie technique

La stéganographie technique regroupe toutes les techniques qui ne jouent pas sur les mots. Les premières utilisations de la stéganographie technique sont racontées par *Hérodote* dans son œuvre *Histoires* [Bar 07]. Pour organiser une révolte contre les perses on utilisait un esclave fidèle pour transmettre un message, on lui rasé la tête pour y tatouer, une fois les cheveux repoussés, le message est alors invisible. Le principale désavantage de cette méthode était l'attente pour l'envoi d'un message.

En chine ancienne, les messages étaient écrits sur de la soie, qui aient ensuite roulé en boule, elle-même recouverte de cire.

Une autre méthode qui consiste à réduire un texte ou une image en un point d'un millimètre ou moins. Celui-ci est ensuite disposé dans un texte ou une image normale. Ce procédé a été utilisé dans les billets de banque suisses<sup>1</sup>.

Dans les années 80 Margaret Thatcher réussit à identifier plusieurs fuites de documents en traçant ceux-ci à l'aide de techniques de dissimulation d'information.

Dernièrement, de nombreux spécialistes ont avancés l'hypothèse selon laquelle les terroristes auraient coordonné les attentats du 11 septembre 2001 en utilisant des messages cachés dans des images [Bar 07].

---

<sup>1</sup>La stéganographie dans les billets de banque

### 1.3.1.4 TYPES DES SUPPORTS

Aujourd'hui, les messages cachés se transmettent de manière numérique et non plus par des techniques manuelles. Il est possible de dissimuler un message dans un support numérique qui peut être un fichier texte, une image, un son, une vidéo, protocole réseaux, un système de fichier ou un code source.

#### 1.3.1.4.1 Dissimuler de l'information dans du texte

De nombreuses solutions ont été proposées pour dissimuler un message dans un texte, voici quelques techniques [Ray 02]:

- **Ajout d'un message à la fin des phrases :**

Le principe de cette méthode consiste à mettre des espaces en fin de phrase, on définit le code à suivre et on commence : Le bit 0 correspond à 0 espaces et le bit 1 correspond à un seul espace en fin de la phrase.

Ex: (ici les espaces sont remplacés par des "\_" pour plus de lisibilité).

Bonjours ceci est un message caché. A vous de lire\_. Je pense que vous commencez à comprendre le principe\_.

Il est facile de remarquer qu'il faut énormément de lignes pour coder peu de texte, en effet, il faut huit lignes pour cacher un seul octet.

- **Espacement entre les mots :**

Une autre possibilité pour dissimuler un message dans un texte basée sur l'espacement entre les lettres, voir des mots, dans deux versions d'un même texte. Le secret apparaît ensuite lorsque les deux textes sont superposés.

Ainsi, l'œil ne distingue aucune différence entre ces deux phrases

A l'envers, cette figure nuit à la compréhension.

A l'endroit, son exposition semble trop convenue.

Et celle-ci

A l'envers, cette figure nuit à la compréhension.

A l'endroit, son exposition semble trop convenue.

La superposition des deux met en relief une phrase :

A l'envers, **cette** figure **nuit à** la compréhension.

A l'**endroit**, son exposition semble trop **convenue**.

#### 1.3.1.4.2 Dissimuler de l'information dans les images

Il est possible de présenter les techniques de stéganographie selon différentes classifications. Frédéric Raynal [Ray 02] le fait d'après les modifications induites sur le support. Il distingue six catégories :

- Un système par substitutions remplace les parties redondantes du support par le message.
- Les techniques par transformations dissimulent l'information dans une transformée du support, comme par exemple, la transformée à cosinus discrète ou le domaine des ondelettes.
- Les techniques par étalement de spectre : les informations sont dissimulées dans toute l'image, et la perte de certaines informations doit pouvoir être compensée par les autres.
- Les méthodes statistiques modifient plusieurs statistiques du support (distribution des pixels, luminosité...) pour cacher le message et le récupèrent en testant ces hypothèses.
- Les techniques par distorsions altèrent le support, la différence avec le support initial constituant alors le message.
- les méthodes par génération de support construisent un support autour du message pour le dissimuler.

#### 1.3.1.4.3 Dissimuler de l'information dans du son

De faibles variations, imperceptible pour l'oreille, dans les basses fréquences ou ce que l'on appelle le bruit de fond peuvent contenir une grande quantité d'information. Un grésillement infime peut cacher des secrets.

Evidemment, ce bruit doit de préférence être transmis de façon numérique sans quoi les vraies pertes de transmission pourraient effacer entièrement le message caché [Lo 08].

#### 1.3.1.4.4 Dissimuler de l'information dans la vidéo

Les techniques sont équivalentes à celles utilisées dans les images à la différence près que les vidéos sont souvent plus bruitées ce qui facilite l'imperceptibilité des données dissimulées mais les rend aussi moins robustes [Lo 08].

#### 1.3.1.4.5 Autres supports

##### ▪ Système de fichiers

Pour stocker un fichier, le système découpe ce dernier en un nombre de morceaux tel que chaque morceau puisse être logé dans un bloc, comme la taille d'un fichier a rarement une taille multiple de la taille des blocs, le dernier bloc est généralement n'est pas remplie.

Le système de fichier laisse la possibilité d'utiliser des techniques stéganographiques, pour caché des données, il suffit de les stockées dans ce dernier bloc, si la taille de ces données

dépassent l'espace du bloc non rempli, il faut les découper et les stockées sur autant de bloc nécessaire, garder la trace des blocs utilisés et l'ordre pour la récupération.

Le problème de cette technique vient du fait que les fichiers peuvent être modifiés, supprimés déplacés, etc.

Il existe plusieurs outils dans le web de bas niveau comme **bmap** et **slacker**<sup>2</sup>, permettant d'analysés en détails les blocs utilisés et récupérer l'espace libre [Web 1.2].

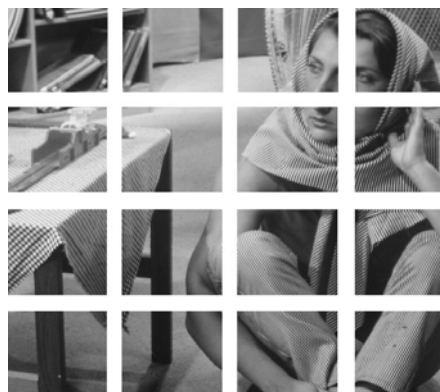
- **Fichier exécutable**

Les fichiers exécutables peuvent être utilisés pour transmettre un message d'une manière secrète. Lors de la compilation d'un programme, le code source est transformé en un ensemble d'instructions compréhensible par la machine, pour exécuter ce dernier, le système d'exploitation lit les sections dont il a besoin, donc il est possible d'exploiter les parties du code non exécuté.

L'inconvénient de cette technique est qu'un simple affichage de code assembleur modifié indiquera la présence d'une information qui a priori n'a rien à y faire [Gal 04].

### 1.3.2 LE TATOUAGE (WATREMARKING)

Le tatouage numérique est une technique qui consiste à insérer dans un support numérique, une information (marque) propre à son ayant le droit qui ne pourra être enlevée sans altération de ce support (rendant alors son utilisation impossible). Cette marque devra rester visible ou invisible et faire suffisamment corps avec le support pour rester présente malgré les manipulations que ce dernier peut subir. Par exemple, les transformations pouvant altérer la marque dans une image sont des rotations, des changements d'échelle, des symétries, des découpages, des changements de format, l'ajout de bruit, attaque par mosaïque (figure 1.2), etc.



**Figure 1.2. Exemple d'une attaque mosaïque**

---

<sup>2</sup>Commandes pour la manipulation de l'espace libre d'un fichier et d'un répertoire



### 1.3.2.1 APPLICATIONS DU TATOUAGE

- **Protection des droits d'auteur**

La protection des droits d'auteur a été une des premières applications étudiée en tatouage d'image. Ce service reste cependant toujours d'actualité et concerne encore la majorité des publications. L'objectif est d'offrir, en cas de litige, la possibilité à l'auteur ou au propriétaire d'une image d'apporter la preuve qu'il est effectivement ce qu'il prétend être, et ce même si l'image concernée a subi des dégradations par rapport à l'original [Rey 03].

- **Vérification de l'intégrité du contenu d'une image**

L'idée de base consiste à utiliser les techniques de tatouage d'image afin de cacher dans certaines zones de l'image des informations sur d'autres zones. Ces informations servent à alerter l'utilisateur face à une éventuelle modification ou découpe de l'image par une personne non autorisée et à localiser précisément les régions manipulées (figure 1.3), voire éventuellement à les restaurer [Rey 03].



Image originale

image attaquée

**Figure 1.3. Détection des régions manipulées**

- **Autres services**

Il existe d'autres applications possibles en dehors de celles décrites précédemment, et dans des domaines autres que des services de sécurité. On peut imaginer utiliser une technique de tatouage d'image pour faciliter la recherche dans une base de données multimédia en cachant par exemple dans le document des informations textuelles sur son contenu.

### 1.3.2.2 TYPES DE TATOUAGE

Plusieurs formes et degrés de tatouages existent [Hen 98]. Ils sont généralement répertoriés par leurs degrés de priorités : robuste ou fragile et visibles ou non visibles.

- **Tatouage robuste** : il s'agit ici de pouvoir récupérer la marque même si l'image marquée a été manipulée. Il est nécessaire de distinguer plusieurs types d'attaques selon qu'elles sont considérées comme étant biens ou malveillantes, destructives ou non. Les attaques bienveillantes regroupent les manipulations effectuées par un utilisateur de bonne foi. On trouve dans cette catégorie la compression JPEG, les conversions de format en général, les changements de résolution (zoom), etc.
- **Tatouage fragile** : le tatouage fragile présente pratiquement un intérêt pour assurer un service d'intégrité de document [Yeu 97]. L'idée de ce service n'est pas de prouver que oui ou non un document est original ; mais plutôt qu'un document est non falsifié.
- **Tatouage visible**: Le principe fondamental du tatouage visible consiste à masquer partiellement un support numérique à l'aide d'une ou plusieurs marques visibles (figure 1.4), qui ne peuvent être correctement effacées que si l'on possède une clé secrète adéquate.



Figure 1.4. Exemple d'un tatouage visible

- **Tatouage invisible** : Le tatouage numérique invisible peut être considéré comme une forme de stéganographie, puisque l'utilisateur final ignore la présence du tatouage et donc de l'information cachée (figure 1.5).



Figure 1.5. Exemple d'un tatouage invisible

### 1.3.3 LE FILIGRANE (FINGERPRINTING)

Contrairement aux données de nature analogique pour lesquelles une succession de reproductions entraîne rapidement une perte significative de la qualité, les données numériques peuvent être dupliquées quasiment à l'infini. Dans ce contexte, une personne ayant accès à ce type de données et au matériel adéquat, est potentiellement capable de les reproduire bit à bit à l'identique.

L'objectif du filigrane est la détection des copies illégales d'un support marqué. Chaque utilisateur authentifié reçoit sa propre copie du medium avec une empreinte pour l'identifier. Ainsi lorsqu'une copie illégale est découverte, la lecture de l'empreinte indique la source de la fuite [Ray 02].

### 1.4 PROCESSUS DE DISSIMULATION

La mise en œuvre d'un schéma de dissimulation d'information s'effectue en deux étapes distinctes [Bar 07] :

#### 1.4.1 Insertion

Cette étape comprend les opérations suivantes :

- La compression du message à insérer et le chiffrer avec une clé cryptographique.
- La sélection d'un support de couverture.
- La sélection des sous parties du support favorable à la dissimulation à l'aide de l'algorithme de la dissimulation.
- La dissimulation aléatoire, à l'aide de la clé stéganographique/tatouage, le message chiffré dans les parties favorables.

Cette étape est illustrée dans la figure 1.6, appelée aussi *étape de dissimulation*.

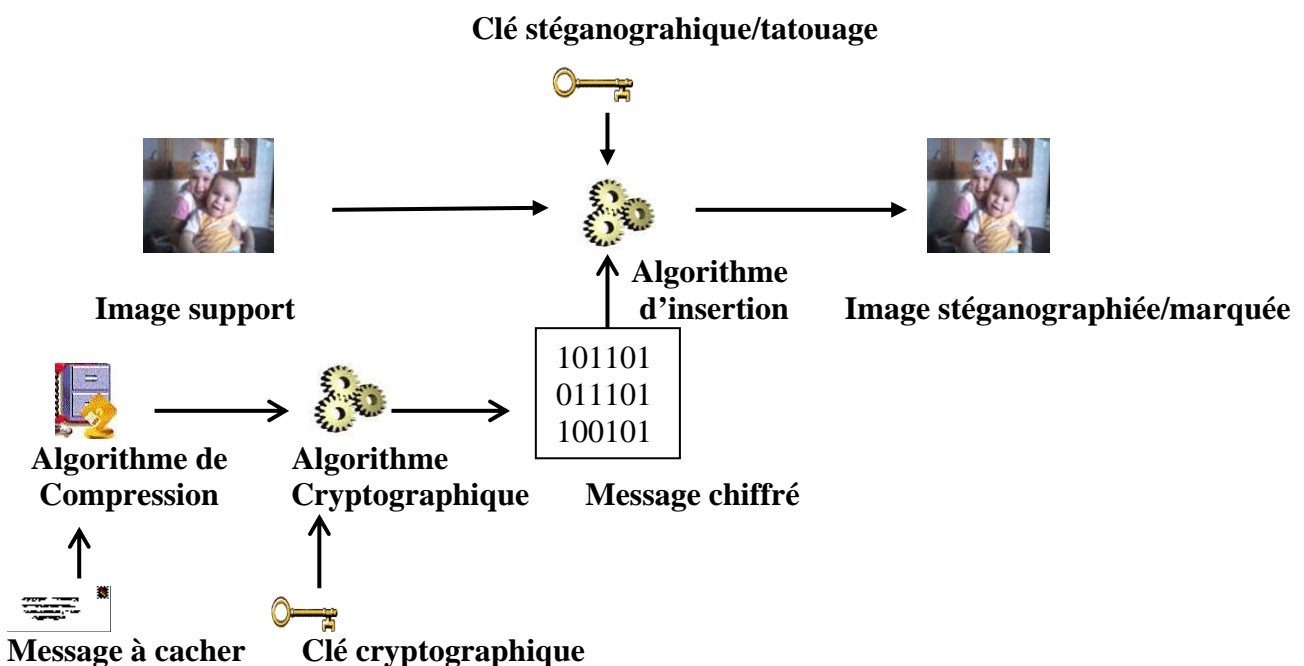


Figure 1.6. Etape de dissimulation pour une image fixe

### 1.4.2 Extraction

Cette étape comprend les opérations suivantes :

- La sélection des sous-parties du support favorable à la dissimulation à l'aide de l'algorithme de la dissimulation.
- Retrouver les positions du message chiffré dans les parties favorable, à l'aide de la clé stéganographique/tatouage.
- Déchiffrer le message à l'aide de la clé cryptographique et le décompresser.

Cette étape est illustrée dans la figure 1.7, appelée aussi étape d'*insertion*.

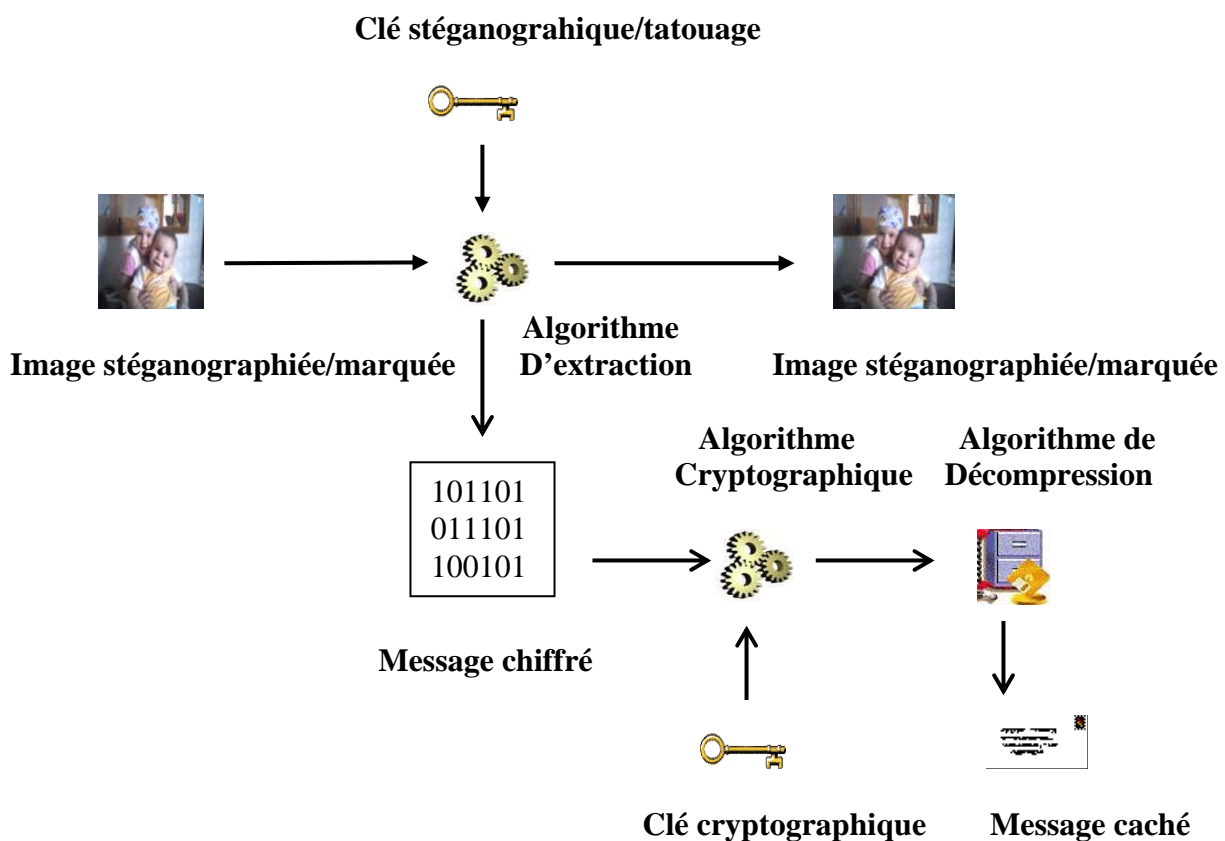


Figure 1.7. Etape d'extraction pour une image fixe

## 1.5 CRITERES DE DISSIMULATION D'INFORMATION

Les applications de dissimulation d'information sont triées en fonction de trois critères [Lu 05]:

### 1.5.1 La capacité

C'est la quantité d'information que l'on désire cachée par rapport à la quantité d'information associée au support image audio, vidéo. Dans le cas du tatouage la capacité se limite souvent de 16

à 64 bits pour assurer un service de droit d'auteurs à l'aide d'un identifiant, mais pas pour cacher des informations explicites comme un logo de société, assurer des services d'intégrité.

### 1.5.2 L'imperceptibilité

Appelé aussi invisibilité, le but est de faire en sorte que le stego-medium reste fidèle au medium original.

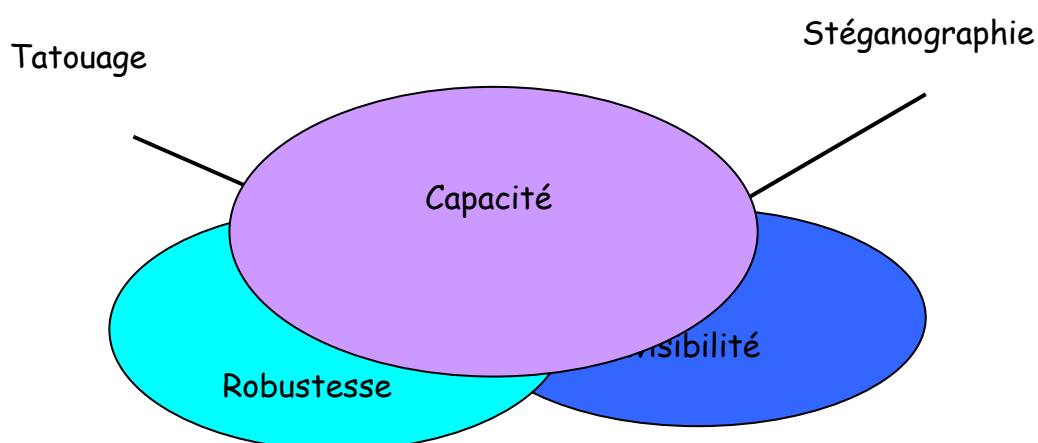
Les données ne doivent pas être «perceptibles» dans le stégo-médium. Pour le tatouage, l'objectif est de ne pas détériorer le stégo-médium protégé. Cependant, la contrainte est plus forte en stéganographie où il s'agit plutôt d'une indétectabilité statistique.

### 1.5.3 La robustesse

Le but de cette propriété est de récupérer les données cachées même si le stego-medium a été manipulé. On peut définir la robustesse par la résistance du marquage face à des manipulations du stégo-medium. Dans le cas où le support est une image, les manipulations peuvent être de type géométrique (rotation, zoom, découpage,...), elles peuvent modifier certaines caractéristiques du support numérique (histogramme des couleurs, saturation,...). Il peut aussi s'agir de tous les types de dégradations fréquentielles de l'image (compression avec pertes).

Il est facile de remarquer que ces trois critères sont contradictoires, si par exemple on augmente la taille de l'information à dissimulé dans ce cas le stégo-medium risque d'être détecté, de la même manière si le but est de rendre le message (marque) plus robuste, cela aura en contrepartie pour rendre ce dernier plus visible.

Donc il est nécessaire de trouver un compromis entre l'imperceptibilité, la capacité et la robustesse. Ce compromis est généralement représenté par la figure 1.8.



**Figure 1.8. : Compromis entre Imperceptibilité, Capacité et la Robustesse**

En stéganographie, le compromis qui nous intéresse est celui entre la capacité et l'invisibilité, l'objectif de la stéganographie et d'envoyer le maximum d'information sans qu'un attaquant ne

puisse le détecter, il s'agit d'une indétectabilité statistique. Quant à la robustesse des mesures doivent être prises lorsque l'attaquant est actif.

La notion de robustesse est plutôt importante pour le tatouage robuste, tous les utilisateurs savent ou soupçonnent très fortement, qu'une marque est dissimulée dans le stégo-médium, et il n'est donc nul besoin de chercher à en détecter la présence. Pour le tatouage fragile, ce dernier ne demande que peu de robustesse afin de détecter qu'un médium, ou une partie de celui-ci, a été modifié.

Enfin, les besoins de filigrane sont à peu près identiques à ceux du tatouage pour l'imperceptibilité et la robustesse. La capacité est importante car un médium doit contenir une marque spécifique à un utilisateur.

### 1.6 COMPARAISON ENTRE LES TECHNIQUES DE LA DISSIMULATION D'INFORMATION

Malgré les objectifs distincts, il est facile de remarquer, que ces trois approches requièrent des paramètres communs [Ray 02] :

- Chaque approche nécessite des informations que se soit un message, une marque ou une empreinte.
- Un support pour dissimuler ces informations, son importance dépend de l'application, aucune pour la stéganographie, capital pour le tatouage et le filigrane.
- Utilisation d'une clé pour insérer ou extraire/détecter l'information, la fonction extraction/détection dépend de l'objectif de l'application, extraction en stéganographie et détection pour les deux autres approches.

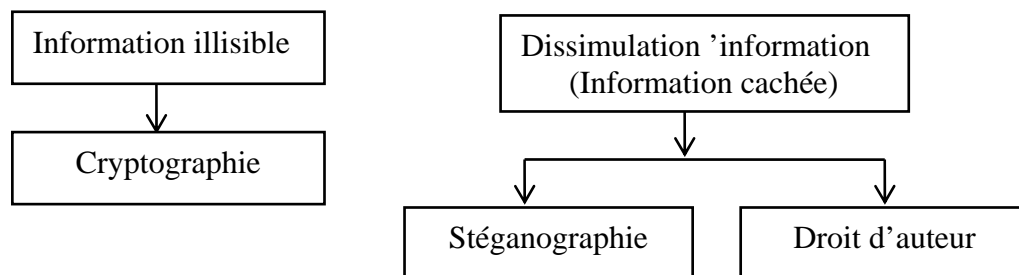
	Stéganographie	Tatouage	Filigrane
<b>Similarités</b>	Algorithmes D'insertion / Extraction		
	Médium de couverture		
<b>Différences</b>	Le message n'a aucune importance avec le médium De couverture	Un lien fort entre le message et le médium de couverture	
	Une clé pour insérer / récupérer	Une clé pour insérer / détecter	
	Objectif : cacher l'existence du message	Existence du message peut être connue	
	Attaque : le but est la détection	le but est la destruction	

**Tab.1.1. : Schéma générale en dissimulation.**

Le tableau 1.1 montre qu'une clé est nécessaire pour insérer les données dans le medium de couverture et les extraire ou en détecter la présence dans le stégo-médium.

## 1.7 STEGANOGRAPHIE VS CRYPTOGRAPHIE

La stéganographie est un art proche de la cryptographie [Den 05]. C'est pourquoi il est essentiel de les distinguer (figure 1.9).



**Figure 1.9 : diagramme représentant la dissimulation d'information vs cryptographie [Den 05]**

La dissimulation d'information cherche à dissimuler la présence d'informations pertinentes au sein de plusieurs autres; le message secret est caché dans un support de manière à passer inaperçu lors de la communication. Quant à la cryptographie l'objectif est de rendre l'information incompréhensible à une personne ne possédant pas les connaissances adéquates : une personne surveillant le canal de communication par lequel transite le message sait qu'un échange a lieu, mais est ainsi incapable d'en interpréter le contenu.

### 1.7.1 NOTIONS DE LA CRYPTOGRAPHIE

La cryptographie désigne l'ensemble des techniques permettant l'envoi d'un message codé de telle sorte que seul le destinataire puisse le décoder.

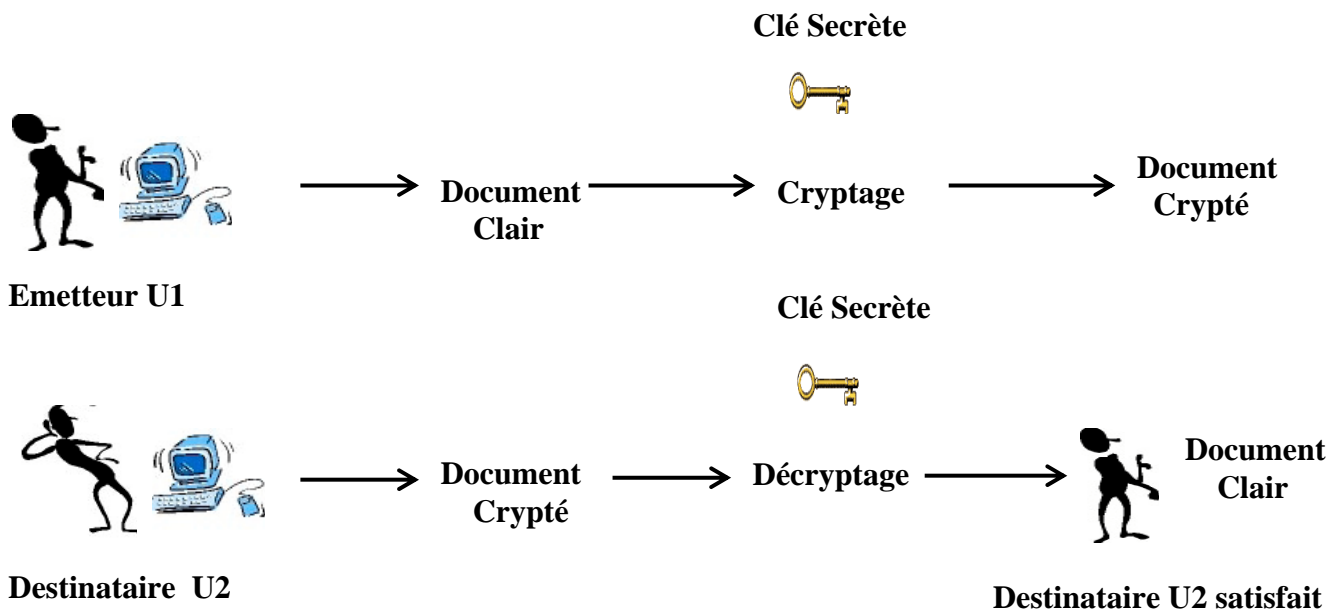
### 1.7.2 TYPES DE CHIFFREMENT

On distingue deux types d'algorithmes de chiffrement: celle à clé *secrète* et celle à clé *publique*.

#### 1.7.2.1 Algorithmes de cryptographie symétrique (à clé secrète)

Ce système est basé sur une même clé de cryptage /décryptage des informations. Cette clé est donc connue et détenue par les parties communicantes.

Les algorithmes de chiffrement symétrique les plus connus sont **DES**, **3DES**, **AES**, **RC4**, **RC5** [Web 1.3].



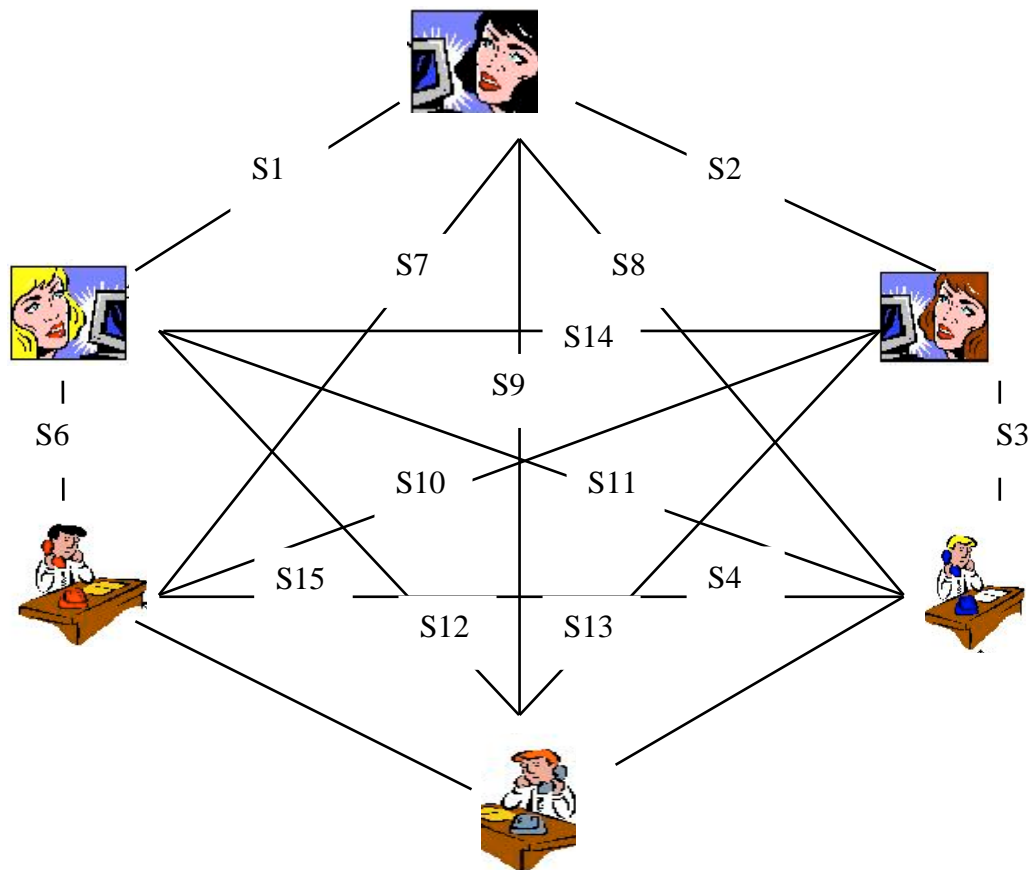
**Figure 1.10 : Principe de chiffrement asymétrique [Clu 02]**

La figure 1.10 montre que :

1. Un utilisateur U1 désire envoyer un document à U2.
2. U1 code avec la clé symétrique connue des deux parties et transmet le document crypté.
3. U2 reçoit le document crypté.
4. Grâce au déchiffrement U2 obtient le document initial en clair.

La technique de cryptographie à clé secrète exige une clé par paire de clés et chaque partie en possèdera deux [Clu 02]. Pour quatre participants, le nombre de clés s'on passe à trois, il faudra donc trois clés et chaque partie en possèdera deux. Pour quatre participants, le nombre de clés s'élève à 6 et chacun en possède trois. Ainsi pour n participants, il faudra distribuer  $((n*(n-1))/2)$  clés (figure1.11).





**Figure 1.11 Complexité de la distribution de clés symétriques [Clu 02]**

Pour palier à cela, en 1976 Whitfield Diffie et Martin Hellman, deux mathématiciens, ont mis au point un système asymétrique, faisant intervenir les notions de clé publique et clé privée.

### 1.7.2.2 Algorithmes de cryptographie asymétrique (à clé publique et privée)

Cette technique ne repose plus sur le partage d'une même clef secrète, mais sur la possession d'une clé privée par l'utilisateur et une clé associée, dite clé publique. Chaque personne possède sa propre paire de clés (privée, publique), la première reste secrète, alors que la seconde est disponible pour tout le monde.

Ainsi, pour envoyer un message à un utilisateur U1, n'importe qui pourra utiliser la clé publique de U1. Seul U1 pourra déchiffrer le message, avec l'aide de sa clé privée (figure 1.12).

Réciproquement, un document codé avec la clé privée de U1 ne pourra être déchiffré qu'avec la clé publique de ce même utilisateur.

Les algorithmes de chiffrement asymétrique les plus connus sont : **RSA** et **DSA** [Web 1.3].

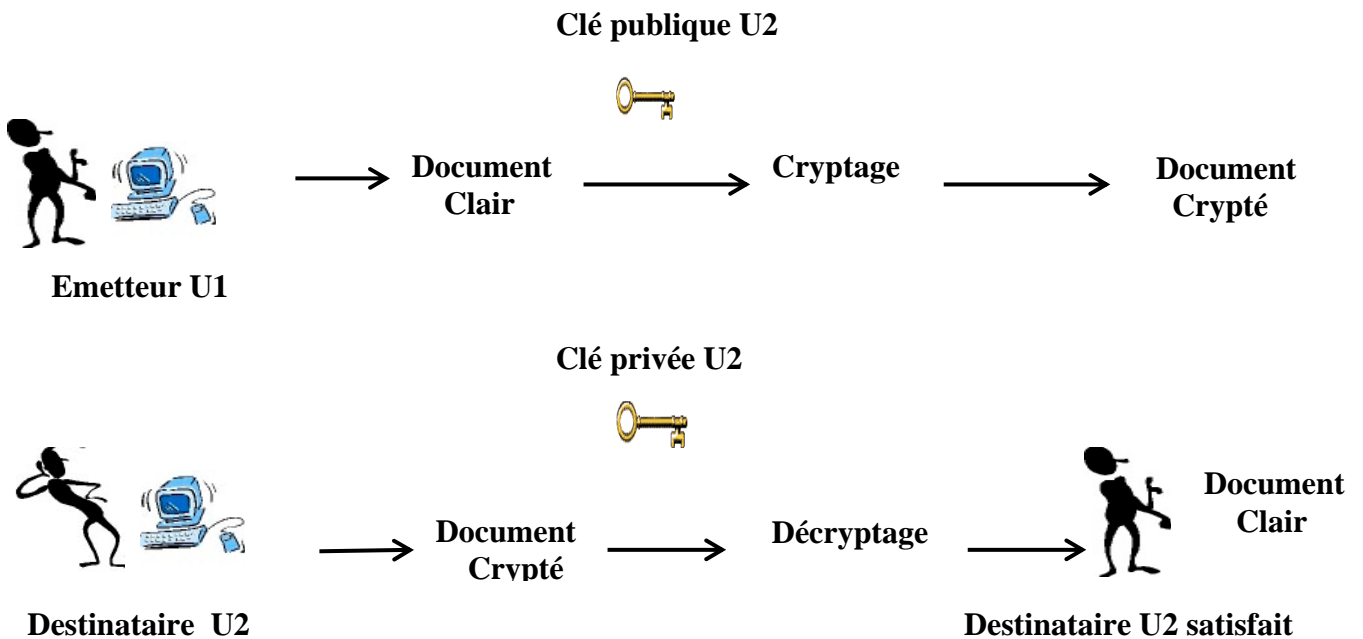


Figure 1.12 : Mécanisme asymétrique [Clu 02]

La dissimulation d'information et la cryptographie sont souvent très proches mais n'utilisent pas les mêmes outils pour protéger l'information. On peut dire que ces deux disciplines sont complémentaires, dans le cas de la dissimulation d'information, la communication n'est pas chiffrée mais le message insérer dans le support numérique peut être chiffré avec un des deux types d'algorithmes de cryptographie. Par exemple le logiciel de stéganographie *Invisible secret 4* [Web 1.4] utilise les algorithmes de cryptographie **RC4**, **Cast128**, et **Gost**.

## 1.8 CONCLUSION

La dissimulation d'information et la cryptographie peuvent jouer le même rôle, à savoir protéger l'information, ce sont bien deux sciences différentes.

La cryptographie est un moyen puissant pour protéger l'information mais celle-ci possède un léger inconvénient. En effet, un texte codé ou une image totalement brouillée attire l'attention des voleurs d'information.

La dissimulation d'information se subdivise en trois catégories en fonctions des objectifs recherchés. La stéganographie cherche à dissimuler un message secret dans support charger du transport qu'une personne qui surveille la communication ne le remarque pas.

Le tatouage consiste à cacher un message ayant un rapport direct avec le support dont le but est de protéger ce dernier contre la copie, contre toute modification, ou dans un but d'identification.

Le filigrane consiste à insérer une empreinte – équivalente d'un numéro de série – pour limiter détecter les copies illégales de ce support.

Comme l'objectif principal de ce travail est la détection passive des informations cachées dans les images fixes, le chapitre suivant décrit, dans un premier temps, les concepts de base sur les images fixes. Dans un deuxième temps on essaiera de répondre à la question suivant : comment utiliser la stéganographie dans ce support ?

**CHAPITRE 2**  
**La Stéganographie adaptée**  
**aux images fixes**

## CHAPITRE 2

### LA STEGANOGRAPHIE ADAPTEE AUX IMAGES FIXES

#### 2.1 INTRODUCTION

Le mot stéganographie puise ses racines du grec, où ‘*stéganos*’ qui veut dire ‘*cache*’ et ‘*graphie*’ ‘*écriture*’. Littéralement, on peut traduire le mot stéganographie par l’expression ‘*écriture cachée*’. Son objectif est de pouvoir dissimuler des données qui doivent être tenues secrètes dans un support paraissant anodin.

La stéganographie peut s’utiliser sur plusieurs supports (image, vidéo, son, texte, protocole réseau), ce dernier doit supporter les différentes modifications par la fonction d’insertion dans le processus de dissimulation : *reste fidèle au support original*, avoir une capacité importante pour la dissimulation. De plus, le stégo-médium doit résister aux différentes manipulations : *permet de récupérer le message* même si le support a subi des manipulations.

L’image est l’un des supports les plus usités par les algorithmes de stéganographie<sup>3</sup> [Bar 07]. La raison de l’importance de ce type de support dans le domaine de stéganographie réside dans les différents types d’images numériques, les domaines de représentations, le codage des couleurs, les différents formats des images et la réduction de la taille de l’information (compression Jpeg, compression par ondelette,...).

Ce chapitre se divise en deux parties. Dans la première partie nous présentons les concepts de base des images fixe, comment elles sont faites, à quoi elles servent et comment sont elles définies. Cela va nous permettre, dans la deuxième partie, de cerner l’utilisation de la stéganographie dans ce type de support.

---

3 Logiciel de stéganographie 2004-2006.

## Première partie

### 2.2 LES IMAGES NUMERIQUES

Pour étudier les techniques de la stéganographie appliquée aux images numériques, il est nécessaire de définir les concepts de base sur lesquels ces techniques s'appuient.

#### 2.2.1 TYPES D'IMAGES

Les images produites et traitées par les ordinateurs sont de deux types : les images *bitmap* et les images *vectérielles* [Web 2.1].

##### 2.2.1.1 Les images de type Bitmap

Les images Bitmap appelées aussi image matricielles : il s'agit d'images pixellisées, c'est-à-dire un ensemble de points (pixels) contenus dans un tableau, chacun de ces points possède une ou plusieurs valeurs décrivant sa couleur.

Le bitmap est aussi un format de fichier. Le mot désigne deux choses différentes, ici on parle du type d'image et non pas du format (.bmp). Les extensions de fichiers bitmap que l'on trouve le plus couramment sont : \*.psd, \*.bmp, \*.tif, et pour l'Internet \*.jpg, \*.gif et \*.png.

##### 2.2.1.2 Les images de type vectoriel

La description vectorielle d'une image consiste en une description géométrique: le document numérisé prend donc la forme d'une suite de formules mathématiques décrivant les formes élémentaires constituant l'image (carrés, rectangles, ellipses, cercles, courbes, etc.). Chaque forme élémentaire constitue un objet et se voit assigné un certain nombre d'attributs tels que la couleur, la transparence, l'épaisseur du trait, le type de trait (pointillé, etc.). Les extensions de fichiers que l'on trouve le plus couramment sont : \*.ai, \*.cdr, \*.fh.

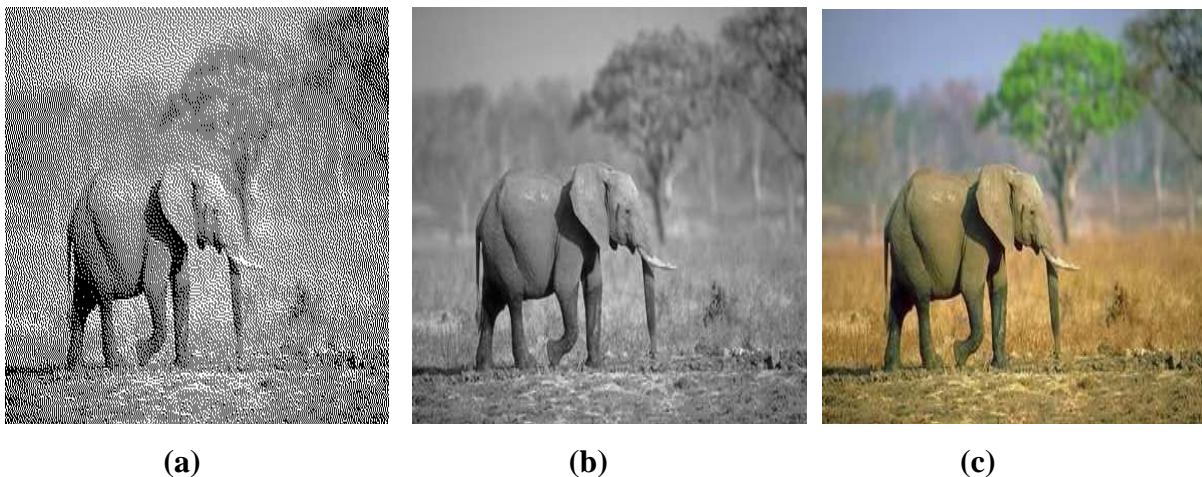
#### 2.2.2 LE CODAGE D'UNE IMAGE NUMERIQUE

C'est la partie qui nous intéressera le plus car pour pouvoir stéganographier une image, on doit savoir comment elle est codée.

Une image est constituée d'un ensemble de points nommés pixels représentant des cases qui forment un tableau à deux dimensions. Chaque case est codée avec un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle aussi **profondeur de codage** (parfois *profondeur de couleur*). Il existe plusieurs standards de codage de la profondeur [Web 2.2]:

- **Bitmap noir et blanc:** en stockant un bit dans chaque case, il est possible de définir deux couleurs (noir ou blanc) (figure 2.1 (a)).

- **Bitmap 16 couleurs ou 16 niveaux de gris:** en stockant 4 bits dans chaque case, il est possible de définir  $2^4$  possibilités d'intensités pour chaque pixel, c'est-à-dire 16 dégradés de gris allant du noir au blanc ou bien 16 couleurs différentes.
- **Bitmap 256 couleurs ou 256 niveaux de gris:** en stockant un octet dans chaque case, il est possible de définir  $2^8$  intensités de pixels, c'est-à-dire 256 dégradés de gris allant du noir au blanc ou bien 256 couleurs différentes (figure 2.1 (b)).
- **Palette de couleurs (Colormap):** il est possible de définir une palette, ou table des couleurs (LUT : Look Up Table), contenant l'ensemble des couleurs pouvant être contenues dans l'image, à chacune desquelles est associé un indice. En codant les indices sur 8 bits, il est possible de définir 256 couleurs utilisables, c'est-à-dire que chaque case du tableau à deux dimensions représentant l'image va contenir un nombre indiquant l'indice de la couleur à utiliser. On appelle ainsi **image en couleurs indexées** une image dont les couleurs sont codées selon cette technique.
- « **Couleurs vraies** » (*True color*) ou « *couleurs réelles* » : cette représentation permet de représenter une image en définissant chacune des composantes (RGB, pour rouge, vert et bleu). Chaque pixel est représenté par un entier comportant les trois composantes, chacune codée sur un octet, c'est-à-dire au total 24 bits (16 millions de couleurs)(figure 2.1 (c)). Il est possible de rajouter une quatrième composante permettant d'ajouter une information de transparence ou de texture, chaque pixel est alors codé sur 32 bits.



**Figure 2.1 : même image en noir et blanc (a), niveaux de gris (b) et en couleurs(c)**

### 2.2.3 FORMATS D'IMAGES

On appelle format d'image la façon dont celle-ci est structurée. Ainsi une image a généralement une zone d'entête contenant les données permettant de la reconnaître (taille d'image, dimension de l'image, type de codage, méthode de compression, palette lorsqu'elle est définie, etc.) et de lire la partie suivante qui est celle des données.

Le diagramme suivant (figure 2.2) représente les formats les plus répandus. Nous nous concentrerons essentiellement sur les trois formats les plus utilisés, à savoir le BMP, le GIF et le JPEG [Den 05].

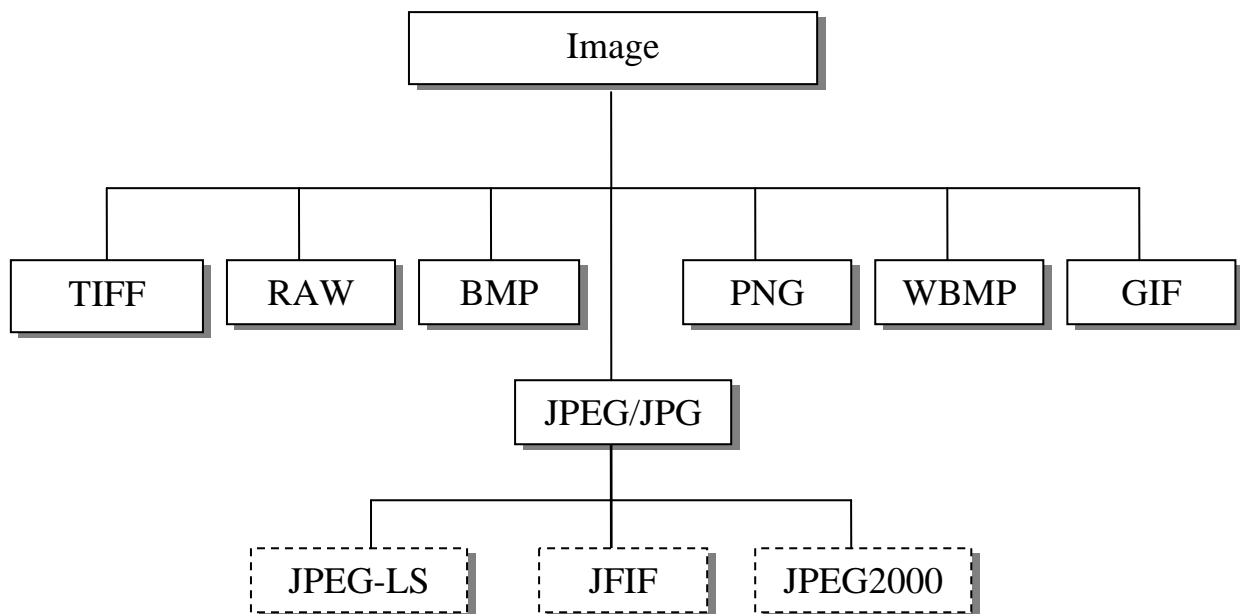


Figure 2.2 Diagramme représentant quelques formats d'images [Den 05]

#### 2.2.3.1 BMP (Windows Bitmap)

Le format BMP est un des formats les plus simples, développé conjointement par Microsoft et IBM. Un fichier BMP est un fichier bitmap, c'est-à-dire un fichier d'image graphique stockant les pixels sous forme de tableau de points et gérant les couleurs soit en couleur vraie soit grâce à une palette indexée.

Si on ouvre un fichier .Bmp avec un éditeur hexadécimal (par exemple, **Hex Editor, v 3.12**), on trouve un ensemble d'octets (figure 2.3), qui peuvent être séparés en trois parties successives [Web 2.3]:



42	4D	86	00	00	00	00	00
00	00	36	00	00	00	28	00
00	00	05	00	00	00	05	00
00	00	01	00	18	00	00	00
00	00	50	00	00	00	C4	0E
00	00	C4	0E	00	00	00	00
00	00	00	00	00	00	00	00
FF	00	00	FF	00	00	FF	00
00	FF	00	00	FF	00	00	00
FF	00	00	FF	00	00	FF	00
00	FF	00	00	FF	00	00	00
FF	00	00	FF	00	00	FF	00
00	FF	00	00	FF	00	00	00
FF	00	00	FF	00	00	FF	00
00	FF	00	00	FF	00	00	00
FF	00	00	FF	00	00	FF	00
00	FF	00	00	FF	00	00	00

**Figure 2.3 : Fragment d'un fichier image.bmp en hexadécimal [Web 2.4]:**

- **Une partie en-tête de fichier (indiquée en rouge) :** l'en tête de fichier, comprend 14 octets. Elle contient de l'information sur le type et sur la taille du fichier :  
 42 4D : type de fichier (BM).  
 86 00 00 00 : taille du fichier.  
 00 00 : deux octets réservés (Zéro).  
 00 00 : deux octets réservés (Zéro).  
 36 00 00 00 : décalage des bits de pixels.
- **Une partie en-tête d'information sur l'image (indiquée en bleu) :** La seconde partie, l'en tête d'information sur l'image, fait 40 octets. Elle contient des informations sur les dimensions (largueur, hauteur), nombre de plans de bit, nombre de bits par pixels et le format de couleur de l'image
- **Une partie de description de chaque pixel (indiquée en vert) :** La troisième partie est constituée de la description des couleurs des pixels constituant l'image. Dans cette partie, les pixels sont décrits dans l'ordre inverse de leur affichage à l'écran. Les pixels associés à la dernière ligne apparaissent en premier et les pixels associés à la première ligne apparaissent en dernier dans le fichier.

### 2.2.3.2 GIF (Graphics Interchange Format)

Le GIF est un format de bitmap compressé utilisant 8 bits par pixel. Le format GIF est l'un des formats les plus utilisés sur le Web (PNG, JPEG). Comme les images GIF n'utilisent que 8 bits par pixel, elles ne peuvent exprimer que 256 couleurs ( $2^8$  couleurs) et ne constituent donc pas le meilleur choix pour les photographies. La compression employée dans les images GIF utilise une détection des zones répétitives de l'image, telles que les zones ayant la même couleur.

La méthode de compression est dite sans perte, ce que signifie qu'aucune information n'est perdue, mais elle n'est pas très efficace pour compresser des images photographiques.

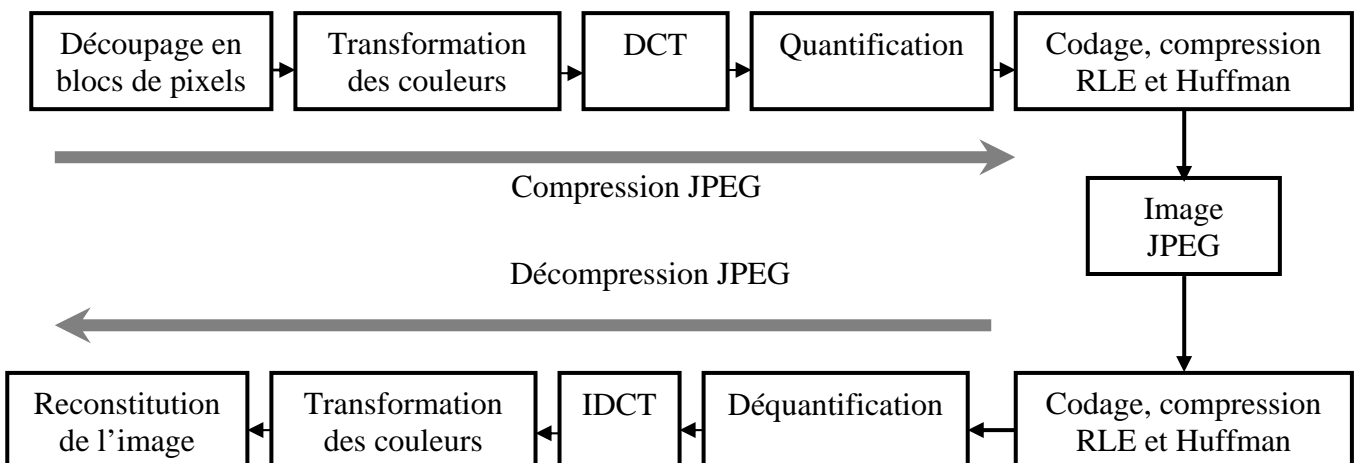
### 2.2.3.3 JPEG (Joint Photographic Experts Group)

Le format JPEG permet une compression importante des images en 24 bits ou 32 bits. Les images JPEG peuvent être compressées jusqu'à une faible fraction de leur taille originale grâce à une technique de compression avec perte (certaines informations contenues dans l'image sont ignorées). Cette compression est possible parce que l'oeil humain n'est pas très sensible aux très légers changements de couleurs. Le niveau de compression peut être précisé lorsqu'on enregistre l'image JPEG. Plus la compression est importante, plus il y a d'informations écartées afin de réduire la taille du fichier. Lorsque la compression est très élevée, la perte des détails peut devenir visible. Les fichiers images JPEG utilisent des extensions .jpg ou .jpeg .

### 2.2.4 COMPRESSION JPEG

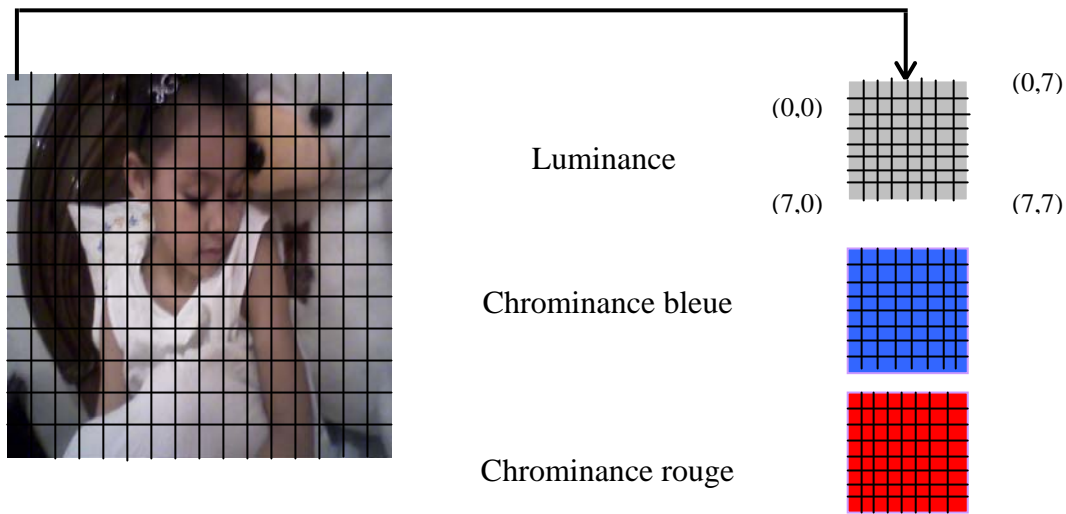
Elle consiste à réduire la taille de l'information pour permettre le stockage de cette information dans un fichier et son transport. La compression se définit par le quotient de compression, c'est-à-dire le quotient du nombre de bits dans l'image compressée par le nombre de bits dans l'image originale. La compression peut être avec ou sans perte.

On peut diviser la compression et la décompression JPEG en cinq étapes [Gui 98], (Figure 2.4) :



**Figure 2.4 : Schéma de compression/décompression JPEG**

- **Découpage de l'image en blocs** : le format JPEG commence par découper l'image en blocs ou carreaux généralement carrés de 64 (8\*8) ou 256 (16\*16) pixels (figure2.5).



**Figure 2.5 : Découpage d'une image en blocs de  $8 \times 8$  valeurs**

- **Transformation des couleurs (optionnelle)**: La deuxième étape de la compression JPEG consiste en un changement de l'espace des couleurs de RVB vers l'espace de couleurs YCbCr.

Un pixel sera donc codé par un triplet (Y, Cb, Cr), où Y désigne la luminance, c'est-à-dire l'intensité lumineuse, Cb la chrominance bleue, c'est-à-dire l'intensité de la couleur bleue et Cr la chrominance rouge, c'est-à-dire l'intensité de la couleur rouge (figure 2.6).

Le changement d'espace de couleurs se justifie par le fait que l'espace YCbCr est très proche du fonctionnement de l'oeil humain. De plus, ce dernier est très sensible aux variations de luminance et très peu sensible aux variations de chrominance [Bar 07].





Figure 2.6 : Décomposition suivant les composantes RVB et YCbCr

▪ **Transformation DCT (discrete cosine transform) :**

La transformée en cosinus discrets ou DCT (Discret Cosine Transform) est une transformation linéaire en général. Elle permet de prendre un ensemble de points d'un domaine spatial et d'en donner une représentation en domaine fréquentiel.

La DCT effectuée sur une matrice de 8x8 pixels, après avoir découpée l'image en blocs de 8x8. Cette taille de 8x8 est celle qui a été retenue par la méthode JPEG.

A un point Imge (x, y) d'une image on associe un point F (u, v) par la transformation suivante :

**Erreur ! Signet non**

**défini.** 
$$F(u, v) = \frac{2}{N} c(u).c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} img(x, y). \cos \left[ \frac{\lambda}{2} u \left( x + \frac{1}{2} \right) \right]. \cos \left[ \frac{\lambda}{2} v \left( y + \frac{1}{2} \right) \right]$$

Voici son inverse (connue aussi sous le nom de IDCT).

$$img(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u).c(v).F(u, v). \cos \left[ \frac{\lambda}{N} u \left( x + \frac{1}{2} \right) \right]. \cos \left[ \frac{\lambda}{N} v \left( y + \frac{1}{2} \right) \right]$$

$$\text{Ou } C(x) = \frac{1}{\sqrt{2}} \text{ si } x \text{ vaut } 0, \text{ et } 1 \text{ si } x > 0.$$

Le coefficient de l'élément (0,0), appelé coefficient continu (*DC*), est le plus élevé et est proportionnel à la valeur moyenne du bloc de 64 pixels. Les autres coefficients, ou coefficients alternatifs (*AC*), correspondent aux variations d'un pixel à l'autre

- **Quantification :** C'est lors de cette étape que l'image va être réellement compressée. Chaque bloc de coefficients DCT est divisé par une table de quantification. Cette division s'effectue coefficient à coefficient et le résultat est arrondi à l'entier le plus proche.

$$\text{Coefficient } [i] [j] = \text{ROUND} \left( \frac{\text{Coefficient}[i][j]}{\text{Quantification}[i][j]} \right)$$

Les tables de quantification sont construites en fonction de la qualité *Q* à partir de la formule :

$$\text{Quantification}[i] [j] = 1 + (i+j+1) * \text{qualité}$$

- **Codage entropique:** Comme le coefficient (0,0) est proportionnel à la valeur moyenne du bloc de pixels, il ne varie pas beaucoup d'un bloc à l'autre : on en code la différence par rapport au bloc précédent.

Les autres coefficients sont parcourus selon une séquence dite "zigzag" (figure 2.7) pour former une suite de coefficients qui contient de nombreux zéros consécutifs.

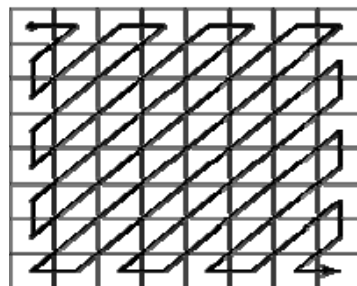


Figure 2.7 : Parcours en Zigzag d'un bloc 8x8

- **RLE (Run Length Encoding)**

Le principe du RLE (Run Length Encoding) est de coder les répétitions d'un même pixel : un premier octet indique le nombre de pixels identiques et le second indique la valeur de ces pixels. Si un même pixel n'est pas répété au moins 3 fois, alors la suite d'octets n'est pas codée.

**Exemple :**

157-157-157-157-157-157-007-007-007-007-000-255-255-255-255-089-089-089-089-089-089

=> 006/157-004/007-001/000-004/255-006/089

▪ **Codage Huffman**

Cet algorithme, inventé par David Huffman en 1952, permet de coder les octets les plus fréquents sur peu de bits.

Le principe de ce codage est la création d'un arbre dont les feuilles sont les valeurs à coder [Bar 07]. Un poids est associé à chacune de ces feuilles, et il correspond à la fréquence d'apparition de leurs valeurs.

Ensuite, on associe les deux noeuds de poids le plus faible pour donner un noeud de poids égal à la somme des poids de ces noeuds. Et on recommence jusqu'à ce que tous les noeuds soient reliés jusqu'à la racine de l'arbre binaire obtenu.

Enfin pour chaque noeud on associe la valeur "0" à la branche amenant au noeud de poids le plus faible et "1" à l'autre.

A l'association de ces valeurs est créée la table des correspondances (table de Huffman), qui sera stockée dans le header du fichier JPEG avec les tables de quantification.

## Deuxième partie

### 2.3 TECHNIQUES DE LA STEGANOGRAPHIE

Nous savons qu'il existe deux types d'images, les images vectorielles et les images matricielles. Les images matricielles sont les plus fréquemment utilisées dans la stéganographie car celles-ci sont issues d'appareils photos numériques, de scanners ou de caméras vidéos numériques [Lo 08]. C'est donc sur ce type d'image que l'étude de la stéganographie sera portée.

Cette partie va détailler quelques techniques stéganographique utilisée couramment dans ce type d'image.

Selon [Nik 08], il est possible d'établir une hiérarchie au niveau des techniques stéganographiques. En partant du moins sécurisé, cette hiérarchie serait :

1. L'ajout du message à la fin du fichier.
2. L'ajout du message dans les espaces inutilisés du conteneur.
3. Ajout du message dans les données de l'image de manière séquentielle.
4. Ajout du message dans les données de l'image en utilisant une séquence pseudo aléatoire.
5. Ajout du message dans les données de l'image en utilisant une séquence pseudo aléatoire, en prenant soin de modifier les données inutilisées afin de ne pas être visible d'un point de vue statistique.

Ces cinq points peuvent être regroupés en deux catégories distinctes. Pour les deux premiers points, ils correspondent à la technique dite **stéganographie basée sur la structure du fichier**. Pour les trois derniers points, la stéganographie dépend de domaine de représentation des images (domaine **spatial** ou **fréquentiel**).

### 2.3.1 STEGANOGRAPHIE BASEE SUR LA STRUCTURE DU FICHIER

Cette technique consiste à utiliser des emplacements inutilisés ou non lu par la plupart des décodeurs d'image. On distingue deux fonctionnements : L'ajout de données en fin de fichier et l'ajout au niveau des en-têtes de fichier.

L'ajout en fin de fichier est rendu possible par le fait que la plupart des décodeurs d'image ne lisent pas le fichier image dans son ensemble. Pour la plupart des formats d'image disponible, une certaine chaîne de bits est définie afin de marquer la fin de l'image (figure 2.8). L'ajout en fin d'image insère simplement après cette chaîne les données dissimulées.

9D	EE	7C	5B	72	B7	0C	D3	0F	B5	81	89	0E	EF	E2	27	
BD	68	F8	96	DA	DE	DA	F6	71	6D	6						FE
AD	02	F6	F6	A2	8A	8A	7F	C4	87	F						OF
3A	9C	9						7	54	63	F					FD
28	A2							C	CO	70	3A	7F	8D	66	CD	FF
00	1F	23	FD	D1	45	14	33	5F	B2	68	D8	7F	AA	15	6E	
7E	3C	BC	71	8C	D1	45	54	4C	67	B9	96	BF	7B	3F	E7	
BD	14	51	43	25	9F	FF	D9	9E	97	BA	2A	00	80	88	C9	
A3	70	97	5B	A2	E4	99	B8	C1	78	7						34
2B	4E	7D	31	7F	B5	E8	70	39	A8	B						91
EO	4F	39	14	1F	96	0D	0A	08	0D	6						38

Diagramme illustrant la structure d'un fichier image en fin de fichier. Le tableau ci-dessus représente des données hexadécimales. Les zones sont étiquées comme suit :

- Données de l'image** : Zone en haut à droite (à partir de l'hexadécimale 71).
- Indicateur de fin d'image** : Zone au milieu à gauche (à partir de l'hexadécimale 3A).
- Données Cachées** : Zone en bas à droite (à partir de l'hexadécimale 9E).

**Figure 2.8 : Exemple d'ajout en fin de fichier JPEG [Nik 08]**

Pour ce qui est de l'ajout dans les en-têtes, certains formats comme le bitmap définissent un champ permettant de spécifier l'offset à partir duquel l'image commencera. En spécifiant un offset un peu plus long il est possible de cacher les données à dissimuler.

Une autre approche est d'utiliser des champs spécifiés dans la norme du format. En prenant exemple sur les formats JPEG ou PNG, ils définissent des champs permettant de saisir des commentaires. Ces champs peuvent être détournés de leur usage afin d'y dissimuler des données. Les décodeurs ne tiennent pas compte du contenu de ces derniers, la modification est donc invisible. **Invisible Secret 2.1** utilise les champs de commentaire du format JPEG pour camoufler l'information.

L'avantage de cette méthode est qu'elle ne limite pas la taille des données cachées. Cependant, elle modifie la taille du fichier dans sa globalité.

Le niveau de sécurité de ce type de stéganographie peut être vu comme très faible. La détection est très facile. Si aucun chiffrement n'a été entrepris sur les données, un simple éditeur hexadécimal peut être utilisé pour les récupérer.

**2.3.2 STEGANOGRAPHIE DANS LE DOMAINE SPATIAL****2.3.2.1 Domaine spatial**

Le domaine spatial est le domaine classique où chaque valeur en  $(x, y)$  correspond à la valeur des pixels, nous pouvons alors la visualiser dans un espace à 3 dimensions où les axes X et Y représentent deux dimensions de l'image, et l'axe Z représente la valeur des pixels.

Les images fixes appartiennent au domaine spatial apparaissent dans de nombreux formats, notamment BMP, Raw, XBitmap, etc. Chaque format correspond à une structure particulière de représentation et de stockage des informations relatives à l'image (données, taille, nombre de bits par donnée . . .).

Toute une série d'algorithmes de stéganographie a été développée mais le point central reste le même : la modification des bits de poids faible (**Least Significant Bits : LSB**) des octets de chaque composante.

**2.3.2.2 Stéganographie dans des bits de poids faible (LSB)**

La méthode d'insertion de données sur les bits de poids faible ou LSB est la technique de stéganographie d'image la plus connue. Son principe est d'utiliser le dernier bit de chaque nombre définissant l'intensité d'une couleur primaire d'un pixel, pour les données à dissimulées (figure



2.9). Ainsi, trois bits peuvent être encodés dans chaque pixel, et la différence de couleur obtenue est imperceptible pour l'œil [Bar 07].

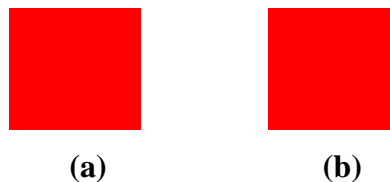
Dans le cas d'une image non compressée codée sur 24 bits, chaque pixel est décrit par trois octets.  
 Pour dissimuler la chaîne de bits, on utilise le bit de poids faible de chaque octet

(00100111 11101001 11001000)		(00100111 11101000 11001000)
(00100111 11001000 11101001)	+ 10000011 =	(00100110 11001000 11101000)
(11001000 00100111 11101001)		(11001000 00100111 11101001)

**Figure 2.9 : Insertion dans les bits de poids faible**

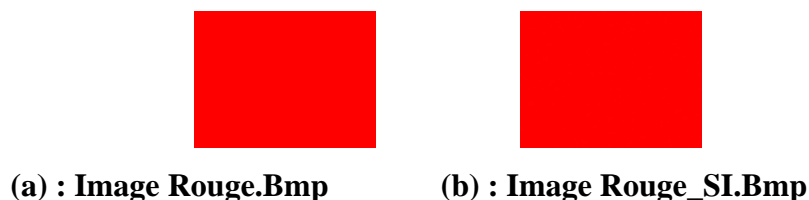
Les changements des bits de poids faible (de 0 à 1 ou de 1 à 0) sont totalement imperceptibles pour l'œil puisqu'il n'est modifié que d'un point.

Prenons l'exemple d'une image rouge ayant 256 possibilités de représentation (figure 2.10). Il est évident qu'à l'œil nu, la différence entre le 254ème et le 255ème rouge n'est pas visible.



**Figure 2.10 : 255ème rouge (a), 254ème rouge (b)**

Insérons maintenant le texte « **petit texte d'essai** » dans l'image **rouge.Bmp**, avec le logiciel **Invisible Secrets**<sup>3</sup>, l'image obtenue par modification des bits de poids faible ne paraît pas suspecte car elle est visuellement proche de l'originale (figure 2.11).



**Figure 2.11 : image Rouge originale (a), stéganographiée avec Invisible secret (b)**

Avec l'éditeur hexadécimal **Hex Editor** [Web 2.3], voici comment se compose l'image rouge.bmp (figure 2.12).

<sup>3</sup> Logiciels de stéganographie 2004-2006

```

42 4D 66 75 00 00 00 00 00 00 36 00 00 00 28 00
00 00 64 00 00 00 64 00 00 00 01 00 18 00 00
00 00 30 75 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 FF 00 00 FF 00 00 FF 00
00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF
00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00
00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00
FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF 00 00 FF

```

**Figure 2.12 : Fragment du fichier Rouge.bmp en Hexadécimal**

A partir de 54<sup>ème</sup> octets, les composantes de couleur RGB de chaque pixel sont marquées selon l'ordre B, G puis R (00 00 FF)

Comme l'image est toute rouge, les composantes RGB de chaque pixel sont :

- Composante bleu (B=00) :  $0*16 + 0 = 0$
- Composante vert (G=00) :  $0*16 + 0 = 0$
- Composante rouge (R=FF) :  $15*16 + 15 = 255$ .

En lisant maintenant le fichier Rouge\_SI (image stéganographiée) en hexadécimal, on trouve l'ensemble des octets illustrés par la figure suivante :





```

42 4D 66 75 00 00 00 00 00 00 36 00 00 00 28 00
00 00 64 00 00 00 64 00 00 00 01 00 18 00 00
00 00 30 75 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 01 00 FE 00 00 FE 01 00 FE 00
01 FE 00 00 FF 00 00 FE 01 00 FE 00 00 FF 00 00
FE 01 00 FF 00 01 FF 01 00 FF 01 01 FF 00 00 FF
01 00 FE 01 01 FF 00 01 FF 01 00 FE 00 00 FF 00
01 FF 01 01 FF 01 01 FE 00 00 FE 01 00 FF 01 10
FE 01 00 FF 00 01 FF 00 01 FF 01 01 FF 01 00 FE

```

**Figure 2.13 : Fragment du fichier Rouge stéganographié.bmp en Hexadécimal**

On remarque que l'insertion du message caché dans l'image propre (Rouge.bmp), conduit à la modification des composantes de couleur de pixels (00 00 FF) par des couleurs illustrées dans la figure 2.14, qui sont vraiment imperceptible pour l'œil.

• 01 00 FE	→	(1,0 254)	
• 00 00 FE	→	(0,0 254)	
• 00 01 FF	→	(0,1 255)	
• 01 01 FF	→	(1,1 255)	

**Figure 2.14 : Différentes couleurs rouge utilisées pour la dissimulation**

L'avantage du LSB est que la taille du fichier n'est pas modifiée, puisque le message est encodé dans les parties peu ou pas utilisées du fichier. C'est également une méthode rapide et facile à mettre en oeuvre. Cependant, le LSB possède un désavantage de taille, à savoir la perte du message lorsque des changements importants ont lieu sur le support, comme par exemple une rotation, ou une compression de l'image.

Une amélioration du LSB consiste à introduire un paramètre aléatoire permettant de distribuer les bits de poids faible utilisés et de dissimuler le message dans différents plans de bits.

Ainsi, les modifications n'auront pas lieu "uniquement" dans les premiers octets de l'image ou dans les derniers bits, mais seront au contraire répartis aléatoirement dans l'entièreté de l'image.

### 2.3.2.3 Stéganographie dans les images GIF

Une image GIF est codée par une entrée (codée sur un seul octet) dans une table de couleurs. Cette table de couleurs, la palette, définit jusqu'à 256 couleurs utilisables dans l'image en donnant la décomposition RGB de chaque couleur sur 3 octets. Ainsi, le fichier comporte essentiellement une série d'octets pour la palette et une série d'octets pour les pixels.

On peut utiliser les bits de poids faible d'un pixel pour dissimuler un message, c'est-à-dire l'entrée dans la table, mais les couleurs de la table peut être très différentes et deux entrées consécutives peuvent correspondre à des couleurs nettement différenciables. Modifier ainsi l'image se traduirait donc très probablement par un phénomène visible, ce qui est contraire à l'objectif. Il existe cependant d'autres possibilités :

- **modifier l'ordre des couleurs dans la table** : La première étape consiste à trier la palette selon un ordre minimisant les différences de couleur entre deux entrées adjacentes, ensuite, on réécrit l'image en fonction de la palette triée.
- **modifier les bits de poids faible** des codes RGB des couleurs de la table.

L'algorithme EzStego [Pet 02] utilise une combinaison des deux : il calcule une palette de couleurs triée selon un ordre qui minimise les différences de couleur entre les entrées adjacentes.

Pour dissimuler un bit, il considère le code du pixel et regarde la position de la couleur correspondant à ce code dans la palette triée. Le dernier bit de la position dans la nouvelle table constitue l'information cachée, il est donc réécrit s'il est différent du bit à dissimuler.

On obtient ainsi une nouvelle position dans la palette triée, donc une nouvelle couleur. Le code du pixel est alors modifié pour correspondre à la position de cette nouvelle couleur dans la palette d'origine.

### 2.3.3 STEGANOGRAPHIE DANS LE DOMAINE FREQUENTIEL

**2.3.3.1 Domaine fréquentiel** : Le domaine fréquentiel est un espace dans lequel l'image sera considérée comme une somme de fréquences de différentes amplitudes (voir DCT).

#### 2.3.3.2. Stéganographie dans les coefficients DCT

Toute une série d'algorithmes a été développée mais le point central reste le même; la dissimulation d'informations lors de la compression de l'image.

Cette technique permet d'insérer des données sur les coefficients de la transformée de chaque bloc après la quantification et modifient les composantes les moins visibles.

L'insertion de données dans les coefficients de la transformée est basée sur :

- **La modification des bits de poids faible de ces coefficients** (algorithme **Jsteg**) [Web 2.5]: comme pour la stéganographie dans le domaine spatial.
- **la décrémentation de la valeur absolue des coefficients DCT** (eg : algorithme **F5**) [Wes 01].

La fonction d'insertion de cet algorithme se décrit comme suit

1. les coefficients nuls ne servent pas

2. si le coefficient est impair négatif ou pair positif, il correspond à un bit 0 du message dissimulé. Sinon, c'est un 1
  3. Lorsque l'on veut dissimuler un bit  $b$  dans un coefficient, soit le coefficient correspond déjà à  $b$  et on n'a rien à faire, soit on décrémente le coefficient
  4. lorsque, après décrémentation, on obtient un coefficient nul, on considère que le bit n'a pas été inséré. En effet, on ne dispose d'aucun moyen de distinguer le cas où le zéro a été obtenu suite à l'insertion ou bien s'il existait auparavant. Cela pose un problème pour l'extraction du message, sauf si on ignore simplement les coefficients qui peuvent être nuls après la décrémentation.
- **L'insertion de données par utilisation de la matrice de quantification :**  
Dans la matrice de quantification (figure 2.15), on utilise les composantes ayant la même valeur afin de dissimuler les données.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	81
49	68	74	87	101	121	120	101
72	92	95	112	112	100	103	33

**Figure 2.15: Matrice de quantification**

Par exemple on peut prendre les composantes (3,2) et (4,1) valant 22. Après la multiplication scalaire des deux matrices (DCT, Quantification), on obtient une matrice B. Si l'on veut coder un 1, alors on mettra la donnée dans la composante ayant la valeur la plus élevée, si l'on veut coder un 0, c'est le contraire.

Le principal désavantage de cette technique est que les algorithmes stéganographiques fonctionnant avec la DCT ne sont pas très résistants aux transformations géométriques comme les translations ou les rotations.

## 2.4 CONCLUSION

Dans ce chapitre nous avons exposé, dans une première partie, quelques concepts de base des images numériques, pour lesquels les techniques stéganographiques s'appuient. On distingue deux types d'images, les images vectorielles et les images matricielles. Les images matricielles sont les plus fréquemment utilisées dans la stéganographie. Ainsi les formats les plus utilisés pour la dissimulation d'information, ensuite nous avons présenté les différents domaines de représentation des images numériques.

Dans une deuxième partie, nous avons montré comment utilisé la stéganographie dans les images fixes.

Les algorithmes de stéganographie, généralement, exploitent les caractéristiques des images fixes (les multiples types d'images, leurs spécificités, changement de format, choix d'un espace de couleurs, et différents domaines de représentations) pour la dissimulation d'information.

Le chapitre qui suit décrit les concepts et les différentes techniques de la stéganalyse ou l'analyse stéganographique. La stéganalyse est la contrepartie de la stéganographie, elle s'intéresse à la détection, extraction ou destruction des informations cachées dans un support numérique.

# **La Stéganalyse dans le domaine spatial**

## CHAPITRE 3

### LA STEGANALYSE DANS LE DOMAINE SPATIAL

#### 3.1 INTRODUCTION

L'objectif principal de la stéganographie est de dissimuler un message secret dans un médium de couverture n'ayant rien à voir avec lui de façon qu'un attaquant ne puisse pas savoir si des informations sont dissimulées dans le médium de couverture.

Le premier argument que nous avons mentionné pour motiver l'intérêt d'une étude de la dissimulation d'information, en particulier la stéganographie, lui donne le beau rôle; assurer la confidentialité. Mais le réel problème si cette confidentialité sert à dissimuler aux yeux de la justice des actions illégales.

C'est le cas par exemple des **terroristes** qui ont utilisés la stéganographie, les incidents de 11 septembre 2001, comme un moyen pour coordonner les attentats en utilisant des messages cachés dans des images [Bar 07].

Les **pirates informatiques** peuvent aussi utiliser cet art pour camoufler leurs attaques. Un **hacker** peut très bien dissimuler des codes fragmentés à travers des **stégo-médium** (ex: images) et procéder au réassemblage du code malveillant directement sur l'ordinateur de la victime. Le hacker peut également dissimuler un cheval de Troie et prendre possession de la machine.

C'est la raison pour laquelle plusieurs études ont été réalisées pour détecter si un support est susceptible de contenir des informations supplémentaires indépendantes de ce dernier par un algorithme de stéganographie et de révéler ensuite ces informations.

Ce type d'étude constitue la **stéganalyse** ou **l'analyse stéganographique**, en d'autres termes la contrepartie de la stéganographie, dont l'objectif principal est la détection de l'utilisation de la stéganographie.

Ce chapitre présente une description générale de l'analyse stéganographique, ainsi les différentes méthodes de stéganalyse. Selon le type des mesures effectuées pour la distinction entre le stégo-



médium et le cover-médium, nous distinguons deux types de stéganalyse. Si les mesures dépendent des algorithmes que nous essayons de détecter, la stéganalyse est dite spécifique.

Lorsque les mesures sont indépendantes de l'algorithme que l'on cherche à détecter. La stéganalyse est dite universelle.

### 3.2. DESCRIPTION DE LA STEGANALYSE

La stéganalyse est la technique qui permet de déceler la stéganographie. Le fait de vouloir déceler des méthodes stéganographique s'appelle une attaque.

Comme en cryptographie les attaques peuvent avoir pour but d'extraire les données cachées (très difficile), ou de les détruisent (très facile).

L'attaquant doit tout d'abord savoir si le document contient des données cachées. Il existe plusieurs formes d'attaques selon les moyens dont dispose l'attaquant [Sil 01]:

- ✚ **Attaque avec stégo-medium seul (Stego-only attack):** Seul le stégo-medium est connu. L'insertion d'un message change certaines caractéristiques statistiques du cover-médium (*e.g.*: histogramme, égalité des cardinaux,...). L'attaque est basée sur cette altération.
- ✚ **Attaque avec cover et stégo medium (Known cover attack):** Le medium de couverture et le stégo-medium sont disponibles. Ce type d'attaque est basé sur la comparaison entre le cover-médium et le stégo-médium (*e.g.*: attaque visuelle).
- ✚ **Attaque sur message connu (Known message attack):** Certaines parties du message caché sont connues de l'utilisateur. L'attaquant va essayer de retrouver dans le stégo-medium les parties du message qu'il connaît afin de faciliter l'analyse des documents futur. Même avec le message connu cette attaque est très difficile et généralement considérée comme équivalent à l'attaque stégo-only.
- ✚ **Attaque avec un algorithme choisis (Chosen stego attack):** L'algorithme et le stégo-medium sont connus.
- ✚ **Attaque avec un message choisis (Chosen message attack):** Le stéganalyste génère un stégo-medium à l'aide du message choisis. Le but est d'observer le résultat pour cracker l'algorithme.
- ✚ **Attaque avec un algorithme connu (Known stego attack):** L'algorithme, le médium de couverture et le stégo-medium sont connues.

### 3.3. TYPES DE STEGANALYSE

La stéganalyse peut être appliquée par deux types de personnes. *L'attaquant actif*, qui connaît la présence de l'information et tente de la modifier ou de l'extraire et *l'attaquant passif*, c'est-à-dire la personne qui arrive à déceler la présence du message et qui ne fait que constater sa présence.

**3.3.1 Attaque active:** dans ce type d'attaque on souhaite non seulement détecter le message caché mais, en plus, on va chercher à extraire, modifier ou supprimer ces données. Cette destruction aura souvent lieu par l'intermédiaire de modification de support (*e.g.*: Compression, changement de format, recadrage, symétrie,...).

**3.3.2 Attaque passive:** il s'agit simplement de détecter la présence de messages dissimulés. Ce type d'attaque peut prendre plusieurs formes:

- La lecture ou l'écoute de fichier,
- La comparaison avec le fichier original (s'il est disponible),
- Certaines attaques statistiques (attaque sur le LSB),
- La détection des signatures des logiciels utilisés (étude du code hexadécimal).

### 3.4. LES METHODES DE LA STEGANALYSE

Plusieurs méthodes peuvent être utilisées pour la détection des images stéganographiées. La plus simple profite de la non performance des logiciels de stéganographie pour déceler les données cachées. De plus, certains logiciels n'hésitent pas à tricher, c'est le cas par exemple du logiciel **Invisible secrets** qui est censé utiliser le format JPEG pour camoufler les données, or ce logiciel ne fait que cacher les données dans les commentaires de l'en-tête du fichier.

La détection d'irrégularité statistique est une autre catégorie d'analyse. Elle se base sur la quantification de distorsion du média analysé, comparativement à des distributions statistiques théoriques représentant un média de base.

Selon le type des mesures effectuées pour la distinction entre les images de couverture et les stégo-images, nous distinguons deux types de stéganalyse. La stéganalyse universelle et la stéganalyse spécifique.

Dans cette partie, seul des stéganalyses dans le domaine spatial seront détaillées. Pour ce qui est de la stéganalyse dans le domaine fréquentielle, une courte description sera donnée.

### 3.4.1 STEGANALYSE UNIVERSELLE

Si les mesures utilisées pour la détection sont indépendantes des algorithmes que nous essayons de détecter, la stéganalyse est dite *universelle*. La stéganalyse universelle permet alors de répondre à la question « *le médium est-il Stéganographié ?* ».

#### 3.4.1.1 ATTAQUE VISUELLE

L'insertion d'un message dans le dernier plan de bit peut se faire de façon aléatoire sur l'ensemble des pixels de l'image ou de façon séquentielle à partir de début de l'image. L'idée de cette attaque est basée sur le fait que une image peu texturée, le plan LSB est corrélé avec l'image d'origine. L'insertion du message perturbe le plan LSB en proportion de la taille de message. Les attaques visuelles appliquent des filtres sur l'image originale (figure 3.1) et l'image stéganographiée, supprimant les composantes les plus visibles (bits de poids forts) et renforçant les autres (bits de poids faible), [Gal 04].



**Figure 3.1. : Image originale**

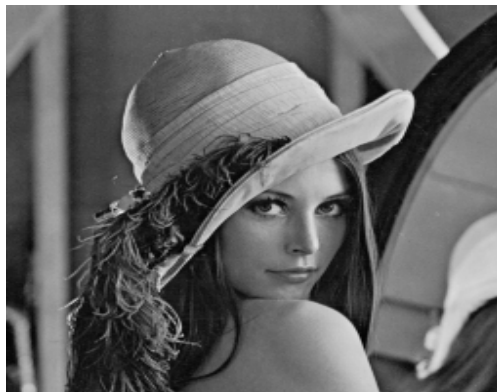
La figure 3.2 représente le dernier plan de bit de l'image suivante, dégradé de 1280x960 pixels, avant et après insertion d'un message dans le plan LSB.

Le dernier plan de bit de l'image initiale montre une régularité qui correspond à la régularité de l'image initiale (figure 3.1). On remarque ainsi que l'image stéganographiée est très bruitée et laisse apparaître de façon claire la présence d'un message dans l'image.

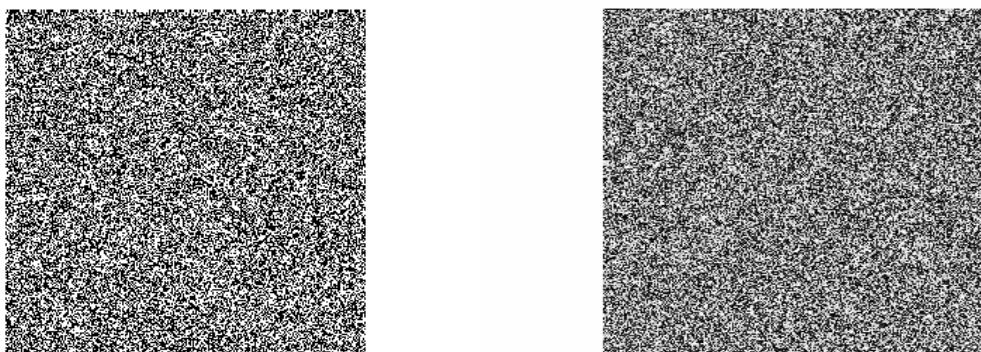


**Figure 3.2.: Dernier plan de bit avant et après insertion de l'image dégradé**

L'image Lina de la figure 3.3, a été choisie car c'est une image naturelle qui possède des zones homogènes assez grandes et en assez grand nombre. En comparant les images de la figure 3.4, on peut conclure que le même test ne montre aucun artéfact à l'œil.



**Figure 3.3 : Image Lena originale**



**Figure 3.4.: Dernier plan de bit avant et après insertion de l'image Lena**

L'attaque visuelle décrite si dessus n'est pas efficace contre les méthodes d'insertion courantes, utilisant essentiellement une insertion aléatoire, ni sur les images très texturées.

### 3.4.1.2 STEGANALYSE BASEE SUR DES PAIRES DE VALEURS DE L'IMAGE

Différentes sont les méthodes de stéganalyse basées sur l'analyse statistique des paires de valeurs de pixels. Le principe de ces méthodes se base sur le choix des sous ensembles des paires de pixels ou bien le choix des groupes de pixels vérifiant des hypothèses proposées pour la stéganalyse (eg: égalité des sous ensembles). La détection se base sur le fait que l'insertion d'un message dans les bits de poids faibles peut modifier ou ne vérifier pas une des hypothèses proposées. Dans cette partie, nous présentons l'analyse statistique à base de  $\chi^2$  [Wes 99], le schéma de Memon basée sur les paires de pixels [Mem 01] et le schéma proposée par Fridrich basée sur les groupes de pixels [Fri 01]. Pour les autres schémas [Rou 04], [Dum 03], c'est le même principe que celui définit dans [Mem 01], mais la différence intervient dans le choix des sous ensembles.

#### 3.4.1.2.1 Analyse statistique à base de $\chi^2$

La stéganalyse  $\chi^2$  proposée par Westfeld et Pfitzmann dans [Wes 99] est basée sur le principe que, les fréquences d'apparition des nuances d'une paire de valeurs tendent à l'égalité sous l'action d'insertion LSB (voir les histogrammes en figure 3.5).

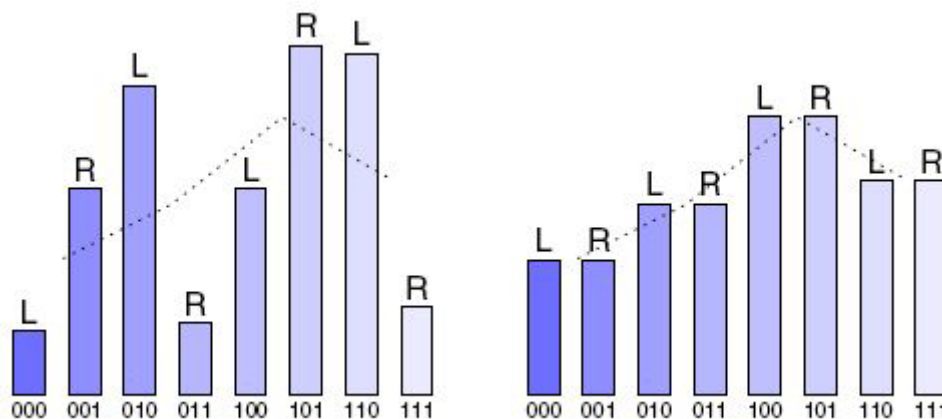


Figure 3.5. : Partie d'un histogramme d'une image avant et après insertion [Wes 99].

Sur la figure précédente, la ligne en traitillé correspond à la moyenne des deux valeurs composant une paire de valeur de pixel. Comme on peut le remarquer, cette moyenne n'est pas affectée par l'insertion.

Cette moyenne sera donc utilisée afin de créer un modèle théorique à partir d'un fichier dont on ne sait pas si il a été modifié. La moyenne théorique sera donc calculée à partir de la formule suivante

$$y_i^* = \text{Erreur ! Signet non défini.} \frac{n_{2i} + n_{2i+1}}{2}$$

(3.1)

La fréquence mesurée sur l'image analysée est  $y_i = n_{2i+1}$ . Alors la valeur du  $\chi^2$  mesurant la différence des distributions est :

$$\chi_{k-1}^2 = \sum_{i=1}^k \frac{(y_i - y_i^*)^2}{y_i^*} \quad (3.2)$$

Où  $k-1$  représente le degré de liberté (nombre des paires de valeurs de pixels).

La probabilité que les deux distributions soient identiques est donnée par l'expression suivante

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi_{k-1}^2} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx \quad (3.3)$$

Avec  $\Gamma$  représente la fonction gamma d'Euler qui s'écrit :

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt \quad (3.4)$$

Pour mieux visualiser l'application de cette méthode à une image, on applique cette dernière à l'exemple suivant.

Soient l'image (C) de  $4 \times 4$  pixels et le message à insérer (1001101000111001).

$$c = \begin{bmatrix} 00 & 00 & 10 & 10 \\ 01 & 11 & 10 & 00 \\ 00 & 11 & 10 & 00 \\ 00 & 00 & 00 & 00 \end{bmatrix} + 1001101000111001 = \begin{bmatrix} 01 & 00 & 10 & 11 \\ 01 & 10 & 11 & 00 \\ 00 & 10 & 11 & 01 \\ 01 & 00 & 00 & 01 \end{bmatrix}$$

Les paires des valeurs de pixels sont [00,01] et [10,11]. La première étape consiste à calculer le nombre d'occurrences de chaque élément, ensuite chaque élément sera classé à l'aide d'un indice suivant le tableau suivant.

Indice	Élément
0	00
1	01
2	10
3	11

**Tab. 3.1 : Correspondance entre indice et élément**

En classe ensuite les éléments des deux images (image originale et image stéganographiée) dans les divers indices, cela donne le tableau d'occurrences suivant.

Indice	0	1	2	3
Image originale	9	1	4	2
Image stéganographiée	5	5	3	3

**Tab.3.2 Occurrences de chaque élément**

Ensuite, à l'aide de la formule (3.1), la distribution des occurrences théorique peut être calculée. Dans cet exemple formé de deux paires de valeurs, elle sera formée de deux valeurs.

$$y_0^* = \text{Erreur ! Signet non défini.} \frac{n_0 + n_1}{2} = 5$$

$$y_1^* = \text{Erreur ! Signet non défini.} \frac{n_2 + n_3}{2} = 3$$

Ensuite, avec la formule (3.2), il est possible d'appliquer le test du  $\chi^2$  à notre image stéganographiée, ainsi qu'à notre image originale.

$$\chi_{k-1}^2_{\text{Stéganographiée}} = \chi_{k-1}^2 = \sum_{i=1}^k \frac{(y_i - y_i^*)^2}{y_i^*} = \frac{(5-5)^2}{5} + \frac{(3-3)^2}{3} = 0$$

$$\chi_{k-1}^2_{\text{Stéganographiée}} = \chi_{k-1}^2 = \sum_{i=1}^k \frac{(y_i - y_i^*)^2}{y_i^*} = \frac{(1-5)^2}{5} + \frac{(2-3)^2}{3} = \frac{53}{15}$$

Il est ensuite possible de calculer la probabilité d'avoir un message dissimulé via l'équation (3.3).

Le résultat du  $\lambda^2$  intervenant dans les bornes d'intégration de la fonction de densité, en ce qui concerne l'image stéganographiée, le résultat est aisément identifiable. Le résultat de l'intégration étant nul, le membre de droite de la soustraction s'annule. Cela donne donc une probabilité de 1.

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)} \int_0^0 e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx = 1 - 0 = 1$$

En ce qui concerne le même calcul pour l'image originale, le même raccourci n'est pas possible. Il faut donc calculer l'entier de l'équation.

$$p = 1 - \frac{1}{2^{\frac{1}{2}} * 1.772} \int_0^{\frac{53}{15}} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx = 0.144$$

Il y a donc une probabilité de 14.4 % que l'image originale dissimule des données, ce qui est bien sûr erroné.

En conclusion, cette méthode donne de bon résultat, mais elle permet uniquement la détection de données cachées de manière séquentielle au niveau des LSB. De plus, elle ne permet qu'une estimation de la taille des données approximative.

Dans [Pro 03], cette méthode a été adaptée avec succès aux images JPEG. Ceci a été rendu possible en appliquant la même logique que ci-dessus, mais cette fois-ci sur les coefficients DCT des images JPEG.

#### 3.4.1.2.2 Stéganalyse RS (Régulier, Singulier)

Cette méthode, développée par Fridrich, Miroslav Goljan et Rui Du [Fri 01], est basée sur la classification des groupes de pixels en catégories distinctes.

Pour une image en 256 niveaux de gris (8 bits), chaque pixel a une valeur comprise dans l'ensemble  $P = \{0, 1, 2, \dots, 256\}$ . La première opération à effectuer, est de diviser l'image en groupes disjoints de  $n$  pixels adjacents  $G = (x_1, \dots, x_n)$ . Ce nombre  $n$  est défini par les auteurs à 4. Ensuite, une fonction  $f$  permettant d'évaluer l'homogénéité de chacun de groupe  $G$  est défini par :

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n-1} (|x_{i+1} - x_i|) \quad (3.5)$$

Il est défini des fonctions réversibles sur l'ensemble  $P$ . Ces fonctions consistent essentiellement en des permutations des valeurs. Ces fonctions sont au nombre de trois :



$$F_1(x) = 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 254 \leftrightarrow 255$$

$F_1(x)$  est l'opération de permutation LSB d'un pixel de l'image.

$$F_{-1}(x) = -1 \leftrightarrow 0, 1 \leftrightarrow 2, \dots, 255 \leftrightarrow 256$$

$F_{-1}(x)$  est l'opération de permutation LSB tradlatée d'un 1.

$$F_0(x) = x$$

$F_0$  est la fonction identité.

Afin d'appliquer ces fonctions de permutation sur les groupes  $G$  de pixels définis plus haut, Il faut définir un masque de permutation  $M$ , ainsi que le masque inverse  $-M$ . Ces masques auront une taille de  $1 \times n$ , avec des valeurs comprises dans  $\{-1, 0, 1\}$ . Cela définira la fonction  $F$  de permutation à appliquer à chacun des membres des groupes de tel sorte de donner le groupe permuté  $F(G) = (F_{M(1)}(x_1), F_{M(2)}(x_2), \dots, F_{M(n)}(x_n))$ .

Pour chaque groupe  $G$ , nous évaluons  $F(G)$  et classifions ce dernier dans une des trois catégories distinctes. Ces catégories sont définies de la manière suivante :

$$\text{Groupe Régulier : } G \in R \Leftrightarrow f(F(G)) > f(G)$$

$$\text{Groupe Singulier : } G \in S \Leftrightarrow f(F(G)) < f(G)$$

$$\text{Groupe identité : } G \in U \Leftrightarrow f(F(G)) = f(G)$$

Les fonctions des permutations des valeurs des pixels simulent l'ajout de bruits à l'image source. Dans le cas d'une image, l'ajout de bruit aura comme influence une augmentation de la fonction de d'homogénéité. Au niveau de la classification des groupes, cela représentera une augmentation de ceux dans le groupe R.

En définissons  $R_M$  comme le pourcentage des groupes de type R et  $S_M$  le pourcentage des groupes de type S après l'application du masque  $M$ . De la même manière il faut définir  $R_{-M}$  et  $S_{-M}$  pour le masque inverse  $-M$ .

Il est possible de faire l'hypothèse que l'application du masque M ou du masque inverse, ne va pas changer de manière significative la distribution des groupes. Donc sur une image, les relations  $R_M \cong R_{-M}$  et  $S_M \cong S_{-M}$  devrait être vérifiées.

La base de cette méthode de Stéganalyse est l'estimation des courbes en fonction du pourcentage de LSB modifié des quatre paramètres  $R_M, S_M, R_{-M}$  et  $S_{-M}$ . Fridrich et son équipe en on déduit que les paramètres  $R_{-M}$  et  $S_{-M}$  peuvent être approximé à l'aide d'une droite. Pour les deux autres paramètres, ils peuvent être approchés par un polynôme du second degré.

Un petit exemple permettant de visualiser l'utilisation de cette méthode, en se basant sur l'image suivante :

$$I_{Base} = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

Cette image de 8x8 pixels sera utilisée comme image de base, donc non stéganographiée. La première opération à effectuer, est de diviser l'image en groupes disjoints de 4 pixels adjacents.

Pour la première ligne de l'image, les groupements suivants :

$$G_1 = (139 \ 144 \ 149 \ 153) \text{ et } G_2 = (155 \ 155 \ 155 \ 155)$$

Sur l'ensemble des groupes G définis, on applique la fonction d'homogénéité (3.5). Le tableau 3.3 donne les résultats obtenus sur l'image originale. Ces résultats seront nécessaires pour déterminer la classification de chaque groupe après avoir appliqué la fonction de permutation.

G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
F(G)	14	0	11	3	13	2	5	1	3	7	0	3	3	5	1	5

**Tab .3.3 : Résultat de la fonction d'homogénéité sur l'image de base**

Après avoir obtenu ces résultats, il faut maintenant perturber notre image d'origine à l'aide des fonctions réversibles, ainsi que le masque de permutation  $M = [0 \ 1 \ 1 \ 0]$ . L'image résultante aura l'allure suivante.

$$I_{Base\_permutée} = \begin{bmatrix} 139 & 145 & 148 & 153 & 155 & 154 & 154 & 155 \\ 144 & 150 & 152 & 156 & 159 & 157 & 157 & 156 \\ 150 & 154 & 161 & 163 & 158 & 157 & 157 & 156 \\ 159 & 160 & 163 & 160 & 160 & 158 & 158 & 159 \\ 159 & 161 & 160 & 162 & 162 & 154 & 154 & 155 \\ 161 & 160 & 160 & 161 & 160 & 156 & 156 & 157 \\ 162 & 163 & 160 & 163 & 162 & 156 & 156 & 157 \\ 162 & 163 & 160 & 161 & 163 & 159 & 159 & 158 \end{bmatrix}$$

Comparons les deux images (image de base et image de base après permutation), on remarque que seuls les éléments centraux sont modifiés en utilisant la fonction  $F_1$ . Le tableau 3.4 compare les valeurs obtenues pour chaque groupe, ainsi que leurs classifications.

G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(G)$	14	0	11	3	13	2	5	1	3	7	0	3	3	5	1	5
$f(F(G))$	14	2	12	3	13	2	7	3	4	9	2	5	7	7	5	5
Classification	U	R	R	U	U	U	R	R	R	R	R	R	R	R	R	R

**Tab. 3.4 : Résultat de la fonction d'homogénéité et classification**

On remarque que la théorie énoncée plus haut, disant que la perturbation de l'image provoque une augmentation de la fonction d'homogénéité est vérifiée. En effet, la majorité des groupes sont défini comme étant réguliers [Nik 08].

Maintenant, en dissimulant le message suivant et observer l'impact sur une image contenant un message.

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 111011

L'image résultant de la dissimulation du message, de manière séquentielle par modification des bits de poids faible (LSB), est la suivante :

$$I_{\text{Stéganographiée}} = \begin{bmatrix} 138 & 144 & 148 & 152 & 154 & 154 & 154 & 155 \\ 144 & 150 & 153 & 156 & 158 & 156 & 157 & 157 \\ 150 & 155 & 160 & 162 & 158 & 157 & 156 & 157 \\ 158 & 161 & 163 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 160 & 162 & 163 & 154 & 154 & 155 \\ 161 & 160 & 161 & 160 & 161 & 156 & 157 & 157 \\ 163 & 163 & 160 & 162 & 163 & 157 & 156 & 157 \\ 163 & 163 & 161 & 160 & 163 & 159 & 159 & 159 \end{bmatrix}$$

Comme sur l’image d’origine, il faut appliquer la fonction d’homogénéité (1), afin de calculer les valeurs de  $f(G)$ . Ensuite, à l’aide du même masque M que précédemment, il faut perturber cette image pour obtenir nos valeurs de  $F(f(G))$ . Le tableau 3.5 est une récapitulation des valeurs obtenues au cours de cet exemple.

G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(G_{\text{Base}})$	14	0	11	3	13	2	5	1	3	7	0	3	3	5	1	5
$f(F(G_{\text{Base}}))$	14	2	12	3	13	2	7	3	4	9	2	5	7	7	5	5
Classification <sub>Base</sub>	<b>U</b>	<b>R</b>	<b>R</b>	<b>U</b>	<b>U</b>	<b>U</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>R</b>	<b>U</b>
$f(G_{\text{Base}})$	12	1	12	3	12	3	8	1	3	10	3	6	5	8	3	4
$f(F(G_{\text{Base}}))$	14	1	12	3	12	3	6	3	3	8	1	6	3	8	3	6
Classification <sub>Base</sub>	<b>R</b>	<b>U</b>	<b>U</b>	<b>U</b>	<b>U</b>	<b>U</b>	<b>S</b>	<b>R</b>	<b>U</b>	<b>S</b>	<b>S</b>	<b>U</b>	<b>S</b>	<b>U</b>	<b>U</b>	<b>R</b>

**Tab. 3.5 : Récapitulation des résultats obtenus**

Le tableau 3.5 montre que l’image stéganographiée possède plus de groupes de pixel classé comme singulier que ceux étant définis comme régulier dans l’image de base.

En conclusion, la stéganalyse RS est très efficace pour la détection des informations cachées par la technique LSB et elle permette d’effectuer une estimation assez précise de la longueur de message dissimulé. D’autre part, cette méthode est plus efficace sur des images modifiées de manière aléatoire [Nik 08].

### 3.4.1.2.3 Stéganalyse basée sur les égalités statistiques

La méthode de stéganalyse proposée dans [Mem 01] est basée sur des égalités statistiques de cardinalités d'ensembles de paires de pixels (par exemple les paires de pixels adjacents). Les auteurs, dans cette méthode, considèrent deux sous ensembles de pixels  $X$  et  $Y$  sur l'ensemble des paires de pixels adjacents  $P$ . La stéganalyse est basée sur le fait que l'insertion d'un message dans le dernier plan de bit modifie l'égalité des cardinalités  $|X| = |Y|$  qui est vraie pour les images naturelles [Lec 03].

Soit  $P$  la palette des pixels utilisée dans l'image,  $P = \{0, 1, 2, \dots, 255\}$  pour une image en nuances de gris sur 8 bits. Soit  $P$  un ensemble de paires des pixels adjacents de l'image, les sous ensembles  $X$  et  $Y$  dans  $P$  sont définies de la manière suivante:

- $X = \{(u, v) \in P \text{ tels que } v \text{ soit pair et } u < v, \text{ ou } v \text{ impaire, et } u > v\}$
- $Y = \{(u, v) \in P \text{ tels que } v \text{ soit pair et } u > v, \text{ ou } v \text{ impaire, et } u < v\}$

De plus, trois sous ensembles sont utilisés pour définir les différentes manipulations LSB:

$Z \subset P$  est l'ensemble des paires de la forme  $(u, u)$

$W \subset P$  est l'ensemble des paires de la forme  $(2K, 2K+1)$  ou  $(2K+1, 2K)$

$V \subset P$  est le complémentaire de  $W$  dans  $Y$ :  $V=Y-W$

Le tableau 3.6 décrit le résultat de l'action d'une permutation LSB sur une paire d'un des sous ensembles précédents. L'opération 00 signifie qu'aucun élément de la paire n'est affecté par le LSB, 01 signifie que le second élément de la paire est affecté, 10 le premier et 11 signifie que les deux éléments de la paire sont modifiés.

Opération $\pi$	X	W	V	Z
00	X	W	V	Z
01	X	Z	V	W
02	V	Z	X	W
03	V	W	X	Z

**Tab. 3.6 : Modifications des différents ensembles sous insertion LSB.**

Soient  $\rho(\pi, P)$  la probabilité qu'une paire de pixels de  $P$  soit modifiée avec  $\pi$  et  $L$  la longueur du message inséré en pourcentage de la taille totale du plan de bits. Alors, si le message aléatoirement inséré dans l'image, cela nous amène aux probabilités suivantes :

$$\rho(00, p) = \left(1 - \frac{p}{2}\right)^2$$

$$\rho(01, p) = \rho(10, p) = \frac{p}{2} \times \left(1 - \frac{p}{2}\right)$$

$$\rho(11, p) = \left(\frac{p}{2}\right)^2$$

L'estimation de la longueur du message se fait en résolvant les équations suivantes en fonction des sous ensembles définies précédemment ( $X, Y, Z, V, W$ ) et de la probabilité  $\rho(\pi, p)$ .

$$|X'| = |X| \times \left(1 - \frac{p}{2}\right) + |V| \times \frac{p}{2}$$

$$|V'| = |V| \times \left(1 - \frac{p}{2}\right) + |X| \times \frac{p}{2}$$

$$|W'| = |W| \times \left(1 - p + \frac{p^2}{2}\right) + |Z| \times p \times \left(1 - \frac{p}{2}\right)$$

Si on calcule  $|X'| - |V'|$  on trouve,  $|X'| - |V'| = (|X| - |V|) \times (1-p)$ . De plus, on a  $|X| = |Y|$ , donc  $|X| = |V| + |W|$ . Alors  $|X'| - |V'| = |W| \times (1-p)$ . Si on pose  $|W| + |Z| = |W'| + |Z'| = \gamma$  alors, en utilisant l'équation de  $|W'|$  on trouve :

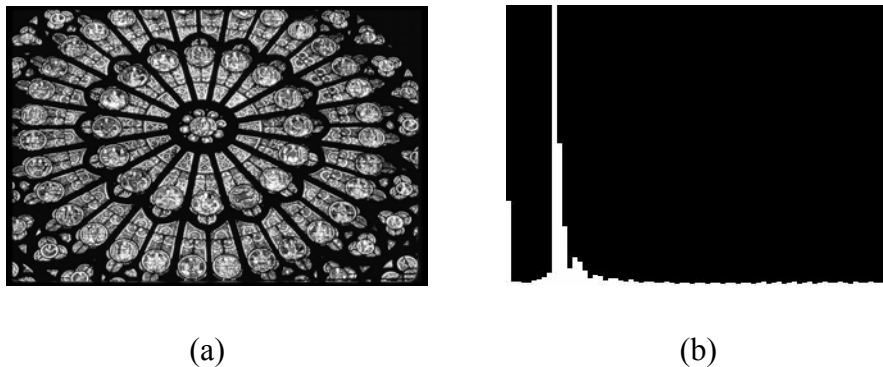
$$|W'| = (|X'| - |V'|) \times (1-p) + \gamma \times p \times \left(1 - \frac{p}{2}\right)$$

D'autre part  $|P| = |X'| + |V'| + |W'| + |Z'| = |X'| + |V'| + \gamma$ . L'équation précédente devient donc

$$1/2 \gamma^2 p^2 + (2|X'| - P) p + |Y'| - |X'| = 0$$

La racine la plus petite de cette équation donne la valeur de  $p$  correspondante aux cardinaux des ensembles considérés dans l'image analysée.

En conclusion la méthode de Memon est très fiable pour la plus part des images, mais dans certains cas l'erreur d'estimation devient significatif. De plus l'hypothèse principale  $|X| = |Y|$  n'est pas vérifiée pour certains images comme l'image notrewindow. L'histogramme de cette image est décrit dans la figure 3.6. Sur cette figure nous pouvons remarquer notamment la présence des pics singuliers fausse l'égalité des cardinaux [Lec 03].



**Figure 3.6: Image notrewindow(a) et son histogramme (b)**

### 3.4.1.3 METHODES DE STEGANALYSE DANS LE DOMAINE SPATIAL

Kobsi *et al.* dans [Kob 08] proposent une méthode de stéganalyse basée sur l'utilisation d'un multi-classifieur constitué d'un réseau de neurone artificiel (RNA) et un deuxième classifieur d'analyse discriminant de Fisher (FLD). L'ensemble des caractéristiques, utilisées dans cette méthode de stéganalyse, est construit après la décomposition de l'image en ondelette suivant une représentation multi-résolution.

Avcibas et Memon présentent des stéganalyses sur les schémas LSB basées sur des mesures de qualité de l'image en exploitant l'idée que la distance de l'image marquée à l'image bruitée est plus importante que la distance de l'image source à la même image bruitée ([Avc 01]). Une extension de cette analyse est donnée dans [Avc 02] qui propose d'étudier les variations de certaines corrélations entre variables statistiques existant entre les différents plans de bits (7<sup>ème</sup> et 8<sup>ème</sup> plan de bit). Cette méthode permet d'attaquer des schémas d'insertion utilisant d'autres plans de bit que le dernier.

S. Lyu *et al.* proposent dans ([Lyu 04], [Lyu 06]) une méthode de détection à l'aveugle basée sur des ensembles d'apprentissage et des décompositions des images en ondelettes. Cette décomposition est effectuée via des filtres miroirs en quadrature qui décomposent l'image en sous-

bandes d'orientations et de fréquences différentes: une sous bande horizontale H, verticale V, diagonale D pour chaque niveau de décomposition  $i$ . L'analyse stéganographique est basée sur la prédiction des coefficients  $V_i(x, y)$ ,  $H_i(x, y)$  et  $D_i(x, y)$ . L'analyseur proposé n'est pas capable d'estimer la taille du message inséré et moins efficace que les analyses stéganographiques spécifiques aux schémas d'insertion.

Cette avancée de la stéganalyse a engendré de nouveaux schémas d'insertion dans le domaine fréquentielle. Ces schémas concernent essentiellement le format JPEG.

L'algorithme OutGuess a ainsi été proposé par Provos [Pro 01] pour contrer l'attaque du  $\chi^2$  et procède en deux temps pour insérer le message : insertion selon un ordre aléatoire, et correction afin de rendre l'histogramme de l'image stéganographiée identique à l'image source. L'algorithme F5 pour les images JPEG a été présenté par Westfeld [Wes 01] en 2001. Il est également insensible au test de  $\chi^2$ , il ne procède plus par permutation, mais par décrétement d'un sur le LSB.

Fridrich présente dans [Fri 04] une méthode de stéganalyse basée sur la différence entre un vecteur de caractéristiques extrait de l'image stéganographiée et le même vecteur de caractéristiques obtenu après les trois opérations suivantes: décompression de l'image stéganographiée, découpage de cette dernière par 4 pixels dans les deux directions et la compression de l'image résultante. Cette méthode de stéganalyse utilise un ensemble de caractéristiques, de premier et deuxième ordre, obtenu dans les deux domaines de représentation des images (spatial et fréquentiel).

### 3.4.2 STEGANALYSE SPECIFIQUE

Si les mesures dépendent des algorithmes que nous essayons de détecter, la stéganalyse est dite *spécifique*. Par exemple, J. Barbier *et al.* dans [Bar 06] proposent une stéganalyse dédiée aux algorithmes Outguess [Pro 01], F5 [Wes 01] et JPHide and JPSeek [Lat 99] en mesurant la variation d'entropie binaire d'une image JPEG après avoir stéganographiée successivement plusieurs fois l'image avec le même algorithme.

Fridrich *et al.* dans ([Fri 02a], [Fri 02b], [Gol 03]) proposent une méthode de stéganalyse spécifique aux algorithmes Outguess et F5. Ces attaques reposent sur la modification d'une certaine donnée statistique macroscopique de l'image dans le processus d'insertion.



### 3.5 CONCLUSION

Dans ce chapitre, nous avons exposé les concepts et les techniques de stéganalyse ainsi que leurs objectifs. Généralement, les algorithmes d'analyse stéganographique sont triés selon le type de la stéganalyse (universelle ou spécifique), le domaine d'insertion (spatial ou fréquentielle), et le type d'attaque (active ou passive).

La stéganalyse universelle ou aveugle permet une détection plus large des images stéganographiées. Le point le plus difficile de ces techniques est comment choisir les caractéristiques permettant de différencier une image propre, d'une autre étant stéganographiée. D'autre part, elles sont moins efficaces comparativement à une technique spécifique sur un algorithme déterminé.

Le chapitre suivant présente une méthode de stéganalyse universelle dans le domaine spatial des images numériques basée loi de Zipf. Cette loi permet d'extraire un certain nombre de paramètres qui peuvent caractériser la structure d'une image. Un discriminant linéaire de Fischer (FLD) est utilisé pour décider si une image  $I$  est stéganographiée ou non.

# **CHAPITRE 4**

## **conception et réalisation**

## CHAPITRE 4

### CONCEPTION ET REALISATION

#### 4.1 INTRODUCTION

La stéganalyse est la contre partie de la stéganographie, elle consiste à attaquer les méthodes stéganographiques pour la détection, extraction ou destruction des données cachées dans un support numérique. Une attaque peut être jugée effective simplement lorsqu'elle détecte la présence du message dans le stégo-médium.

La détection de l'information cachée dans les images est un des problèmes les plus difficiles à résoudre. La difficulté vient du fait que, dans le cas de la stéganalyse universelle, il est difficile de déterminer le vecteur caractéristique permettant la distinction entre l'image propre et la stégo-image.

Plusieurs catégories de méthodes existent pour la détection des messages cachées. La plus simple consiste à la détection de signature d'un logiciel donné. Les logiciels utilisés en stéganographie ont des caractéristiques qui leurs sont propres. Ces outils laissent en quelques sortes des signatures après leurs utilisations. L'attaquant sera alerté par ces signatures et pourra alors déceler l'information cachée.

De même, quand la stéganographie utilise le bit de poids faible (LSB), l'attaquant remarque la présence de bruit au niveau des bits les moins significatifs. De plus, cette méthode du LSB a d'autres inconvénients. Par exemple, si l'on utilise des images à fort contraste, l'image comportera des modifications visuelles.

La détection basée sur les déviations statistiques est une autre catégorie d'analyse. elle se base sur le fait que le processus de dissimulation altère les propriétés statistiques originales du cover- médium.

Dans ce chapitre, nous proposons une méthode de stéganalyse basée sur les déviations statistiques en utilisant les propriétés statistiques de la loi de Zipf pour extraire un ensemble de caractéristiques permettant la distinction entre les images propres et les stégo-images.

La loi de Zipf a été utilisée dans des domaines aussi divers que la linguistique ou l'étude du trafic sur internet. Dans le domaine de l'analyse d'images, la loi de Zipf a déjà été utilisée avec succès par Vincent et al. [Vin 00] pour la mesure de la qualité des images compressées et par Caron et al. [Car 03] pour la détection des zones d'intérêt.

Nous proposons une méthode de stéganalyse passive et universelle, c'est-à-dire, les mesures extraites sont indépendantes des algorithmes que nous essayons de détecter.

Nous exposons, dans une première partie, le principe général de la loi de Zipf, ensuite nous expliquons comment cette loi peut être appliquée à l'analyse des images. Dans une deuxième partie, nous utilisons la loi de Zipf pour extraire le vecteur des caractéristiques permettant la distinction entre les images propres et les stégo-images.

## 4.2 PRINCIPE DE LA LOI DE ZIPF

La loi de Zipf est une loi empirique énoncée en 1949 par G.K Zipf [**Zip 49**]. Elle peut s'énoncer comme suit : Dans un ensemble de symboles organisés topologiquement, les n-uplets de symboles ne s'organisent pas de manière aléatoire. On peut constater que les fréquences  $N_1, \dots, N_n$  d'apparition des n-uplets présents  $M_1, \dots, M_n$  sont en relations avec ces motifs. Plus précisément, si l'on classe les motifs suivant l'ordre décroissant des fréquences, la suite  $(N_i, \dots, N_n)$  avec  $i = 1$  à  $n$ , vérifie la formule fondamentale :

$$Fréq_R = K \times Rang^{-\alpha} \quad (4.1)$$

Dans cette formule,  $K$  et  $\alpha$  sont des constantes positives et la valeur de l'exposant  $\alpha$  caractérise la loi puissance. Dans le cas des textes en langage naturel, la valeur de l'exposant  $\alpha$  est proche de 1. Cette distribution en loi puissance peut se représenter graphiquement en échelle bi-logarithmique, avec en abscisse le rang des motifs et en ordonnée leurs fréquences d'apparition [**Car 03**].

### 4.2.1 APPLICATION AUX IMAGES

Une image est un ensemble de pixels organisés dans le plan sous la forme d'une matrice. Les symboles considérés pour la vérification de la loi de Zipf sont donc les niveaux de gris utilisés pour coder les pixels. Les n-uplets sont choisis comme la suite des niveaux des pixels appartenant à des masques susceptibles de prendre des formes variées, par exemple des masques carrés 3x3 ou des masques linéaires verticaux 3x1, 7x1 ou horizontaux, 1x3, 1x7. Ce choix est déterminé par le type de motif que l'on souhaite rechercher : motif linéaire ou motif surfacique. D'autres tailles de motifs seraient possibles, mais une taille plus grande des motifs aurait pour conséquence que chaque motif n'aurait qu'une faible probabilité de se retrouver plusieurs fois dans l'image, et la distribution des fréquences des motifs ne serait pas véritablement significative.

De plus, une taille de motifs trop importante aurait pour conséquence d'augmenter considérablement le temps de calcul. C'est pourquoi une taille des motifs de 3x3 nous a semblé un bon choix pour notre méthode de stéganalyse.

#### 4.2.2 CODAGE DES MOTIFS

Un premier type de codage consiste à partitionner l'échelle des niveaux gris en un nombre réduit de classes. L'échelle des niveaux de gris est divisée en plusieurs intervalles ou classes numérotées de 0 à  $n - 1$  et on affecte à chaque pixel le numéro de la classe qui correspond à son niveau de gris.

Un autre codage possible est celui des rangs généraux, utilisé dans [Bi 96]. La méthode des rangs généraux consiste, à l'intérieur du bloc 3x3, à numéroté les pixels en fonction des niveaux de gris classés dans l'ordre croissant de leur valeur en affectant le même rang quand si les niveaux de gris ont même valeur. On affecte la valeur 0 au niveau de gris le plus bas, et on incrémente la valeur d'une unité quel que soit l'écart relatif entre deux niveaux de gris consécutifs.

La figure 4.1 montre un exemple de l'utilisation des rangs généraux. Cette méthode permet d'analyser les faibles variations locales des niveaux de gris de l'image, ce qui la rend particulièrement adaptée à l'étude de la déviation statistique introduite par l'insertion de l'information.

250	200	200
25	4	29
35	4	35

(a)

5	4	4
1	0	2
3	0	3

(b)

**Figure 4.1: motif original (a), motif codé avec la méthode des rangs généraux (b)**

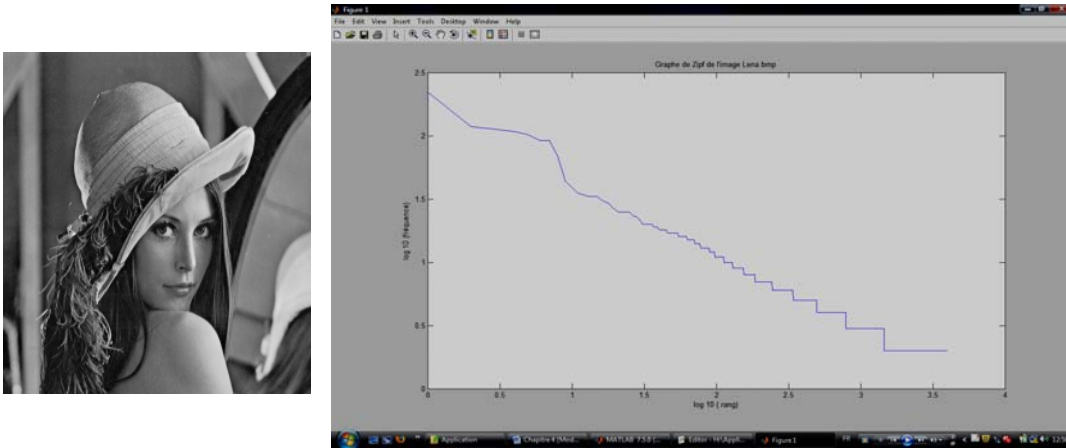
#### 4.2.3 ANALYSER UNE IMAGE AVEC LA LOI DE ZIPF

Pour analyser une image avec la loi de Zipf, on parcourt l'image avec un masque de dimension  $m \times m$ , les motifs rencontrés sont codés en utilisant l'une ou l'autre des méthodes et on classe ces motifs suivant l'ordre des fréquences décroissantes.

Dans notre travail, la taille des motifs a été fixée à 3x3. On trace ensuite la courbe de Zipf dans un repère bi-logarithmique.

La figure 4.2 présente un exemple de courbe de Zipf obtenu à partir d'une image non compressée en utilisant la méthode des rangs généraux pour le codage des motifs.

Sur cette courbe, on a représenté uniquement les fréquences d'apparition des motifs qui apparaissent plus d'une fois dans l'image, on a considéré que les autres motifs n'avaient pas une importance significative.



**Figure 4.2: Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Lena.bmp**

Dans [Mak 99] et [Car 03] les auteurs montrent que l'utilisation de la loi de Zipf peut s'étendre à de nombreux domaines. Cette dernière permet d'extraire un certain nombre de paramètres qui peuvent caractériser la structure d'une image. Ces paramètres sont: les pentes de droite de régression, le nombre de motifs recensés, le nombre des motifs les moins fréquents qui sont souvent les plus discriminants quant au contenu de l'image.

Ils ont montré aussi que la courbe de Zipf d'une image en niveau de gris est généralement divisée en deux parties linéaires. Une partie de la courbe correspond aux zones uniformes de l'image, c'est-à-dire, les régions et une deuxième partie correspond aux zones non uniformes, comme les contours et les détails fins de l'image.

L'ensemble de ces données qui traduit en quelque sorte la marque propre à chaque image représente un outil d'un grand intérêt pour procéder à une discrimination des images entre elles ou pour distinguer des zones au sein d'une même image.

### 4.3 JUSTIFICATION DE CHOIX DE LA STEGANALYSE BASEE LOI DE ZIPF

Nous avons fait le choix d'utiliser la loi Zipf dans notre méthode de stéganalyse des images numériques pour plusieurs raisons:

- ✚ La stéganalyse RS [Fri 01] (étudier dans le chapitre précédent) est basée sur le partitionnement de l'image en un ensemble de motifs horizontaux de dimension  $1 \times 4$ . Ensuite une fonction de discrimination  $f$  permettant d'évaluer l'homogénéité de chacun des motifs  $M$  est défini.
- ✚ Les méthodes proposées dans [Mem 01], [Rou 04], [Dum 03] sont basées sur des égalités statistiques des cardinalités d'ensembles de paires de pixels. Chaque ensemble comprend des motifs formés de deux pixels séparés par une certaine distance  $d$  ( $d > 1$ ) dans une direction particulière  $\square$  ( $\square = 0$ ).
- ✚ Avcibas & al [Avc 02] proposent une méthode basée sur les variations de certaines corrélations entre des motifs existant dans les différents plans de bits ( $7^{\text{ème}}$  et  $8^{\text{ème}}$  plan de bit). Chaque motif est formé de quatre pixels.
- ✚ L'objectif de la loi de Zipf consiste à étudier les statistiques des fréquences d'apparition des motifs (motifs de pixels, Coefficient DCT) dans une image.
- ✚ Elle permet d'analyser les faibles variations des niveaux de gris de l'image, ce qui la rend particulièrement adaptée à la détection des informations cachées.
- ✚ La loi de Zipf permet d'extraire un ensemble de paramètres qui traduit en quelque sorte la marque propre à chaque image [Mak 99].

### 4.4 PRINCIPE DE LA STEGANALYSE PROPOSEE

Notre travail consiste à concevoir une méthode de stéganalyse qui s'appuie sur les déviations statistiques pour détecter les images stéganographiées.

En se basant sur les propriétés statistiques de loi de Zipf pour extraire un ensemble de caractéristiques pertinentes est qui permet de caractériser une image. Le type de l'analyse stéganographique proposé est universelle c'est-à-dire les caractéristiques utilisées pour la détection sont indépendantes des logiciels de stéganographie utilisées pour la dissimulation d'information.

La méthode de stéganalyse proposée se fait en deux étapes principales: d'une part l'extraction des caractéristiques des images pour former l'espace de caractéristiques, ou l'espace de représentation,

et d'autre part le choix d'une méthode de classification ou un classificateur qui assigne à chaque point de l'espace de représentation une probabilité d'appartenance à une classe donnée.

**a) Extraction des caractéristiques :** Il existe un nombre important de caractéristiques que l'on peut extraire à partir des propriétés de la loi de Zipf. L'objectif est de trouver un vecteur de caractéristiques qui sépare les images non stéganographiées et les stégo-images.

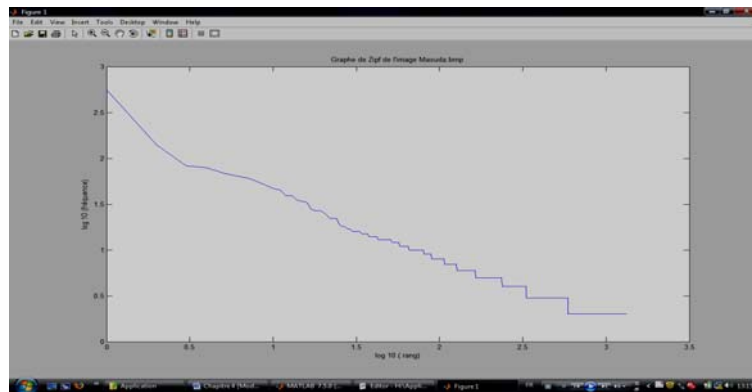
**b) Utilisation d'un discriminant linéaire de FISCHER pour la phase d'entraînement à l'aide de ces caractéristiques.**

#### 4.4.1 EXTRACTION DES CARACTERISTIQUES

L'étape d'extraction des caractéristiques est basée sur les changements statistiques des fréquences d'apparition des motifs (motifs de pixels) dans une image. Le principe se décrit comme suit:

- ✚ Balayer l'image par un masque 3×3, puis coder les motifs par la méthode des rangs généraux et affecter à chaque motif distinct sa fréquence d'apparition dans l'image.
- ✚ Les motifs sont ensuite classés dans l'ordre décroissant de leurs fréquences d'apparition.
- ✚ Représentation graphique dans un repère bi-logarithmique, cette représentation est la courbe de Zipf associée à l'image.
- ✚ Le résultat est un ensemble de points alignés sur une droite dont la pente est égale à  $-\alpha$  selon la formule (4.1).

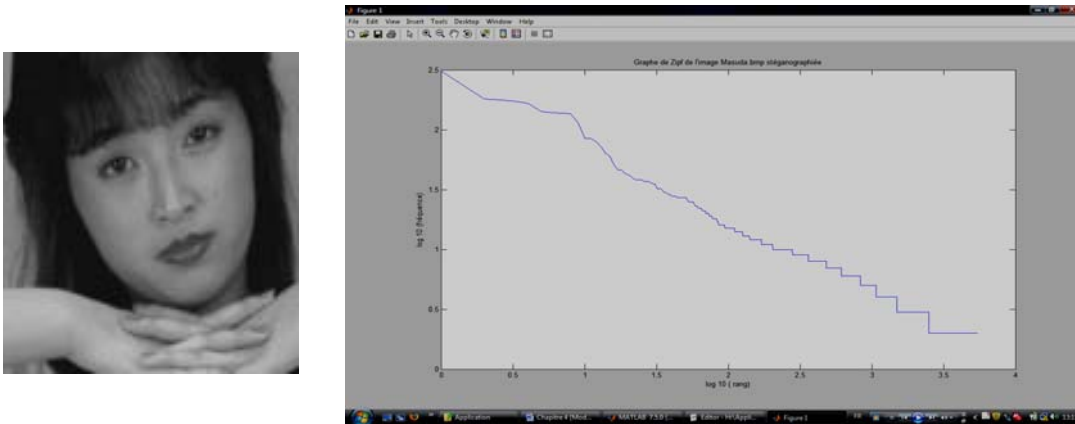
La figure 4.3 présente un exemple de la courbe de Zipf obtenu à partir d'une image non stéganographiée (image Masuda de type Bmp) en utilisant la méthode des rangs généraux pour le codage des motifs.



**Figure 4.3 : Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Masuda non stéganographiée**

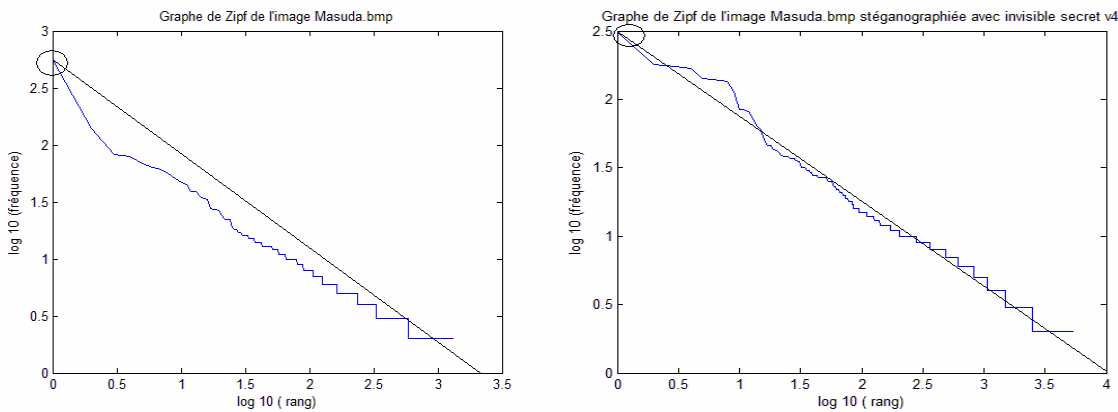


L'insertion d'un message dans le plan LSB se traduit par une répartition différente des motifs de l'image. Ce qui se traduit par une modification de la courbe de Zipf, comme la montre la figure 4.4, ou l'image a été stéganographiée avec *Invisible Secret v4* avec un taux d'insertion égale à 10%.



**Figure 4.4: Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Masuda stéganographiée**

On remarque que les deux courbes des deux figures 4.3 et 4.4 sont différentes surtout dans leur partie gauche. Sur la courbe obtenue avec l'image stéganographiée, l'ordonnée à l'origine est moins élevée, ce qui signifie que la fréquence d'apparition des motifs les plus présents dans l'image stéganographiée a diminué par rapport à l'image originale (figure 4.5). La taille des deux images étant identique, l'image stéganographiée comporte donc plus de motifs différents.



**Figure 4.5 : Différence entre les courbes de Zipf associées à l'image Masuda originale et l'image Masuda stéganographiée**

On va maintenant définir le vecteur de caractéristiques permettant la distinction entre les images propres et les images stéganographiées à partir des propriétés de la loi de Zipf.

#### 4.4.1.1 La pente $\alpha$

La loi de Zipf est caractérisée principalement par la valeur " $\alpha$ " de la puissance. Le moyen le plus facile d'estimer cette valeur " $\alpha$ " consiste à étudier le lien qui existe entre les logarithmes respectifs des fréquences  $Freq$  et des rangs  $R$ . Dans le cas où la loi est vérifiée, les deux grandeurs sont liées par une relation linéaire et la valeur de " $\alpha$ " peut être estimée par le coefficient directeur de la droite de régression approximant, au sens des moindres carrés, les couples  $[\ln(rang), \ln(Freq)]$ . La pente de la droite correspond alors à la valeur de  $\alpha$ . Cette dernière dépend généralement de la qualité de l'alignement du graphe de Zipf, c'est-à-dire l'adéquation de la loi de Zipf aux images.

#### 4.4.1.2 L'entropie

La notion d'entropie peut également être utilisée pour caractériser la complexité d'un ensemble de symboles comme un texte ou une image.

L'entropie au sens de la théorie de l'information a été définie par Caron [Car 03] comme suit: Pour un texte de longueur  $T$  contenant  $R$  mots distincts, l'entropie peut être définie par la formule suivante :

$$(4.2) \sum_{r=1}^R \frac{f(r)}{T} \log_R \frac{f(r)}{T} \quad H_w = -$$

Dans cette formule, on utilise un logarithme à base  $R$ , de manière à permettre de comparer les résultats obtenus avec des textes différents, il permet de donner une valeur de l'entropie comprise entre 0 et 1 quelle que soit la valeur de  $R$ . Cette entropie est maximale quand tous les mots du texte ont la même fréquence d'apparition, elle est minimale quand la fréquence relative d'un des différents mots atteint 1. L'entropie permet de mesurer l'uniformité de la distribution des différents mots. Une autre formulation de l'entropie a été utilisée par Cohen et al. [Coh 97] dans leurs travaux sur les textes.

L'entropie est ici définie relativement à la fréquence d'apparition des différents mots du texte.

L'entropie relative à la fréquence est définie par la formule suivante :

$$(4.3) \sum_{f=1}^F \frac{I(f)}{R} \log_F \frac{I(f)}{R} \quad H_f = -$$

Dans cette formule,  $I(f)$  représente le nombre de mots distincts ayant une fréquence d'apparition égale à  $f$  et  $F$  représente le nombre total d'occurrences des mots dans le texte. L'utilisation d'un logarithme à base  $F$  assure que la valeur de l'entropie soit comprise entre 0 et 1.

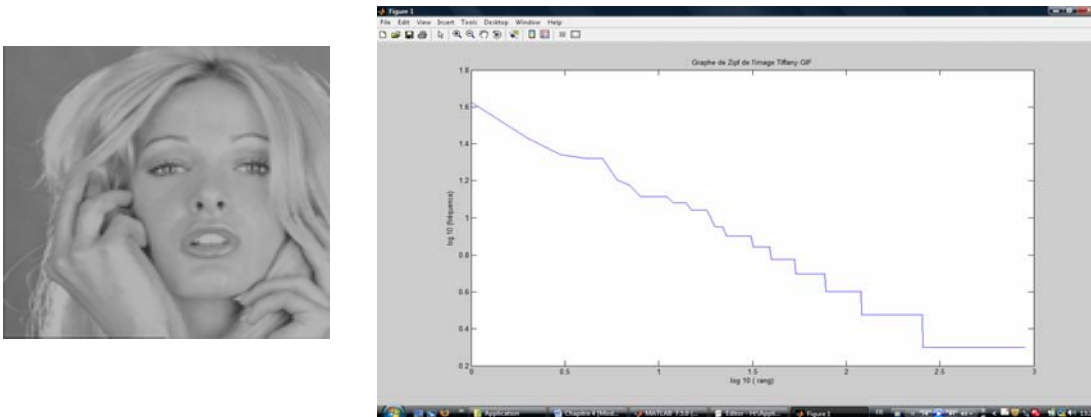
Par rapport à la précédente, cette définition de l'entropie permet de donner un poids plus important aux mots les moins fréquents. Or, ce sont les mots peu fréquents qui sont souvent les plus discriminants quant au contenu du texte.

L'entropie permet de donner une mesure de la quantité de détails présents dans une image, on peut l'utiliser afin de détecter des messages cachés dans une image. Pour détecter des messages cachés avec l'entropie, on calcule l'entropie relative à la fréquence des motifs en utilisant la formule de Cohen rappelée en (4.2) et (4.3).

#### 4.4.1.3 ordonné à l'origine du graphe de Zipf

La modification de la courbe de Zipf introduite par l'insertion de message caché se traduit par une répartition différente des motifs de l'image stéganographiée, ce qui conduit à la modification de la fréquence de motif le plus présent dans l'image.

La Figure 4.6 présente la courbe de Zipf obtenue à partir d'une image non stéganographiée (image Tiffany.Gif) en utilisant la méthode des rangs généraux pour le codage des motifs.



**Figure 4.6 : Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Tiffany non stéganographiée**

L'insertion d'un message caché, par exemple en utilisant le logiciel de stéganographie *Caméléon* (figure 4.7), conduit à une répartition différente des motifs et en particulier la fréquence des motifs le plus présents dans l'image.



Figure 4.7: Logiciel de stéganographie Caméléon 1.0

Par une compilation avec Matlab 7.5, on trouve que l'ordonnée à l'origine de l'image "Tiffany" non stéganographiée est égal à 1.6232 alors que pour l'image "Tiffany" stéganographiée égale à 2.8209 (voir les courbes de Zipf des deux figures 4.6 et 4.8).

Ce qui signifie que la fréquence d'apparition des motifs le plus présent dans l'image a augmenté par rapport à l'image originale.

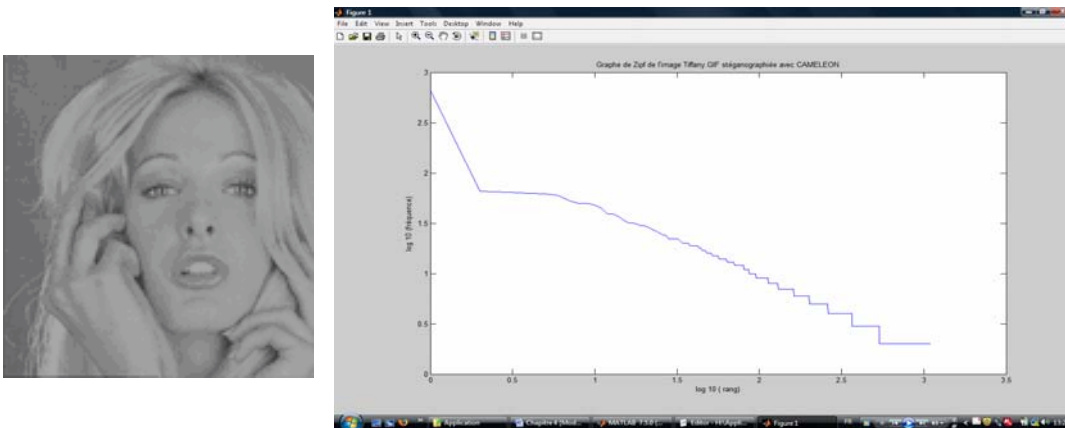


Figure 4.8 : Courbe de Zipf obtenue avec le codage des rangs généraux à partir de l'image Tiffany stéganographiée par Cameleon1.0

#### 4.4.1.4 la qualité Zipf ZQ (Zipf quality)

Avcibas & al [Avc 02] proposent une méthode basée sur les variations de certaines corrélations entre des motifs existant dans les différents plans de bits (7<sup>ème</sup> et 8<sup>ème</sup> plan de bit). Cette méthode permet d'attaquer des schémas d'insertion utilisant d'autres plans de bit que le dernier. Les auteurs rapportent ainsi de bons résultats sur plusieurs logiciels de stéganographie [Lec 03].

On s'inspire de cette méthode pour utiliser un paramètre de qualité qui va mesurer la corrélation entre les deux images de 7<sup>ème</sup> et 8<sup>ème</sup> plan de bit (figure 4.9) à partir des courbes de Zipf. Ce paramètre est appelé  $ZQ_{78}$  (Zipf Quality).

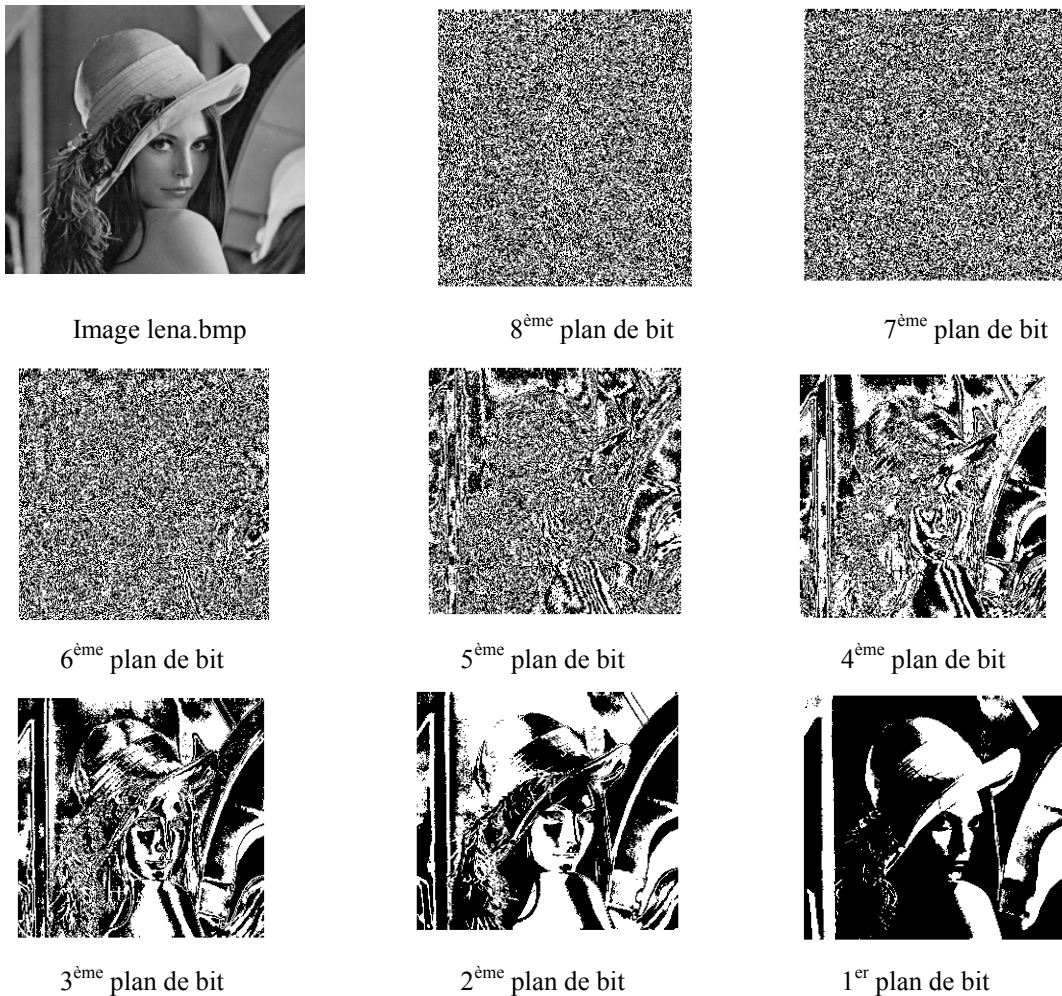


Figure 4.9: Différents plans de bit de l'image Lena

Une formule Pour le calcul du ZQ a été définie par Caron [Car 03] comme de la manière suivante:

$$ZQ = \frac{1}{M} \frac{\sum_{i=1}^M |\log(F_i) - \log(F'_i)|}{\log(F_M)} * \frac{T/L}{T'/L'} * (|P - P'|) * B$$

(4.4)

Dans cette formule :

- M : représente le nombre de motifs considérés pour le calcul de ZQ.
- $F_i$  : représente le nombre de motifs de rang i dans l'image I de 7<sup>ème</sup> plan de bit.
- $F'_i$  : représente le nombre de motifs de rang i dans l'image I' de 8<sup>ème</sup> plan de bit.
- T et T' représentent le nombre de motifs qui apparaissent plus d'une fois dans les images I et I'.
- L et L' sont le nombre total des motifs dans les images I et I', y compris ceux qui n'apparaissent qu'une seule fois.
- P et P' sont les pentes des droites de régression au sens des moindres carrées des courbes de Zipf associées aux images I et I'.
- B est le rapport entre l'ordonnée à l'origine de la courbe associée à l'image stéganographiée et celle associée à l'image originale.

Le paramètre ZQ mesure l'écart entre les courbes associées à l'image I de 7<sup>ème</sup> plan de bit et à l'image I' de 8<sup>ème</sup> plan de bit. Ce paramètre est nul si les deux images sont identiques et il est élevé si la distorsion de l'image I' est importante.

#### 4.4.1.5 Corrélation linéaire

La méthode des moindres carrés consiste à rechercher une droite telle que la somme de ses distances aux différents points représentant les données soit minimale. Pour vérifier l'adéquation de loi de Zipf à une image on calcule le coefficient de corrélation linéaire défini de la manière suivante:

$$r = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}$$

Ce coefficient est compris entre -1 et +1. S'il est voisin en valeur absolue de 1, l'ajustement est valide.

#### 4.4.1.6 Utilisation de l'écart type

Il est également possible d'utiliser l'écart type des niveaux de gris pour détecter des informations cachées dans l'image. L'utilisation de cette mesure commence par la construction d'une image de l'écart type. Pour cela, on balaie l'image par un masque 3x3 et on calcule l'écart type des niveaux de gris des pixels du masque. On associe la valeur de l'écart au pixel situé au centre du masque. Ensuite en analyse l'image avec la loi de Zipf en utilisant le codage des rangs généraux, on trace la courbe de Zipf dans un repère bi-logarithmique et on calcule la pente  $\alpha_\sigma$  et le coefficient de corrélation  $r_\sigma$  de la droite de régression de la courbe obtenue.

#### 4.4.2 EVALUATION DU VECTEUR DE CARACTERISTIQUES

Finalement, le vecteur de caractéristique utilisé pour la discrimination est de dimension 12, les huit premières sont extraites à partir la loi de Zipf (la pente  $\alpha$ , l'entropie  $H_w$ , l'entropie  $H_f$ , la qualité Zipf  $ZQ_{78}$ , l'ordonné à l'origine  $OO$ , le coefficient de corrélation  $r$ , la pente  $\alpha_\sigma$ , le coefficient de corrélation  $r_\sigma$ ).

Dans [Lai 09a, Lai 09b, Lai 09d], les auteurs indiquent comment ces caractéristiques sont altérées lorsqu'un message est dissimulé dans une image.

Les quatre caractéristiques restantes sont les moments centraux normalisés d'ordre 1 à 4 (i.e. la moyenne  $\mu$ , la variance  $\sigma$ , l'asymétrie  $\xi$  et le Kurtosis  $\kappa$ ) utilisés par Kobsi et al. [Kob 08].

Le vecteur de caractéristiques est testé sur quatre images (figure 4.10) qui ont l'avantage de présenter chacune des propriétés différentes quant à leur composition: Singe et Arbre sont des images texturées, Eau présente beaucoup de zones planes et Lena combine les zones planes et les zones texturées.



Figure 4.10 : Images Eau, Lena, Arbre et singe

Les images non compressées ont toutes de la même taille 256\*256 de type BMP, nous avons utilisé un logiciel pour la conversion d'image, nommé CIL [Web 4.1] afin de convertir les images du format BMP aux formats GIF et JPEG. La figure 4.11 illustre la conversion des quatre images.

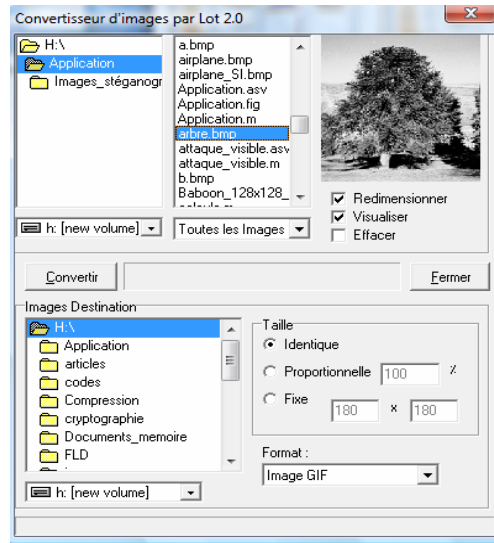


Figure 4.11: Conversion des images Arbre, Lena, Singe et Eau.

Il existe un grand nombre de logiciels utilisant les techniques stéganographiques. Ces programmes sont plus ou moins bien faits et principalement créés par des personnes individuelles, intéressées par la stéganographie.

Dans [Lai 09d] les auteurs montrent l'efficacité du vecteur de caractéristiques pour la discrimination entre les images propres et les images stéganographiées avec les algorithmes de stéganographie : Outguess et F5.

Nous avons utilisé trois algorithmes de dissimulation: *Invisible secret v4.0* [Web 4.2], *Caméléon VI.0* [Web 4.3] et *JPHSwin v0.5* [Web 4.4]. Le choix de ces algorithmes de dissimulation est justifié par la préservation des caractéristiques du premier ordre (*JPHSwinv0.5*) [Avc 02] et permettent d'utiliser différents formats d'images (BMP, GIF et JPEG).

#### **Caméléon v1.0**

Caméléon est un logiciel français qui cache du texte dans des fichiers GIF et offre une possibilité d'encryption (figure 4.12). Il a une interface conviviale et une aide utilisateur très complète.



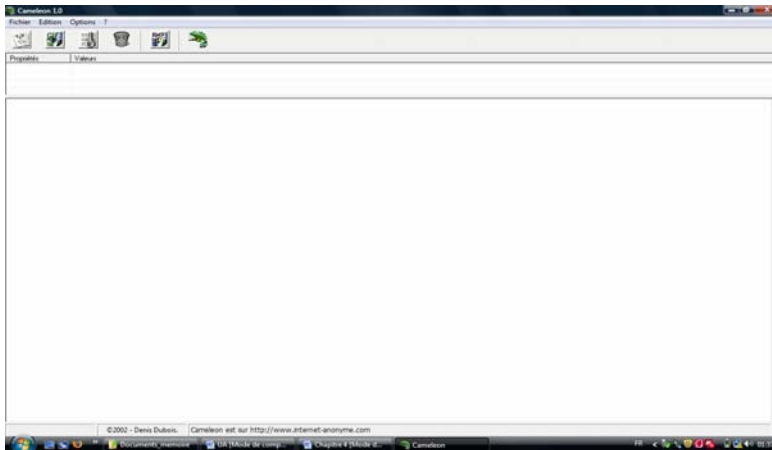


Figure 4.12: Logiciel de stéganographie Cameleon 1.0

**Invisible Secret v4.0**

Invisible Secret V4.0 comporte une fonctionnalité supplémentaire : le transfert automatique de l'image caché via Internet ou e-mail (figure 4.13). Il est sans conteste le logiciel le plus complet et le plus axé utilisateur.



Figure 4.13: Logiciel de stéganographie Invisible secrets 4.0

### ✚ JPHSwinv0.5 (JPHide and JPSeek)

JPHide and JPSeek est un logiciel de stéganographie implémenté par A. Latham [Lat 99] en 1999 selon deux versions, 0.3 et 0.5. La version 0.5 (figure 4.14) intègre en plus un algorithme de compression du message à dissimuler dans les images JPEG [Bar 07].

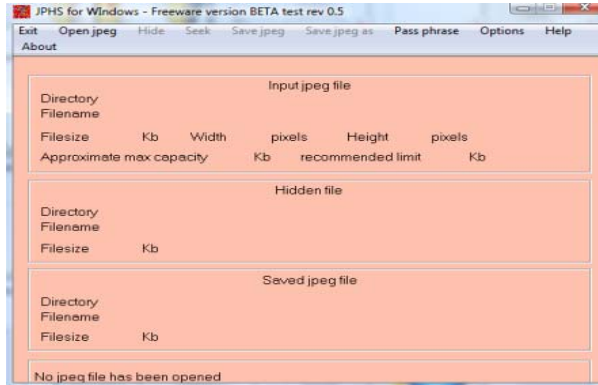


Figure 4.14: Logiciel de stéganographie JPHS pour windows 5.0

En appliquant ces trois algorithmes de dissimulation, chacune des images propres de la figure 4.9 génère trois stégo-images. De ce fait le vecteur des caractéristiques sera testé sur 16 images (propres + stéganographiées).

Ce travail a été développé avec l'outil MATLAB 7.5 (figure 4.15) afin de montrer, dans un premier temps, les déviations statistiques du vecteur de caractéristiques proposé après la dissimulation d'un message dans une image.

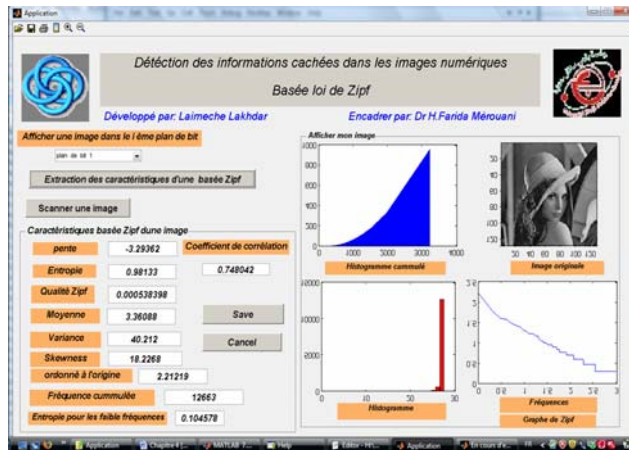


Figure 4.15: Extraction des caractéristiques

La figure 4.16 montre qu'il est quasiment impossible de détecter la dissimulation à l'œil nu. Les figures 4.16.a, 4.16.b et 4.16.c représentent les images propres (non stéganographiées) dans différents formats (BMP, GIF et JPEG), tandis que les figures 4.16.a', 4.16.b' et 4.16.c' sont générés par les algorithmes de dissimulation : Invisible Secret V4, JPHS et Cameleon.

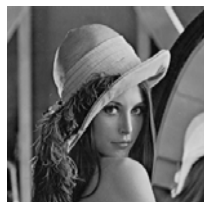


Image Lena.JPEG (a)



Image stéganographiée avec JPHS (a')



Image Singe.BMP (b)



Image stéganographiée avec Invisible Secret (b')



Image Eau.gif (c)



Image stéganographiée avec Cameleon (c')

**Figure 4.16: Dissimulation d'information dans les images Lena.JPEG, Singe.BMP et Eau.GIF**

Le tableau suivant montre les déviations de caractéristiques statistiques après la dissimulation des données au sein des images Lena.jpeg, Singe.bmp et Eau.gif, en utilisant *Invisible secret V4* pour le format Bmp, *Cameleon* pour le format Gif et *JPHS* pour le format Jpeg.

	Lena.jpeg propre	Lena.jpeg stégo	Singe.bmp propre	Singe.bmp stégo	Eau.gif Propre	Eau.gif stégo
<b>Pente <math>\alpha</math></b>	1.4844	1.41823	1.51585	1.63724	1.11479	1.28672
<b>Entropie <math>H_w</math></b>	0.98380	0.970667	0.939157	0.915551	0.141057	0.622414
<b>Entropie <math>H_f</math></b>	0.09538	0.199993	0.21872	0.278472	0.69482	0.301072
<b>Ordonné à L'origine (OO)</b>	2.34342	2.95999	3.0569	3.03663	4.75881	4.28587
<b>Qualité de Zipf (ZQ)</b>	$1.85^E-5$	0.004435	0.0002615	0.0007125	0.0008427	0.302765
<b>Coefficient de Corrélation (r)</b>	0.74168	0.590938	0.79029	0.917	0.652364	0.545361
<b>Pente de l'image écart type <math>\alpha_\sigma</math></b>	2.70276	2.60913	2.74705	2.91679	2.17062	2.42106
<b>Coefficient de Corrélation de l'image Ecart type (<math>r_\sigma</math>)</b>	0.842308	0.645301	0.905588	0.998401	0.72621	0.58587
<b>Variance <math>\sigma</math></b>	36.9582	65.6459	24.4039	29.8415	1.3952	40.8849
<b>Moyenne <math>\mu</math></b>	3.35344	3.83883	5.4729	5.42588	27.3	67.4662
<b>Skewness <math>\xi</math></b>	3.10679	5.54789	4.21414	3.93177	2.7618	3.85736
<b>Kurtosis <math>\kappa</math></b>	16.271	52.2853	30.8408	26.821	10.8123	21.168

**Tab.4.1 : Vecteur de caractéristiques extrait des images propres et stégo-images**

Le tableau 4.1 montre que la valeur du  $ZQ_{78}$  entre le 7<sup>ème</sup> et le 8<sup>ème</sup> plan de bit de l'image Lena non stéganographiée est tend vers 0, ce que signifie que les deux images binaires sont identiques. Par contre la différence entre les deux images du 7<sup>ème</sup> et le 8<sup>ème</sup> de l'image Lena stéganographiée avec *JPHS* est augmentée, ce que signifie que l'insertion d'un message dans l'image Lena.jpeg conduit à la diminution de la corrélation entre les plans de bit.

D'une autre coté, on remarque que l'entropie des motifs les moins fréquents pour l'image lena.jpeg stéganographiée a augmenté, ce que signifie que le nombre des fréquences d'apparition des motifs les moins fréquents a augmenté. Le tableau 4.1 montre que la fréquence d'apparition maximale de

l'image Lena non stéganographiée est égale à **2.34342** alors que pour l'image Lena stéganographiée égale **2.95999**.

Le coefficient de corrélation  $r$  pour l'image Lena non stéganographiée montre que l'ajustement linéaire des fréquences d'apparition des motifs est valide ( $0.7 < 0.74168 < 1$ )  $\Rightarrow$  l'ajustement linéaire est valide [Fou 87]), par contre l'ajustement linéaire des fréquences d'apparition des motifs n'est pas valide pour l'image stéganographiée ( $0.590938 < 0.7$ ).

#### 4.4.3 METHODE DE CLASSIFICATION

Une fois l'espace de caractéristiques fixé, il s'agit maintenant de choisir une méthode de classification ou un classifieur qui, après une certaine phase d'apprentissage, assigne à chaque point de l'espace de représentation une probabilité d'appartenance à une classe donnée.

Nous rappelons que le terme classifieur linéaire représente une famille d'algorithmes de classement statistique. Le rôle d'un classifieur est de classer dans des groupes (des classes) les échantillons qui ont des propriétés similaires, mesurées sur des observations. Un classifieur linéaire est un type particulier de classifieur, qui calcule la décision par combinaison linéaire des échantillons.

Pour un vecteur de caractéristiques, la sortie d'un classifieur est donnée par:

$$g(x) = f(w^T x + w_0) = f\left(\sum_{j=1}^N w_j x_j + w_0\right)$$

Où  $w$  est un vecteur de poids,  $w_0$  est le biais, et  $f$  est une fonction qui convertit le produit scalaire des deux vecteurs dans la sortie désirée. La fonction  $f$  est souvent une simple fonction de seuillage, par exemple la fonction signe ou des fonctions plus complexes comme la tangente hyperbolique, où la fonction sigmoïde.

L'estimation des paramètres  $w$  est basée sur la minimisation d'un critère  $J$ , le plus souvent fonction des erreurs de classification. Il existe plusieurs algorithmes d'apprentissage (d'estimation des paramètres) : le perceptron, les SVM, ...

On trouve aussi des méthodes qui maximisent un critère  $J$  (maximiser la variance inter classe) comme l'analyse discriminant linéaire de Fischer.

#### 4.4.3.1 Mesures de performance du classifieur

Les performances d'un classifieur pour la discrimination d'exemples appartenant à une classe  $C$  de ceux qui appartiennent à la classe complémentaire  $C'$  sont estimées par 4 valeurs :

- ✚ VP : (Vrai Positif) est le nombre d'exemples de la classe  $C$  classés comme appartenant à la classe  $C$  par le classifieur, (i.e. bien classés dans la classe  $C$ )
- ✚ VN : (Vrai Négatif) est le nombre d'exemples de la classe  $C'$  classés comme n'appartenant pas à la classe  $C$  par le classifieur, (i.e. bien classés dans la classe  $C'$ )
- ✚ FP : (Faux Positif) est le nombre d'exemples de la classe  $C'$  classés comme appartenant à la classe  $C$  par le classifieur, (i.e. mal classés dans la classe  $C$ )
- ✚ FN : (Faux Négatif) est le nombre d'exemples de la classe  $C$  classés comme appartenant à la classe  $C'$  par le classifieur, (i.e. mal classés dans la classe  $C'$ ).

Supprimé : c

À partir de ces 4 mesures, on définit habituellement trois quantités :

- **la sensibilité du classifieur ( $S_e$ )** : Probabilité d'avoir un test positif quand on est des images stéganographiées:

$$S_e = \frac{V_p}{V_p + F_p}$$

- **spécificité ( $S_p$ )** : Probabilité d'avoir un test négatif quand on est des images propres:

$$S_p = \frac{V_N}{V_N + F_N}$$

- **Probabilité de succès**: C'est la probabilité des vrais positifs et des vrais négatifs par rapport à la totalité des images propres et stéganographiées.

Un bon classifieur est donc un classifieur dont la sensibilité et la prédiction positive sont élevées.

#### 4.4.3.2 Justification de choix de classifieur

Nous avons fait le choix d'utiliser un Discriminant linéaire de Fischer (FLD) pour plusieurs raisons:

- ✚ Le classifieur FLD est plus adapté pour le processus de classification avec nombreuses données manquantes et imprécises (e.g: classification d'une image avec peu de données d'apprentissage).

- ✚ L'utilisation d'un FLD est justifiée par sa grande robustesse lorsque l'hypothèse de normalité n'est pas trop vérifiée. L'hypothèse suppose que les attributs possèdent une certaine capacité à discriminer les classes. [Web 4.5].
- ✚ La dimension de vecteur de caractéristique n'a pas une influence sur le FLD [Web 4.6].
- ✚ La vitesse du classifieur FLD qui désigne à la fois le temps nécessaire pour la construction du classifieur et pour classer une image.

#### 4.4.3.3 Discriminant Linéaire de Fischer

Etant données  $k$  classes distinctes d'observations, l'objectif de l'analyse discriminante est d'affecter à une nouvelle observation l'une des classes. Dans le cadre de la stéganographie,  $k = 2$  et les classes sont la classe des images de couverture ( $C1$ ) et la classe des stégo images ( $C2$ ). Une observation est la donnée d'un médium. L'analyse se décompose en deux étapes. La première consiste à répartir  $n$  observations dont on connaît les classes. La seconde, consiste à affecter une nouvelle observation à une classe.

L'affectation est alors équivalente à une hypothèse sur l'appartenance de l'observation à une classe. Plus précisément, à chaque observation sont associées  $p$  variables explicatives  $V_i$ , qui permettent de décrire une observation.

Chaque classe est représentée par un nuage  $E_i$  de  $\mathbb{R}^p$  (figure 4.17), composé de  $n_i$  individus ( $e_i^j$ ),  $i=1 \dots n_i$  de coordonnées  $(V_1(e_i^j), \dots, V_p(e_i^j))$ .

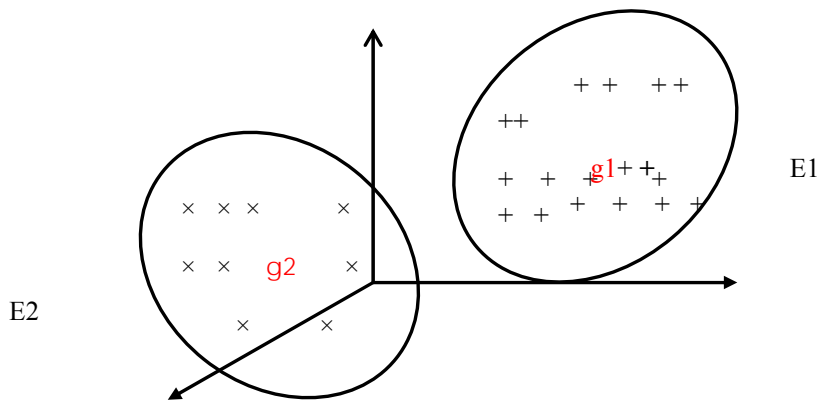


Figure 4.17: Représentation des nuages pour  $p = 3$  et  $k = 2$

On définit la fonction linéaire discriminante  $f$  de la manière suivante:

$$f(x) = f(w^T x + w_0) \quad (4.5)$$

Le vecteur  $W$  des paramètres optimaux est celui qui maximise le critère:

$$J(w) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \quad (4.6)$$

Où  $\mu_j$  est la moyenne des observations de la classe  $E_j$  après projection par  $f$  :

$$\mu_j = \frac{1}{n_{E_j}} \sum_{x_i \in E_j} f(x_i) \quad (4.7)$$

et  $\sigma_j^2$  est la variance définie comme suit:

$$\sigma_j^2 = \frac{1}{n_{E_j}} \sum_{x_i \in E_j} (f(x_i) - \mu_j)^2 \quad (4.8)$$

A partir de l'ensemble des exemples on définit la moyenne et la matrice de variance-covariance pour chaque classe.

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_i \quad \text{Pour } x_i \in E_j \quad (4.9)$$

$$V_j^i = \frac{1}{n_j} \sum_{x_i \in E_j} (x_i - \mu_j)(x_i - \mu_j)^t \quad (4.10)$$

Les expressions de (4.8) et (4.9) deviennent:

$$\mu_j = \frac{1}{n_j} \sum_{x_i \in E_j} (w^t x_i + w_0) = w^t \mu_j + w_0 \quad (4.11)$$

$$\begin{aligned} \sigma_j^2 &= \frac{1}{n_j} \sum_{x_i \in E_j} (w^t x_i - w^t \mu_j)(w^t x_i - w^t \mu_j) \\ &= w^t \frac{1}{n_{E_j}} \sum_{x_i \in E_j} (x_i - \mu_j)(x_i - \mu_j)^t w \\ \sigma_j^2 &= w^t V_j^i w \end{aligned} \quad (4.12)$$



En substituant (4.11) et (4.12) dans (4.5), on obtient

$$J(w) = \frac{(w^t \mu_1 - w^t \mu_2)^2}{w^t (V_1^i)^2 w + w^t (V_2^i)^2 w} \quad (4.13)$$

On définit la matrice  $V_{Int}^i$  de variance-covariance intra-classe et la matrice  $V_{Ext}^i$  de variance-covariance inter-classe par :

$$V_{Int}^i = V_1^i + V_2^i$$

$$V_{Ext}^i = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t$$

La formule (4.13) devient

$$J(w) = \frac{w^t V_{Ext}^i w}{w^t V_{Int}^i w} \quad (4.14)$$

Le vecteur  $w$  optimum est celui qui maximise le rapport (4.14), c'est-à-dire qui maximise la distance entre les centres des classes sur le nouvel axe et minimise leurs variances.

Ce rapport est maximisé quand **[Ben 04]** :

$$V_{Ext}^i w = \lambda V_{Int}^i w$$

Et alors

$$(V_{Int}^i)^{-1} V_{Ext}^i w = \lambda w$$

La solution optimale obtenue à partir des vecteurs propres de la matrice  $(V_{Int}^i)^{-1} V_{Ext}^i$  est donnée par:

$$a = (V_{Int}^i)^{-1} (\mu_1 - \mu_2) \quad (4.15)$$

$w$  est appelé fonction de Fischer et  $(V_{Int}^i)^{-1} (\mu_1 - \mu_2)$  représente l'axe qui sépare le mieux par projection les deux classes.

Nous nous intéressons maintenant au classement d'un nouvel individu  $e$  dont on connaît les variables explicatives  $V^i$  mais pas à la catégorie à laquelle il appartient. Nous allons donc émettre

une hypothèse sur son appartenance à l'une des deux classes. Pour cela, nous projetons  $e$  sur "a" et regardons s'il est plus près de  $\mu_1$  ou  $\mu_2$ .

La règle de Mahalanobis-Fisher consiste à affecter à  $e$  la classe C1 si

$$e^t (V_{Int}^i)^{-1} (\mu_1 - \mu_2) > \frac{1}{2} (\mu_1 + \mu_2) (V_{Int}^i)^{-1} (\mu_1 - \mu_2) \quad (4.16)$$

et la classe C2 sinon.

Pour le biais  $w_0$ , plusieurs formules sont disponibles :

$$w_0 = \frac{\mu_1 + \mu_2}{2}$$

$$w_0 = \frac{n_1 \mu_1 + n_2 \mu_2}{n_1 + n_2}$$

#### 4.5 TESTES ET RESULTATS

Pour valider notre méthode de stéganalyse, il serait nécessaire de comparer les performances de notre discriminant linéaire de Fischer avec une méthode de stéganalyse de même type (universelle) et dans le même domaine de représentation des images (domaine spatial).

Pour ce faire, nous avons récupéré du [Web 4.7], la même base d'images utilisée par Avcibas & al. [Avc 02].

Cette base est très variée, prises dans divers thèmes, elle contient des images de nature, des portraits, objets synthétiques et d'autre image (figure 4.18).

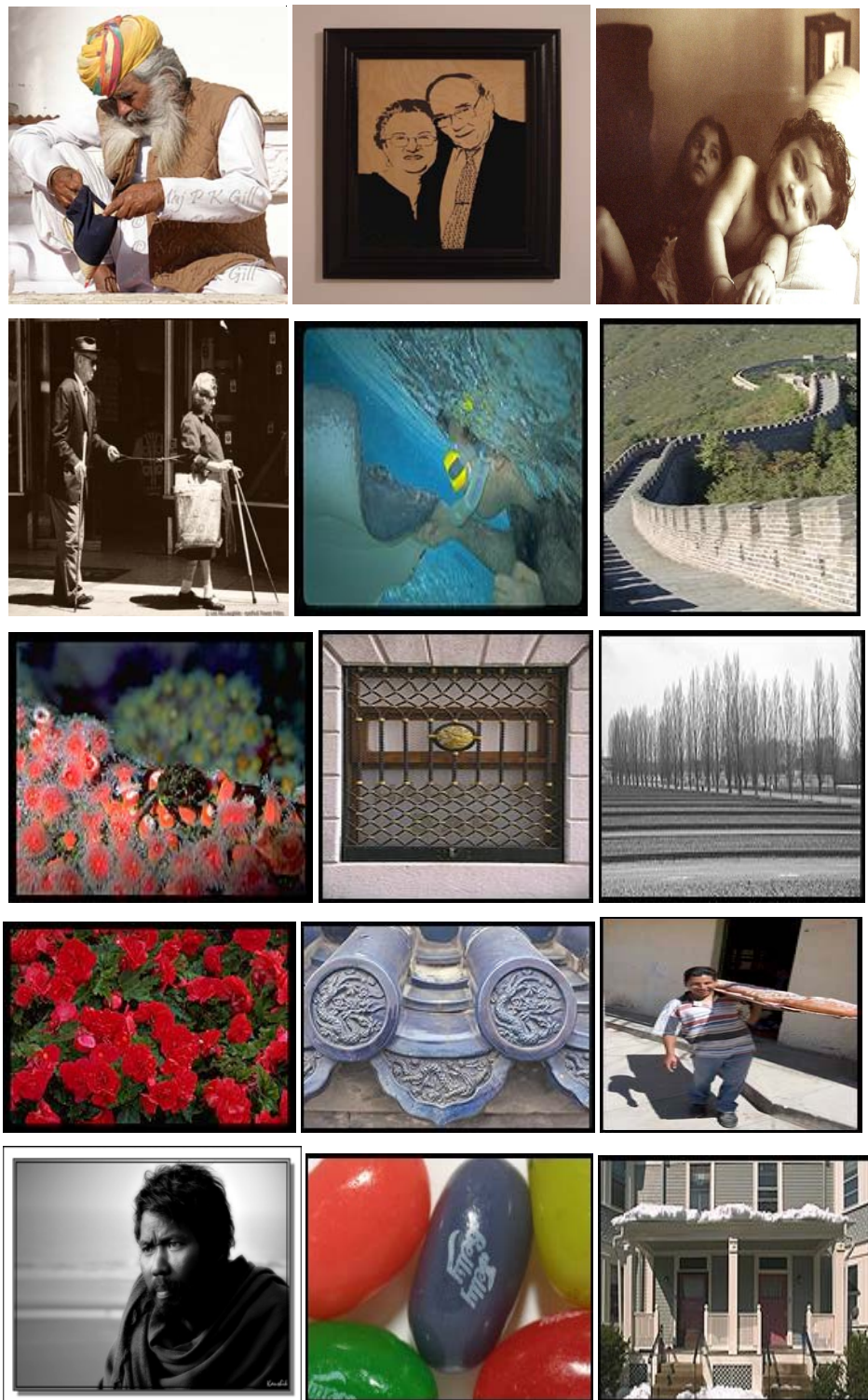


Figure 4.18: Images extraites de la base Philip

Détection des informations cachées dans les images fixes basée loi de Zipf

Dans la phase d'apprentissage, nous avons choisi 350 images au format BMP de la base d'images [Web 4.7] et 350 autres dans la même base au format JPEG. Chaque format d'images est stéganographié respectivement avec *Invisible Secret v4* et *JPHSwin* pour un taux stéganographique de 10 et 15%.

Ensuite, nous avons associé à chaque image  $I$  un vecteur statistique  $V(I)$  de douze coordonnées, tel que:

$$I \rightarrow V(I) = (\alpha, H_w, H_f, ZQ, OO, r, \alpha_\alpha, r_\alpha, \mu, \sigma, \zeta, \kappa) \quad (4.17)$$

Ce vecteur statistique est le point central de notre stéganalyse.

Enfin, nous avons obtenus la moyenne de chaque classe (propre et stéganographiée) et le vecteur discriminant présenté dans le tableau 4.2 ainsi le seuil  $T$  sera égale à **8.0764**.

La moyenne $\mu_1$	La moyenne $\mu_2$	Le vecteur discriminant a
1.6538	1.70044	-0.2051
0.9484	0.9873	3.0077
0.0002	0.0147	-12.7244
3.9582	3.2873	0.0706
44.2398	11.6488	0.0760
5.0084	2.3710	1.6175
2.8925	1.9281	0.7390
0.8008	0.7325	-9.0998
0.2022	0.1778	12.2129
0.711	0.6586	9.7525
2.9563	3.0111	-0.7531
29.2076	15.0295	-0.0029

**Tab. 4.2 : Calcule des paramètres du classifieur**

Dans la phase de teste, nous avons choisi aléatoirement 300 images aux formats BMP et JPEG stéganographiées respectivement avec *Invisible Secret v4* et *JPHSwin* pour un taux stéganographique de 10 et 15%.

Nous les avons ensuite soumis à notre classificateur. Les performances du détecteur pour un taux stéganographique de 10% sont représentées dans la table de confusion suivante:

	Stégo-image	Image propre	Total
Stégo-image	263	37	300
Image propre	64	236	300
Total	327	273	600

**Tab.4.3: Matrice de confusion pour un taux stéganographique de 10%**

A partir de la table de confusion (tab.4.3) on calcule la sensibilité, la spécificité et la probabilité de succès:

La sensibilité = **0.80**

La spécificité = **0.86**

Probabilité de succès = **0.83**

Les performances du détecteur pour un taux stéganographique de 15% sont représentées dans la table de confusion suivante:

	Stégo-image	Image propre	Total
Stégo-image	277	23	300
Image propre	21	279	300
Total	298	302	600

**Tab.4.4: Matrice de confusion pour un taux stéganographique de 15%**

A partir de la table de confusion (tab.4.4) on calcule la sensibilité, la spécificité et la probabilité de succès

La sensibilité = **0.93**

La spécificité = **0.92**

Probabilité de succès = **0.92**

Nous comparons les résultats de notre méthode de stéganalyse avec les résultats de la stéganalyse proposée par Avcibas & al. [Avc 02]. La stéganalyse de [Avc 02] donne de meilleurs résultats que la notre. Cependant, la stéganalyse basée Zipf fournit des résultats meilleurs que ceux obtenus par la méthode proposée par [Avc 02].

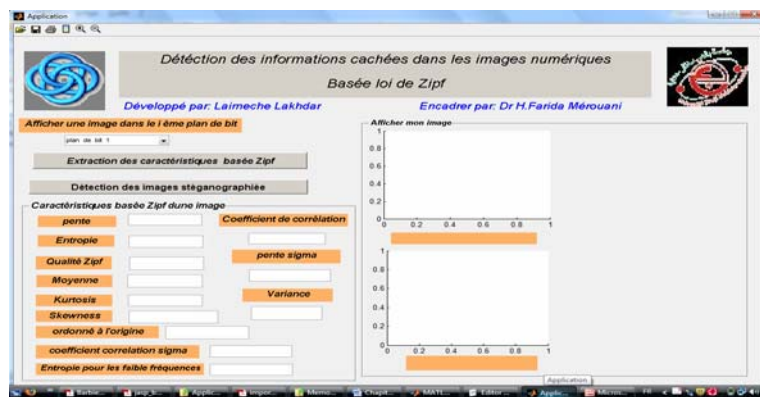
Probabilité de succès	Stéganalyse basée Zipf	Stéganalyse basée mesure de similarité binaire
10%	0.83	85.61
15%	0.92	91.06

**Tab.4.5: Probabilités de succès de la stéganalyse basée Zipf et la Stéganalyse basée mesure de similarité binaire**

#### 4.6 UTILISATION DU DETECTEUR\_ZIPF

Le *détecteur\_Zipf* que nous avons développé est destiné à mettre en application les notions acquises précédemment. Ce dernier détecte les informations cachées dans une image au format Bmp. Une seconde fonctionnalité permet, à partir d'une image (non stéganographiée ou stéganographiée), de trouver le vecteur de caractéristique et d'afficher une image selon les différents plans de bits. La méthode employée pour l'extraction des caractéristiques et la détection des images stéganographiée est la loi de Zipf

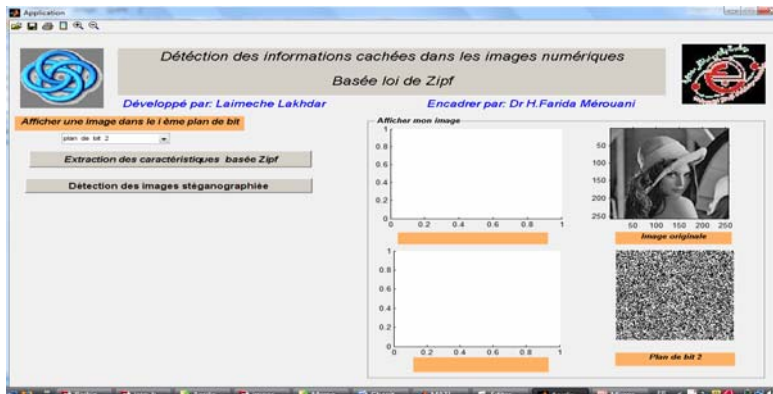
##### 4.6.1 Fenêtre principale



Après avoir lancé le *détecteur\_Zipf*, la fenêtre principale apparaît offrant trois choix:

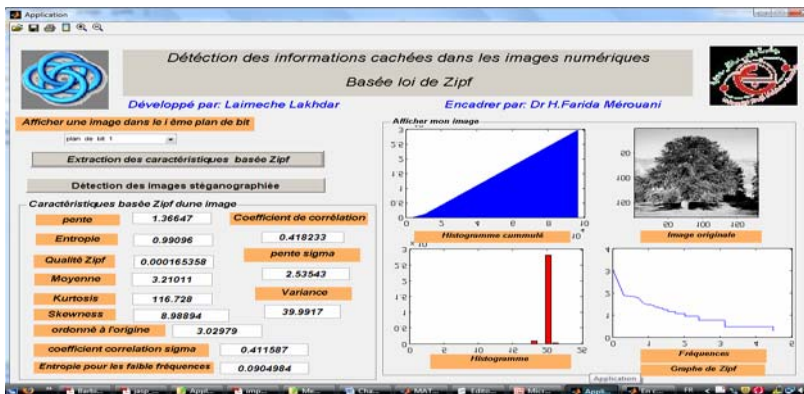
- ✚ Afficher une image dans le  $i^{\text{ème}}$  plan de bit.
- ✚ Extraction des caractéristiques basée loi de Zipf.
- ✚ Détection des images stéganographiées

#### 4.6.2 Afficher une image dans le $i^{\text{ème}}$ plan de bit



Pour afficher le  $i^{\text{ème}}$  plan de bit d'une image, sélectionner l'image source. Une représentation visuelle de cette image permet de constater l'attaque visuelle.

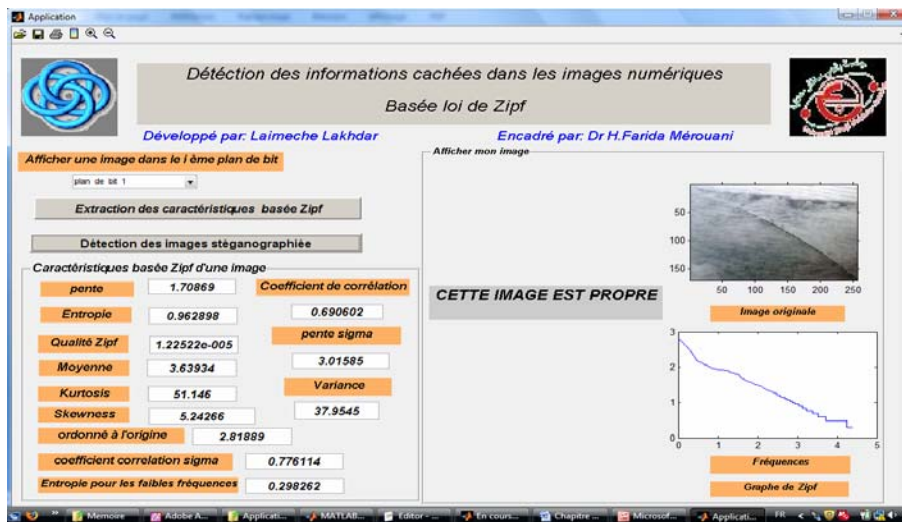
#### 4.6.3 Extraction des caractéristiques basée loi de Zipf



Pour extraire les caractéristiques d'une image, sélectionner l'image source souhaitée, une vue est ouverte, vous permettant de vérifier le vecteur de caractéristiques basée Zipf, la courbe de Zipf, l'histogramme cumulé et l'histogramme des fréquences d'apparition des motifs présent dans l'image.

#### 4.6.4 Détection des images stéganographiées

Pour vérifier si une image est stéganographiée, sélectionner l'image source souhaitée, une vue est ouverte, vous permettant de vérifier le vecteur de caractéristiques basée Zipf, la courbe de Zipf et la nature de cette image.



#### 4.7 CONCLUSION

Dans ce chapitre, nous avons présenté une méthode pour la détection des informations cachées dans les images numériques. Le processus de dissimulation dans les images cause une déviation dans les propriétés statistiques de l'image. En se basant sur ce principe, une méthode de stéganalyse universelle dans le domaine spatial est proposée, utilisant les propriétés statistiques de la loi de Zipf pour extraire un ensemble de caractéristiques permettant la distinction entre les images propres et les stégo-images.

La méthode de stéganalyse proposée se fait en deux étapes principales: d'une part l'extraction des caractéristiques des images pour former l'espace de caractéristiques et d'autre part l'utilisation d'un discriminant de Fischer.



Pour valider notre méthode de stéganalyse, nous avons récupéré la même base d'images utilisée par Avcibas & al [Avc 02]. Dans la phase d'apprentissage, nous avons choisis 350 images au format BMP et 350 autres dans la même base au format JPEG. Chaque format d'images est stéganographié respectivement avec *Invisible Secret v4* et *JPHSwin* pour un taux stéganographique de 10 et 15%.

Dans la phase de teste, nous avons choisi aléatoirement 300 images au format BMP et 300 autres stéganographiées respectivement avec *Invisible Secret v4* et *JPHSwin* pour un taux stéganographique de 10 et 15%.

Nous avons comparé les résultats de notre méthode de stéganalyse avec les résultats de la stéganalyse proposée par Avcibas & al [Avc 02]. La stéganalyse de [Avc 02] donne de meilleurs résultats que la notre. Cependant, la stéganalyse basée Zipf fournit des résultats meilleurs que ceux obtenus par la méthode proposée par [Avc 02].

## CONCLUSION ET PERSEPECTIVES

Le travail présenté dans ce manuscrit s'inscrit dans le cadre de l'insécurité de la stéganographie et plus précisément la détection des informations cachées dans les images fixes.

Dans le cadre de notre travail, nous avons proposé une méthode de stéganalyse universelle dans le domaine spatial basée loi de Zipf. Cette dernière consiste à détecter la présence d'un message caché dans une image, seul cette présence est déterminée.

Les méthodes de stéganalyse tirent profit du fait que l'insertion des données cachées altère les propriétés statistiques de l'image originale. En se basant sur ce principe, nous avons utilisé la loi de Zipf pour extraire un ensemble de caractéristiques pertinentes qui permet de caractériser une image.

Le choix de la loi de Zipf n'est pas arbitraire, elle constitue le fruit d'une étude comparative entre plusieurs méthodes de stéganalyse. En général, la loi de Zipf permet de: (1) étudier les statistiques des fréquences d'apparition des motifs (motifs de pixels) dans une image, (2) analyser les faibles variations des niveaux de gris de l'image, ce qui la rend particulièrement adaptable à la détection des informations cachées, (3) extraire un ensemble de paramètres qui traduit en quelque sorte la marque propre de chaque image.

La méthode de stéganalyse proposée se fait en deux étapes: d'une part l'extraction d'un ensemble de caractéristiques en utilisant la loi de Zipf, c'est avec ces caractéristiques qu'il serait possible de discriminer les deux classes d'images (images propres et stégo-images) et d'autre part l'utilisation d'un discriminant linéaire de Fischer (DLF) qui permet le classement d'une nouvelle image  $I$  dont on connaît les variables explicatives à l'une des deux classes (images propres et stégo-images).

L'affectation d'une image  $I$  à l'une des deux classes est basée sur la règle de Mahalanobis-Fisher et la comparaison entre la fonction de Fischer à une valeur de seuil  $T$ .

Pour valider notre méthode de stéganalyse, nous avons récupéré la même base d'images utilisée par Avcibas & al [Avc 02]. Dans la phase d'apprentissage, nous avons choisis 350 images au format BMP et 350 autres dans la même base au format JPEG. Chaque format d'images est stéganographiées respectivement avec *Invisible Secret v4* et *JPHSwin* pour un taux stéganographique de 10 et 15%.

Dans la phase de teste, nous avons choisi aléatoirement 300 images au format BMP et 300 autres stéganographiées respectivement avec *Invisible Secret v4* et *JPHSwin* pour un taux stéganographique de 10 et 15%.

Comparons les résultats de notre méthode de stéganalyse avec les résultats de la stéganalyse proposée par Avcibas & al [Avc 02], on remarque que lorsque le taux stéganographique est très faible ( $\leq 10\%$ ), la stéganalyse proposée par Avcibas & al [Avc 02]. donne des bonnes résultats que la méthode de stéganalyse proposée.

La stéganalyse basée Zipf fournit des résultats meilleurs par rapport à la stéganalyse proposée par Avcibas & al. pour des taux supérieurs à 15%.

## PERSPECTIVES

Notre travaille reste ouvert et extensible. Nous prévoyons plusieurs extensions à ce travaille, dont les principales sont les suivantes:

- ✚ Notre travaille ne s'applique que dans le domaine spatial. Nous proposons de compléter le travaille pour qu'il s'applique dans le domaine fréquentielle (DCT, DWT).
- ✚ Afin de rendre la méthode de stéganalyse proposée de type active, nous proposons de compléter le travaille pour estimer la taille du message insérer. Cela peut être effectué par une étude de la répartition des motifs présents dans l'image.

## La stéganographie dans les billets de banque

<http://www.apprendre-en-ligne.net/crypto/stegano/10francs.html>

On trouve un exemple de stéganographie élémentaire sur tous les billets de banque suisses.

Voici l'actuel billet de dix francs:



Recto

Verso

L'image ci-contre contient le texte insérer dans le petit carré entouré d'un cercle vert



## La stéganographie dans les fichiers système

[www.noxistes.org/manipuler\\_le\\_slack\\_space\\_4.php](http://www.noxistes.org/manipuler_le_slack_space_4.php)

Le *bmap* permet de manipuler l'espace slack (espace libre) d'un seul fichier.

bmap-mode slackbytes [nom du fichier]	Permet de voir la taille du slack disponible sur un fichier.
bmap-wipslack [nom du fichier]	Permet d'effacer l'espace slack d'un fichier.
bmap-checkslack [nom du fichier]	Permet de vérifier s'il y'a du slack dans un fichier.
bmap-putslack [nom du fichier]	Permet d'écrire une chaîne de caractères dans le slack d'un fichier.
bmap-slack [nom du fichier]	Permet de récupérer ce qui contenu dans le slack d'un fichier.

### Commandes de Bmap

Le *slacker* permet de manipuler l'espace libre d'un répertoire entier, les options les plus intéressantes de *slacker* sont

Slacker- capacité [chemin complet]	Permet de calculer la taille totale disponible dans un répertoire
Slacker-fill	Permet de remplir le slack d'un répertoire avec le contenu d'un fichier
Slacker pour [chemin complet] > fichier de sortie	Permet d'écrire tout le slack d'un répertoire dans un fichier
Slack-wipe [chemin complet]	Permet de remplacer tout le slack d'un répertoire par des zéros
Slack-frob [chemin complet]	Permet de remplacer tout le slack d'un répertoire par des caractères aléatoires

### Commandes de Slacker

### *Logiciels de stéganographie 2004-2006*

- BMP Secret:** Programme pour cacher n'importe quel type de fichier dans une image BMP.  
Auteur: Parallel Worlds  
Homepage: [Http://www.pworlds.com/products/secrets.html](http://www.pworlds.com/products/secrets.html)
- Blindside:** Blindside permet de cacher un fichier ou un ensemble de fichiers dans une image BMP. Il effectue un petit changement de couleur imperceptible pour l'oeil. Une image peut contenir environ 50 KB de données.  
Auteur : John Collomosse  
Homepage: <http://www.cs.bath.ac.uk/~jpc/blindside/index.htm>
- Contraband hell:** Programme pour cacher n'importe quel type de fichier dans une image bitmap de 24 bits. Evolution de Contraband qui utilise l'algorithme IDEA pour chiffrer le fichier à dissimuler.  
Auteurs : Julius Thyssen, Hens Zimmerman  
Homepage : <http://www.jthz.com/puter>
- EmptyPic:** Cache des images GIF dans des pages web en les transformant en image unie couleur.  
Auteur : Robert Wallington  
Homepage : <http://www.crtelco.com/~robertw>
- EncryptPic:** Cache l'information dans des images bitmap de 24 bits. Utilise une protection par mot de passe et l'algorithme Cast pour chiffrer les données.  
Auteur: Frederic Collin  
Homepage: <http://www.softlookup.com/preview/dis24355.html>
- EzStego:** Chiffre et cache l'information dans des images GIF.  
Auteur : Romana Machado  
Homepage : <http://www.stego.com>

- F5:** Utilise l'algorithme stéganographique F5 pour dissimuler l'information dans des images en vraies couleurs BMP, GIF ou JPEG.  
Auteur: Andreas Westfeld  
Homepage: [http : //wwwrn.inf.tu-dresden.de/~westfeld/f5.html](http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html)
- Gifshuffle:** Programme en ligne de commandes qui cachent des messages dans des images GIF en mélangeant la palette des couleurs. Les données sont compressées et chiffrées.  
Auteur : Matthew Kwan  
Homepage: <http://www.darkside.com.au/gifshuffle/index.html>
- Hermetic Stego:** Cache l'information dans des fichiers BMP.  
Auteur: Hermetic System  
Homepage: <http://www.hermetic.ch>
- Hide In Picture:** Programme écrit en langage Euphoria. Chiffre, protège par mot de passe et cache n'importe quelle donnée dans un fichier BMP. Supporte le format GIF.  
Auteur : Davi Tassinari de Figueiredo  
Homepage: <http://www16.brinkster.com/davitf/hip/>
- Hide and Seek:** Chiffre l'information à dissimuler avec l'algorithme IDEA et la cache dans des images GIF.  
Auteur : Colin Maroney
- ImageHide:** Logiciel de stéganographie gérant plusieurs formats d'images.  
Auteur: Dancemammal  
Homepage: <Http://prem-01.portlandpremium.co.uk/p1-28/imagehide.htm>
- In The Picture:** Chiffre les données à cacher et les dissimule dans des images BMP.  
Auteur : Intar  
Homepage: <http://www.intar.com>
- Invisible Secrets:** Permet de dissimuler des données dans des fichiers JPEG, PNG, BMP, HTML et WAV. Chiffre les données avec AES-Rijdael, Blowfish, Twofish, RC4, Cast128, GOST, Diamond 2, Sapphire 2. Gestion à base de mots de passe et plein d'autres fonctionnalités de sécurité.

Auteur : NeoBytes

Homepage: <http://www.neobytesolutions.com>

**JP Hide and Seek:** Programme de stéganographie désigné pour dissimuler peu d'information, moins de 5 %, dans des images JPEG.

Auteur : Allan Latham

Homepage: <http://linux01.gwdg.de/~alatham/stego.html>

**Jsteg Shell:** Cache les données dans des images JPEG. Utilise un chiffrement RC4 sur 40 bits.

Auteur : Korejwa

**OutGuess:** Outil stéganographique pour les images JPEG qui préserve les statistiques sur les fréquences.

Auteur : Niels Provos

Homepage : <http://www.outguess.org>

**Steganografia:** Programme en Perl permettant de cacher de l'information dans des fichiers BMP sans en changer la taille.

Auteur : Cers

Homepage : <http://www.cers.tk>

**Steganography:** Chiffre et cache l'information dans des fichiers audio et image. Guillermito nous montre comment retrouver les données stéganographiées.

Auteur : Pipisoft

Homepage : <http://www.pipisoft.com>

**Steganos:** Chiffre les données avec des clés d'au moins 2048 bits et les dissimule dans des fichiers BMP, DIB, HTML, TXT, VOC et WAV.

Auteur : Centurionsoft

**StegHide:** Programme de stéganographie qui dissimule l'information dans de nombreux types de fichiers image et audio. Résistant aux attaques statistiques du premier ordre.

Auteur: Stefan Hetzl

Homepage: <http://steghide.sourceforge.net/index.php>

**StegoTif:** Cache l'information sur les bits les moins significatifs dans des images TIFF.



Auteur : Giovambattista Pulcini

Homepage : <http://www.geocities.com/SiliconValley/9210>

**S-Tools:**

Cache l'information dans des fichiers image ou audio ou même sur le disque dur. Utilise les algorithmes IDEA et DESS pour chiffrer les données à dissimuler. Implémente un générateur de pseudo aléa pour choisir les bits supports. C'est un des outils les plus complet.

Auteur : Andy Brown

- [Avc 01] I. Avcibas, N. Memon, and B. Sankur, **Steganalysis Using Image Quality Metrics**. In Security and Watermarking of Multimedia Contents, SPIE. San Jose, CA, 2001.
- [Avc 02] I. Avcibas, N. Memon, and B. Sankur, **Image Steganalysis With Binary Similarity Measures**. In IEEE International Conference on Image Processing, Rochester, New York, 2002.
- [Bar 06] J. Barbier, E. Filiol, and K. Mayoura, **New Features for Specific JPEG Steganalysis**. In Proc. 3rd International Conference on Computer, Information, and Systems Science, and Engineering, CISE 2006.
- [Bar 07] Johann Barbier, **Analyse de Canaux de Communication dans un Contexte non Coopératif**, Thèse pour obtenir le grade de docteur, ESAT - Laboratoire de Virologie et Cryptologie, B.P. 18, 35 998 Rennes Cedex, 2007.
- [Ben 04] Mohamed Bentoumi, **Outils Pour La Détection et La Classification**, Thèse pour obtenir le grade de docteur, Université Henri Poincaré-Nancy1, 2004.
- [Bi 96] D. Bi, J.P. Asselin de Beauville, M.Mraghni, **Spatial Gray Levels Distribution Based Unsupervised Texture Segmentation**. In Proceedings of 3rd International Conference of Signal Processing (ICSP96), Pékin (Chine), 1996
- [Car 03] Y. Caron, N. Vincent, P. Makris : **Mesure de la Qualité de la Compression par l'Utilisation de la Loi de Zipf** Publication de l'équipe RFAI, Compression et Représentation de Signaux Audiovisuels – CORESA'03 , Lyon (France), pp.239-242, 2003.
- [Clu 02] Jean-Pierre CLUTIER, **Cryptographie et certification**, Travail de diplôme, Conservatoire National Des Arts et Métiers (CNAM) ,2002.
- [Coh 97] A. Cohen, R.N. Mantegna,, S. Havlin, **Numerical Analysis of Word Frequencies in Artificial and Natural Language Texts**, *Fractals*, Vol. 5 No.1, pp. 95-104,1997.
- [Den 05] Jenny Dentand, **Stéganographie**, travail de diplôme, Haute Ecole de Gestion de Genève (HEG-GE), 2005.
- [Dum 03] S.Dumitrescu, X.Wu, and Z.Wang, **Detection of LSB Steganography via Sample Pair Analysis**. In IEEE transactions on Signal Processing, pp. 1995-2007, 2003
- [Far 02] H. Farid and S. Lyu, **Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines**. In Pre-proceedings 5th Information Hiding Workshop, Noordwijkerhout, Netherlands, 2002.
- [Fou 87] Jacqueline Fourastie Jean-François Laslier, **Probabilité et Statistique**, 3<sup>ème</sup> édition, ISBN 2-04-016938-5,1987.
-

- [Fri 02a] J. Fridrich, M. Goljan, and D. Hoge, **Steganalysis of JPEG Images: Breaking the F5 Tlgorithm**. In Proceedings, Information Hiding, 5th International Workshop, IH 2002, pp, 310-323. Noordwijkerhout, The Netherlands, 2002.
- [Fri 02b] Jessica Fridrich, Miroslav Goljan, and Dorin Hoge, **Attacking the Outguess**. In ACM Workshop on Multimedia and Security 2002, Juan-les-Pins, France, 2002.
- [Fri 01] J. Fridrich, M. Goljan, and R. Dui, **Reliable Detection of LSB Steganography in Color and Grayscale Images**. In ACM Workshop on Multimedia and Security, pp.27–30, 2001.
- [Fri 04] J. Fridrich, **Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes**. in Information Hiding, 6th International Workshop, IH 2004. Toronto, Canada: Springer, pp.76-81, ISBN: 3-540-24207-4,2004.
- [Gal 04] Fabian Galand, **Stéganaographie**, Traité de Sécurité des systèmes d'information, Techniques de l'Ingénieur, Ch H 5870, 2004.
- [Gol 03] M. Goljan J. Fridrich and D. Hoge, **New Methodology for Breaking Steganographic Techniques for Jpegs**. In EI SPIE Santa Clara, CA, 2003.
- [Gui 98] Jean-Paul GUILLOIS, **Compression de Données, Compression des Images**, Techniques de l'Ingénieur. Traité d'électronique, E5340, 1998.
- [Hen 98] Maitre Henri, **Image Watermarking: Why is Watermarking a Hard Problem?** Proc. Korea-France Workshop on Multimedia, 1998.
- [Kob 08] Nouha Kobsi, F.H.Merouani, **Proposition d'un Multi-Classifieur pour une Stéganalyse d'Image**, Infodays'08-Chlef, Algérie,15-16,2008.
- [Lai 09a] L. Laimeche & H.F.Merouani, **Steganalysis of LSB Steganography Using Ziph's Law**, International Conference on Software, Knowledge, Information Management and Applications, SKIMA'09-Fes, Morocco, October 21-23, 2009.
- [Lai 09b] L. Laimeche & H.F.Merouani, **Les propriétés de loi de Zipf pour l'analyse stéganographique des images**, Journée de l'étudiant à ESI (Ecole nationale Supérieure d'Informatique), JEESI'09- Oued Smar ,Alger, 19 Mai 2009.
- [Lai 09c] L. Laimeche & H.F.Merouani, **Proposition d'une Stéganalyse basée loi de Zipf**, Journées Gestion Electronique de Documents & Réseaux de Recherche en Sciences et Technologies de l'Information, GED'09-Annaba, Algérie, 20 et 21 Mai 2009.

- [Lai 09d] L. Laimeche & H.F.Merouani, **Detection Hidden Messages Using Zipf's Law**, 5ème Symposium International, Image Multimédias Applications Graphiques et Environnements, IMAGE'09-Biskra, Algérie, Novembre 3-5, 2009.
- [Lat 99] A.Latham, **Steganography: Jphide and Jpseek**, <http://linux01.gwdg.de/alatham/stego.html>, 1999.
- [Lec 03] Pierre Le Chapelain, **Analyse Stéganographique d'Images Numériques, Comparaison de Différentes Méthodes**, Rapport de stage, Laboratoire des Images et des Signaux, 961, rue de la Houille Blanche, 2003.
- [Lu 05] Chun-Shien Lu, **Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property**, Published in the United States of America by Idea Group Publishing. 701 E. Chocolate Avenue, Suite 200, ISBN 1-59140-192-5, 2005
- [Lo 08] Cheikh LO, **La Stéganographie Appliquée aux Textes & Images**, Travail de diplôme, IUP de Rouen- Institut Universitaire Professionnalisé, 2008.
- [Lyu 04] S. Lyu H. and Farid, **Steganalysis Using Color Wavelet Statistics and One Class Support Vector Machines**. in Proc. SPIE, Security and Watermarking of Multimedia Contents VI, San Jose, CA, USA, 2004.
- [Lyu 06] S. Lyu and H. Farid, **Steganalysis Using Higher-Order Image Statistics**. IEEE Transactions on Information Forensics and Security, vol. 1, 2006
- [Mak 99] Pascal Makris, Jean Pierre Bonnefoy et Nicole Vincent, **Etude Statistique des Motifs Présents dans une Image**, Dix-septième colloque GRETSI, Vannes, 1999.
- [Mem01] N. Memon and R. Chandramouli, **Analysis of LSB Based Image Steganography techniques**. In Proceedings of the International Conference on Image Processing, Thessaloniki Greece, 2001.
- [Nik 08] Ljupce Nikolov, **Stéganographie Détection de messages cachés**, Haute Ecole Spécialisé de Suisse occidentale (HES.SO), 2008.
- [Pet 02] Petitcolas, FAP, **MP3stego**, <http://www.cl.cam.ac.uk/fapp2/steganography/mp3stego>, 2002.
- [Pro 01] Niels Provos, **Defending Against Statistical Steganalysis**. In 10th USENIX Security Symposium, pp 323–336, 2001.
- [Pro 03] Niels Provos et Peter Honeyman, **Hide and Seek: An Introduction to Steganography**, IEEE Computer Society, 2003.
- [Ray 02] Frédéric Raynal, **Etude d'Outils pour la Dissimulation d'Information**, Thèse de doctorat, Université paris XI, 2002.
-

- [Rey 03] Christian Rey, **Tatouage d'Image: Gain en Robustesse et Intégrité des Images**, Thèse pour obtenir le grade de docteur, Université d'Avignon, 2003.
- [Rou 04] Benoit Roue, Patrick Bas, Jean-Marc Chassery, **Improving LSB Steganalysis Using Marginal and Joint Probabilistic Distributions**. MM&Sec : 75-80, 2004.
- [Sil 01] Joshua Silman, **Steganography and Steganalysis: An Overview**, gsec 1.2 f, 2001.
- [Vin 00] N.Vincent, P.Makris, J.Brodier. **Compressed Image Quality and Zipf's Law**, Proceedings of International Conference on Signal Processing (ICSP – IFIC-IAPRWCC2000), pp. 1077-1084, Pékin (Chine), 2000.
- [Wes 99] Andreas Westfeld and Andreas Pfitzmann, **Attacks on Steganographic Systems**. In 3<sup>rd</sup> Info. Hiding Workshop, Dresden, Germany, September 28-October 1, LNCS vol. 1768, Springer Verlag, New York, 1999.
- [Wes 01] Andreas Westfeld, **F5-A Steganographic Algorithm**, Information Hiding: 289-302, 2001.
- [Yeu 97] Minerva M. Yeung & Fred Mintzer, **An Invisible Watermarking Technique for Image Verification**. IEEE Int. Conf. on Image Processing, pp. 680-683, 1997.
- [Zip 49] G.K. Zipf, **Human Behavior and the Principle of Least Effort**. Addison-Wesley, New York, 1949

- [Web 1.1] <http://bcs.fltr.ucl.ac.be/FE/08/stegano.htm>
- [Web 1.2] [www.noxistes.org/manipuler\\_le\\_slack\\_space\\_4.php](http://www.noxistes.org/manipuler_le_slack_space_4.php)
- [Web 1.3] [fr.wikipedia.org/wiki/Cryptographie](http://fr.wikipedia.org/wiki/Cryptographie)
- [Web 1.4] [www.zdnet.fr/.../invisible-secrets-11006550s.htm](http://www.zdnet.fr/.../invisible-secrets-11006550s.htm)
- [Web 2.1] [www.europaschool.net/static/formation/.../images\\_bitmap.pdf](http://www.europaschool.net/static/formation/.../images_bitmap.pdf)
- [Web 2.2] <http://www.commentcamarche.net/contents/video/cmy-cmj-cmyk-cmjn.php3>
- [Web 2.3] [www.clubic.com/telecharger-fiche10489-hex-editor.html](http://www.clubic.com/telecharger-fiche10489-hex-editor.html)
- [Web 2.4] [www.turrier.fr/tutoriels/form\\_02/form\\_02.html](http://www.turrier.fr/tutoriels/form_02/form_02.html)
- [Web 2.5] [www.computing.surrey.ac.uk/teaching/2006.../jsteg-h.pdf](http://www.computing.surrey.ac.uk/teaching/2006.../jsteg-h.pdf)
- [Web 4.1] [alexsoft.chez-alice.fr/logiciels/cil.htm](http://alexsoft.chez-alice.fr/logiciels/cil.htm).
- [Web 4.2] [www.zdnet.fr/.../invisible-secrets-11006550s.htm](http://www.zdnet.fr/.../invisible-secrets-11006550s.htm)
- [Web 4.3] [www.infos-du-net.com/.../Cameleon,0301-6364.html](http://www.infos-du-net.com/.../Cameleon,0301-6364.html).
- [Web 4.4] [www.scanwith.com/.../JPHS\\_for\\_Windows.htm](http://www.scanwith.com/.../JPHS_for_Windows.htm)
- [Web 4.5] [Fr.wikipedia.org/.../Analyse\\_discriminante\\_lineaire](http://fr.wikipedia.org/.../Analyse_discriminante_lineaire)
- [Web 4.6] [hal.archives-ouvertes.fr/docs/00/38/66/52/PDF/p92.pdf](http://hal.archives-ouvertes.fr/docs/00/38/66/52/PDF/p92.pdf)
- [Web 4.7] <http://philip.greenspun.com/>.