MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
وزارة التعليـــم العـالـي والبـحـث العلمـي

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITÉ BADJI MOKHTAR-ANNABA

جامعـة باجي مختار– عنابـــــــــة

FACULTE DES SCIENCES DE L'INGENIORAT
DEPARTEMENT D'INFORMATIQUE

كليـة علـــــــوم الهندســـــــة
قسـم الإعـــــــلام الآلـــــــي

Année : 2016/2017

THESE

Présentée en vue de l'obtention du diplôme de
Doctorat en Informatique

La Sélection des Caractéristiques Parallèle pour la Stéganalyse

Filière : **Informatique**

Spécialité : **Traitement d'image et vision artificielle**

Par
BOUGUERNE IMEN

Devant le jury :

Merouani Hayet Farida	Pr à l'Université Badji Mokhtar-Annaba	(Président)
Tlili Guiassa Yamina	Pr à l'Université Badji Mokhtar-Annaba	(Directeur)
Azizi Nabiha	Dr à l'université Badji Mokhtar-Annaba	(Examinateur)
Kimour M.Taher	Pr à l'Université Badji Mokhtar-Annaba	(Examinateur)
Redjimi Mohamed	Pr à l'université de skikda	(Examinateur)

ملخص

مبدأ اكتشاف المعلومات المخبئة (ستيغانااليز) هو لتصنيف وثيقة مشبوهة ما إذا كانت أصلية أو تحتوي على معلومات دخيلة. تقترح هذه الأطروحة طريقة جديدة عالمي ستيغانااليز لصور جيباك(ج ب ا ك) و هذا استنادا على اساس معالجة الخصائص بطريقة تحويل الهجينة جيب التمام المنفصلة و تحويل كونتغلات ، حيث ان الكشف يعتبر عموما كمشكلة تصنيف، ونحن نستعمل اختيار الخصائص الموجهة بطريقة انقاص مجالات الثقة. طريقة اختيار الخصائص تمكننا أيضا من تفسير الصور المختارة ،بحيث ان الهدف يكون بفهم أداء الرئيسي للخوارزميات إخفاء المعلومات، كذالك نستخدم الجبر الخطي لنواة مجموعة من الاشعة الحوامل(س ف س) لتخفيض كلفة الحوسبة منخفضة التكلفة في التصنيف. من ناحية أخرى، يمكن إبراز الخصائص المحددة (عادة 10 إلى 13 مرات أقل مما كانت عليه في المجموعة الكاملة) تتيح لنا تسليط الضوء على نقاط الضعف والقوة في الخوارزميات المستعملة.

المفاتيح : ستيغانااليز , جيب التمام المنفصلة , اختيار الميزات , ناقلات دعم, إخفاء المعلومات.

ABSTRACT

The principle of steganalysis is to classify an offending document as original or as stego. This thesis proposes a new method for universal steganalysis for JPEG images based on the characteristics of hybrid processing (discrete cosine transform and contourlet) then the detection is generally considered as a classification problem, we use feature selection, oriented towards declining confidence intervals usually given results. The selection of characteristics also possible to envisage an interpretation of images of selected characteristics, in order to understand the inner workings of algorithms for steganography and using linear algebra to a core of support vectors (SVS) and low-cost computing to reduce the classification calculation cost. Furthermore, the features selected (usually 10 to 13 times less than in the complete set) actually serve to highlight weaknesses and the benefits of the algorithms used.

Keywords: steganalysis, discrete cosine transform, features selected, support vectors, steganography.

RESUME

Le principe de la stéganalyse est de classer un document incriminé comme original ou comme stéganographié. Cette thèse propose une nouvelle méthode de stéganalyse universelle pour les images JPEG basées sur les caractéristiques de transformation hybride (Transformer en cosinus discrète et transformer en contourlet), Puis la détection est généralement considérée comme un problème de classification. Nous utilisons la sélection de caractéristiques, orientée vers une diminution des intervalles de confiance. La sélection des caractéristiques permet également d'envisager une interprétation d'images sélectionnées, dans le but de comprendre le fonctionnement essentiel des algorithmes de stéganographie, ainsi en utilisant l'algèbre linéaire d'un noyau des vecteurs de support (SVS) et à faible coût de calcul. D'autre part, les caractéristiques sélectionnées (généralement 10 à 13 fois moins nombreuses que dans l'ensemble complet) permettent effectivement de faire ressortir les faiblesses ainsi que les avantages des algorithmes utilisés.

Mots clés : stéganalyse, cosinus discrète, La sélection des caractéristiques, vecteurs de support, stéganographie.

DEDICACE

أيا لغة الذات .. فبعنفوان أبجديتك العصماء تنساب مفرداتي ...

إلى الذي سقاني إكسير النبل و الإباء.. إلى الحب المهاجر من دنيا البشر إلى الشرايين ..

إلى إبنتي جمانة و زوجي

إلى الذي ينسج الأمان على درب خطاي و ينثر الأقمار في عتمة أزمنتي إلى إخوتي و أخواتي...

إلى مرفإ حناني و حبي الأبدي أمي و أبي

إليكما أجتهد.. لأعزف بلغة البحث, قصيدة حلم كان ينام خلف أعمدة الزمن.

REMERCIEMENTS

Toutes les louanges sont dues à ALLAH, Dieu le tout-puissant, le Noble Généreux et le Majestueux, pour m'avoir accordée la bénédiction , le courage et la patience pour accomplir ce travail.

Qu'il me soit permis de témoigner ma profonde et sincère reconnaissance envers mon directeur de mémoire, la professeur Tlili Yamina, pour avoir accepté de diriger mon travail de recherche. Je le remercie pour la confiance qu'elle m'a témoignée, les encouragements qu'elle m'a donnés et les remarques apportées à ce mémoire.

Je tiens à exprimer ma profonde gratitude envers Mm. Merouani Hayet, *Pr à l'université Badji Mokhtar- Annaba. Je voudrais la remerciée pour l'honneur qu'elle me fait en tant que président du jury.*

J'exprime également ma reconnaissance envers Mm. Azizi Nabiha, M. *Kimour M.Taher et M. Redjimi Mohamed, d'avoir accepté de rapporter ce manuscrit. Leur rapports témoignent de l'effort et du temps qu'ils y ont consacré, je suis très reconnaissante.*

Mes remerciements vont également à tous les membres du département informatique de l'université d'Annaba pour l'accueil, l'estime, l'encouragement et l'aide que j'ai reçus durant mon séjour au département. Je remercie mes enseignants et les membres du laboratoire LRI.

Je tiens à adresser un remerciement particulier à SIEF la personne, qui était la plus proche à mon cœur ces années, et qui m'a vraiment soutenu pour achever ce travail.

Je remercie ma famille pour son soutien, ses encouragements et d'avoir toujours été là pour moi, dans les bons comme dans les mauvais moments. Je ne remercierais jamais assez mes parents; si j'en suis là aujourd'hui c'est grâce à eux.

LISTE DES TABLEAUX

LISTE DES FIGURES

Liste des symboles

BMP : BitMaP format.

bpp : bit per pixel ou bit par pixel.

DCT : Discret Cosinus Transform.

DFC : Domaine Fréquentiel Compressé.

CT : Transformation en Contourlet.

D_H: La distance de Hamming $d_H(x, y)$

FFT : Fast Fourier Transform.

GIF : Graphics Interchange Format

GPA : Générateur Pseudo-Aléatoire.

IDCT : Inverse Discret Cosinus Transform.

IV : Init Vector.

JPEG : Joint Picture Experts Group.

LDPC : Low Density Parity Check.

LSB : Least Significant Bit.

CGC : Codage de Gray Canonique.

MDS : Maximum Separable Distance.

MBPIS : Multi Bit Plane Image Steganography

MEM: Multiple Embedding Method.

MGP : Matrice Génératrice Polynomiale.

MSM : Méthode de Stéganographie Multiple.

PA : Passive Attack.

bpp : bit per pixel ou bit par pixel.

ROC : Receiver Operating Characteristic.

RLE : Run Length Encoding.

RVB : Rouge, Vert, Bleu.

SSA : Specific Steganalysis Attack.

TRANSEC : TRANsmission SECurity.

SVM : Support Vector Machines.

USA : Universal Steganalysis Attack.

VLC : Variable Length Coding.

YCbCr : Luminance, Chrominance bleue, Chrominance rouge.

Table de matière

Chapitre 3 : Le modèle d'attaquant pour la stéganalyse

INTRODUCTION GENERALE

INTRODUCTION GENERALE

L'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

Le développement des supports numériques et des réseaux de communication a facilité le partage et le transfert des données numériques, introduisant ainsi de nouvelles formes de piratage de documents et de nouveaux défits de sécurité à relever. 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, et assurer la confidentialité d'une transmission des nouvelles méthodes ont été développé. Il s'agit des méthodes de dissimulation d'information.

La dissimulation d'information cherche à cacher une information de n'importe qu'elle 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.

L'utilisation de la stéganographie paraît bien adaptée au vol d'informations confidentielles, car les messages cachés sont difficilement détectables, il est nécessaire de prendre en mesures de sécurité liées à la mauvaise utilisation de la stéganographie. Il existe des techniques permettant de découvrir les média stéganographie : 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 passive, et la stéganalyse active.

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 JPEQ. Ces caractéristiques statistiques sont basées sur l'utilisation de deux domaines de transformation (DCT , CONTOURLET) . Ensuite, la détection est souvent

présentée comme un problème de classification, nous avons utilisé une approche pour réduire la complexité du classifieur et réduire la dimensionnalité du problème par la sélection de variables.

La méthode de stéganalyse proposée se fait en deux étapes principales: d'une part l'extraction des caractéristiques pour former l'espace de caractéristiques, ou l'espace de représentation, et d'autre part l'utilisation du classifieur SVR permettant de classer une nouvelle image sur l'une des deux classes d'images (propres, stéganographiées), et tente à réduire le nombre de caractéristiques par la sélection, qui rend possible l'analyse de cette dernières , et mettent en évidence des possibles faiblesses de l'algorithme de stéganographie.

Nous décrivons au premier chapitre la terminologie et les notions de base de la stéganographie, Nous allons d'abord expliquer la dissimulation d'information, puis distinguer brièvement la stéganographie de la cryptographie. Il est intéressant de pouvoir déterminer quelles sont les méthodes de dissimulations les plus largement utilisées par ces produits. Nous nous sommes attachés principalement à étudier les techniques de stéganographie dédiées aux images fixes. Le chapitre 2 présente les grandes lignes des étapes qui composent la compression JPEG, et nous détaillons précisément les algorithmes de stéganographie pour lesquels nous avons spécifié des distingueurs. Nous illustrons cette technique par la description de distingueurs stéganographiques universels, d'une part, et spécifiques, d'autre part, pour les algorithmes Outguess , F5 et JPHide and JPSeek . En outre finement le format de compression JPEG afin de comprendre les mécanismes sous-jacents et les contraintes imposées pour utiliser les fichiers JPEG comme média de couverture. Au chapitre 3, nous nous sommes inspirés des définitions et des modèles de sécurité classiques afin de définir formellement la notion de stéganalyse. Au chapitre 4 nous détaillons les différentes méthodes d'apprentissage automatique et nous allons donner quelques résultats de comparaison appliquée à la stéganalyse. Chapitre 5 traites les méthodes d'extraction de caractéristiques, et son analyse de la performance, la description du classifieur est détaillée dans la deuxième partie. Le Chapitre 6 il s'agit des résultats et les analyses expérimentales, les résultats expérimentaux montrent que notre méthode est en mesure d'améliorer les résultats de la classification dans un temps efficace.

Chapitre 1

Stéganographie et image

1.1 Introduction

La stéganographie est une science et un art utilisé depuis des siècles pour faire passer inaperçu un message secret dans un fichier anodin. Ce mot vient du grec " Stéganô ", qui signifie couvrir et " Graphô " qui veut dire écriture. Ainsi, on dissimule les informations que l'on souhaite transmettre confidentiellement dans un ensemble de données d'apparence anodine afin que leur présence reste imperceptible. Contrairement à la cryptographie, les informations sont cachées mais pas nécessairement chiffrées. Bien que considérées comme deux disciplines différentes, il est possible d'intégrer la cryptographie dans un message ; ainsi, la communication n'est pas seulement dissimulée mais également chiffré.

Il est une discipline connue émulant l'imagination et la curiosité, c'est bien la cryptographie. Du code de César au « Da Vinci code », la cryptographie fascine ; tantôt elle est l'apanage des militaires et espions au secours de l'histoire ou d'amours impossibles, tantôt elle préoccupe les mathématiciens par les énigmes qu'elle offre [37, 119, 130]. Peut- être plus ancienne et souvent amalgamée à la cryptographie, la stéganographie vit dans l'ombre des « codes secrets », dissimulée derrière un objectif et un formalisme à la fois proches et différents de ceux de la cryptographie. Son étymologie grecque « stego », le secret et « graphia », l'écriture, l'enracine dans l'antiquité. La stéganographie est donc l'art de l'écriture des donnés secrète.

1.2. Historique de la stéganographie

Dans les siècles passés, les chercheurs se sont penché à connaitre l'origine de la stéganographie . En Chine, la coutume veut que le signal de la révolte des chinois contre la dynastie mongole lors de la fête de la lune, a été donné par des messages cachés dans des gâteaux de lune. L'invention de l'encre sympathique est attribuée au naturaliste Pline l'Ancien, il est toujours utilisé par des organisations mondiales. Les techniques de stéganographie devenant de plus en plus difficile à découvrir, les premiers ouvrages traitant du sujet voient le jour à partir du XVIe siècle. En 1499, l'abbé Jean Trithème (1462-1516) publie le premier traité de stéganographie, intitulé Steganographia et composé de trois livres. Le troisième livre n'a été finalement « décodé » par Thomas Ernst qu'en 1996 et indépendamment par Jim Reeds [80] en 1998.

Gaspart Schott entre 1608-1666 explique dans son livre Schola Steganographica comment dissimuler des messages en utilisant des notes de musique. Certains de ces ouvrages, à l'exemple de Stéganographia ont été interdits en leur temps, néanmoins, l'intérêt croissant du public pour les sciences du secret a rendu possible la diffusion de ces livres. La littérature en tant que vecteur de diffusion d'information est elle-même considéré par la majorité des chercheurs comme support servant à dissimuler des messages.

Médiévale ou moderne, les plus célèbres d'entre eux sont notamment le poème de Boccacio (1313-1375), Amorosa visione, long d'environ 1500 vers et la correspondance privée entre George Sand et Alfred de Musset en 1883.

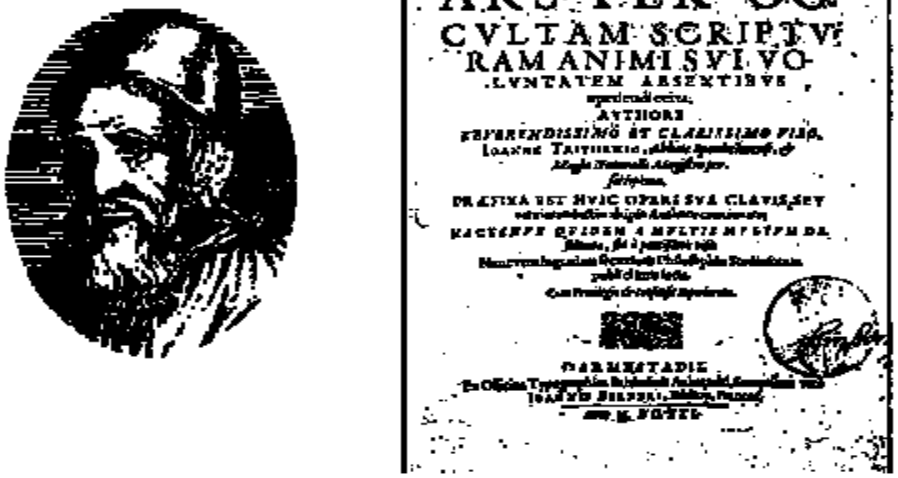


Figure1.1 : Jean Trithème et Stéganographia [99].

Les méthodes deviennent de plus en plus complexes avec le temps et marqué aussi par l'invention du micro-film en 1857 par Sir Brewster, puis du micro-point, a donné un nouveau souffle à la stéganographie.

Elles permettent ainsi de réduire des photos ou des images de taille déférente dans la taille d'un point sur un i et de les cachées dans un texte. Pendant les différentes guerres franco-allemandes, les militaires et les services de renseignement ont utilisés ces techniques. L'histoire contemporaine, notamment celle de la France a été pointé par l'emploie de la stéganographie. Le message de Verlaine, propagé en deux parties sur les ondes de la BBC le 5

juin 1944 à 21h15, « Les sanglots longs des violons de l'automne » et « Blessent mon cœur d'une langueur monotone », annonce le débarquement imminent des alliés. Margaret Tatcher réussit à identifier la source de nombreuses fuites de documents en Grande Bretagne en traçant ceux-ci à l'aide de techniques de dissimulation d'information. Enfin, plus récemment, de nombreux média [50, 180, 49, 40] avancent l'hypothèse de Bin Laden « les attentats du 11 septembre 2001 » qu'utilise des messages cachés dans des images de sites à caractère pornographique. Le lecteur féru d'épistémologie trouvera son bonheur dans [45, 81, 39, 63, 53].

La stéganographie dite moderne, lorsque il est adaptée aux données numériques (images, son,..). Elle suit depuis le milieu des années 90 un démarrage corrélé à celui d'Internet ; plusieurs conférences scientifiques proposant des sessions dédiées à la dissimulation d'information augmentant chaque année. En 1996, c'est la naissance de la communauté des stéganographes à partir de de la première édition d'Information Hiding et l'adoption d'un corpus relatif à la dissimulation d'information [56], et en 1997 qu'est soutenue l'une des premières thèses [120] dans le domaine [167].

Figure 1. 2 : Correspondance entre George Sand et Alfred de Musset [129].

1.3. La dissimulation d'information

Il existe plusieurs ouvrages excellents qui se référer à [56, 159,146] traitants de dissimulation d'information. G.J. Simmons pose en 1983 la base de la stéganographie moderne en définissant la notion de canal subliminal[1].

1 : la communication entre les deux prisonniers se faisant de manière ouverte, mais indétectable, il considère que « subliminal » est plus adapté. Deux méthodes secondaires permettent aussi la dissimulation d'informations : les canaux cachés et l'anonymat.

Dans cette partie, en premier nous allons expliquer ce qu'est la dissimulation d'information, et ensuite nous distinguons brièvement la stéganographie de la cryptographie. D'après [100] nous constatons que la stéganographie fait partie du domaine de la dissimulation d'information comme l'illustre la figure 1.3, il existe trois méthodes principales assurent la dissimulation d'informations: la stéganographie, le tatouage et le fingerprinting[2].

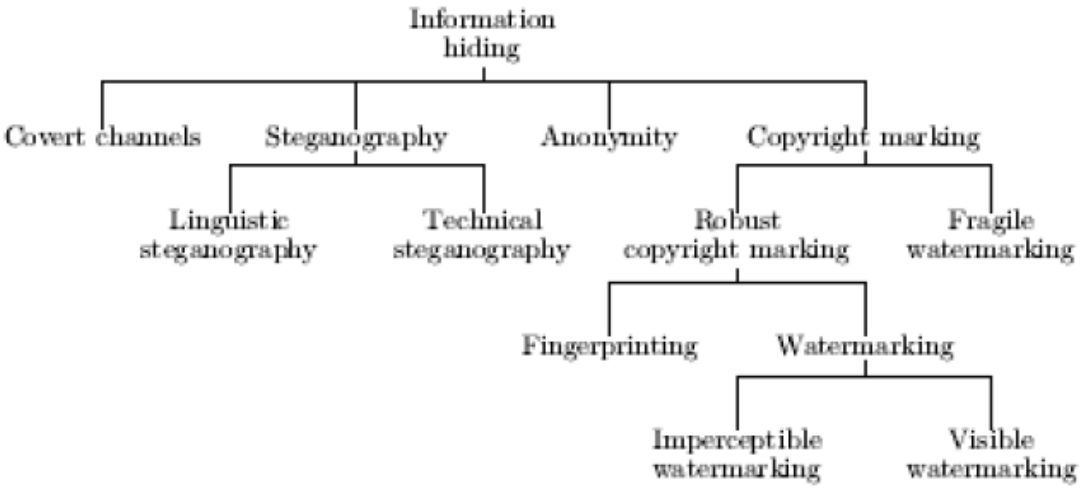


Figure 1.3 : La dissimulation d'information et ses sous ensemble [100].

1. *la stéganographie* : a pour objectif de cacher l'existence du message donc personne ne peut voir qu'il y a un message. La stéganographie est dite *passive [32]* lorsqu'une personne anodin tente uniquement de détecter la présence d'un message dans un cover-medium transmit par un canal de communication. Cependant, il peut aussi être *actif* : si une personne sait que le stégo-medium contient de l'information. Il tente de la modifier ou de l'extraire. Pour plus de détaille voir section 1.4

2 : Nous avons choisi de traduire ainsi respectivement watermarking et fingerprinting. Signalons que le terme "filigrane" est parfois employé pour watermarking.

2. *Le tatouage* permet l'identification de l'entité du copyright. Donc il tente de protégé les droits d'auteurs.

Des données sont insérées dans les documents, de manière plus ou moins discrète, l'essentiel étant de ne pas nuire à l'usage du document. Ces données doivent être difficiles à retirer. Plus précisément, réussir à les enlever doit aboutir à un document très dégradé. Toutes les copies d'un même document d'origine sont rigoureusement identiques. L'insertion du tatouage doit limiter les modifications subies par le médium. Par suite, chaque copie du stégo medium contient une marque identique : celle du propriétaire légal. Il ne s'agit pas de dissimulation à proprement parler. La présence du tatouage dans le stégo medium est connue. Cependant cette connaissance est insuffisante. L'effet à obtenir est préventif. Les modifications apportées au stégo-medium tatoué attestent d'une contrefaçon.

3. *Le fingerprinting* assure la détection des copies illégales d'un stégo-objet. Chaque utilisateur authentifié reçoit sa propre copie du document qui contient une empreinte : l'identifiant. Ainsi, lorsqu'une copie illégale est découverte, la lecture de l'empreinte indique la source de la fuite. A la différence du tatouage où l'origine du médium est déterminante, le fingerprinting se préoccupe du destinataire. Chaque copie du médium contient une information différente, relative à son utilisateur, rendant alors chaque stégo-objet unique [123].

Il existe une relation entre la stéganographie, le tatouage et le fingerprinting : on dispose d'un document et on souhaite y insérer une information additionnelle sans saboter de manière importante le document d'origine. Cependant, le cahier des charges de la stéganographie diffère légèrement de celui du tatouage ou du filigrane : la dissimulation tient une place plus centrale tandis que d'autres propriétés ne sont pas requises [85,116] .

Contrairement aux chiffrements, qui s'appliquent sans réserve à tout type de données numériques, les algorithmes stéganographiques sont tributaires du format des documents numériques dans lesquels doit avoir lieu l'insertion. Le document d'origine doit être modifié de manière indétectable, ce qui implique de se restreindre à des zones particulières, qui dépendent naturellement du type de document [121 , 122].

(le tatouage) a pour objet de permettre l'identification de l'entité à l'origine du document, cela correspond au copyright. Des données sont insérées dans les documents, de manière plus ou moins discrète, l'essentiel étant de ne pas nuire à l'usage du document. Ces données

doivent être difficiles à retirer. Plus précisément, réussir à les enlever doit aboutir à un document très dégradé. Toutes les copies d'un même document d'origine sont rigoureusement identiques. Tout comme le tatouage, Le fingerprinting a également un but d'identification, mais il ne s'agit plus d'identifier l'émetteur : on souhaite marquer chaque copie distribuée de manière unique. Le filigrane joue donc le rôle de numéro de série. La principale contrainte reposant sur le filigrane est la résistance à la contrefaçon. L'accès à plusieurs copies, comportant chacune une marque différente, ne doit pas permettre de fabriquer une nouvelle copie avec une marque valide. Une copie ainsi formée doit permettre d'identifier au moins l'une des copies ayant servi à sa construction. Cela impose, comme pour le tatouage, une certaine robustesse de l'insertion des filigranes ainsi qu'une importante furtivité.

1.4 La stéganographie

La stéganographie arrive comme un moyen de pouvoir avoir contrôle de ce que nous laissons transparaître vers l'extérieur. A une époque où l'on arrive plus à estimer la puissance de certaine agence de renseignement, ni à connaître leur véritable capacité, il semble légitime de s'assurer que notre sphère privée soit respectée.

Pour illustrer le domaine d'application de la stéganographie, cette dernière est souvent introduite par le problème des prisonniers [64].

Deux prisonniers souhaitent établir un plan d'évasion. Pour ce faire, ils ont la possibilité de se transmettre des messages. Cependant, ces messages transitent à travers le gardien, qui a donc accès au contenu. Tout contenu jugé inapproprié sera détruit, et pourra engendrer une lourde sanction. Dans cette optique, le contenu doit donc paraître anodin aux yeux du gardien. Cette situation rend inutilisable la cryptographie, car un contenu indéchiffrable attirera l'attention du geôlier [33].

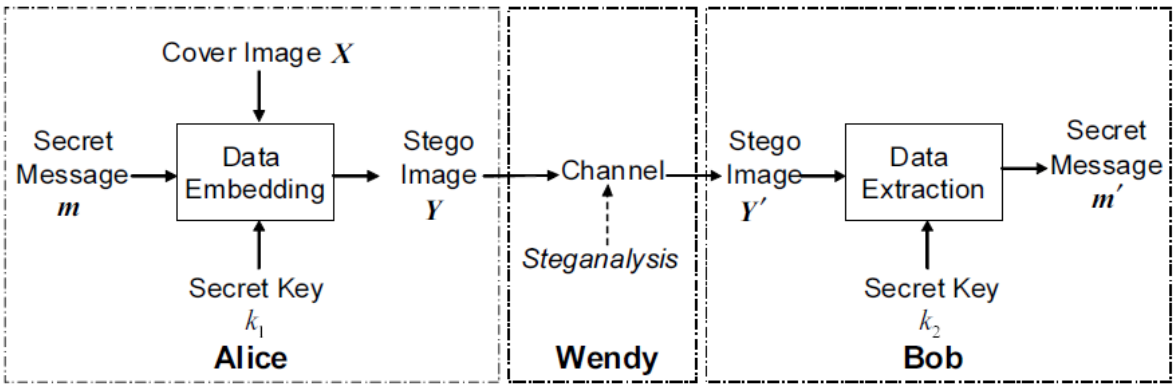


Figure 1.4 : Problème des prisonniers [7].

L'objectif de la stéganographie est de pouvoir entretenir des communications sécurisées, sans attirer l'attention d'autrui. La stéganographie connaît ses premières prémices à l'antiquité. A cette époque, les messages étaient cachés sur le crâne rasé d'un esclave. Le cheveu pouvait transmettre le message sans se faire inquiéter [110].

D'autres méthodes sont apparues par la suite. Et l'utilisation de l'encre invisible eu grand succès il y a quelques décennies. Le principe était d'écrire le message devant rester secret avec une encre invisible (jus de citron, urine, etc...), puis d'écrire cette fois-ci avec une encre bien visible un message au contenu anodin. Une fois arrivé à destination, un traitement particulier permettait de recouvrir le message. De nos jours, la stéganographie a pris de l'ampleur dans des supports bien particuliers, les supports numériques. Que ce soit des fichiers audio, vidéo ou image, il représente des supports privilégié pour la transmission d'information. L'Apparition d'Internet permet que la taille des données soit énorme. Cela représente autant de support possible à la stéganographie [63].

1.4.1. Le problème du prisonnier et la stéganographie

Le contexte général du problème est le suivant. Soient Alice et Bob, deux personnages partageant un secret commun et désirant communiquer ensemble de façon « sécurisée » ;
Wendy une amie indiscrète qui voudrait bien avoir accès au contenu de leur correspondance.

Un premier moyen pour Alice et Bob de protéger leurs communications est d'utiliser la cryptographie afin d'assurer notamment la confidentialité, l'intégrité, l'authenticité des messages qu'ils s'échangent. En employant la cryptographie, ils mettent ainsi en œuvre la sécurité de communication. Dans de nombreux cas de figure, cette seule protection est suffisante. Prenons maintenant l'exemple d'un agent infiltré dans une organisation mafieuse qui doit rester en contact avec un agent de liaison de la police. Dans ce cas très précis, les deux agents doivent évidemment protéger leurs communications afin qu'un tiers interceptant le message ne puisse apprendre aucune information. De plus, l'existence même de leurs communications, indépendamment de leur contenu, peut compromettre la couverture de l'agent infiltré. En effet, la présence du numéro de la police sur le portable d'un membre de la pègre le désignerait rapidement comme suspect. Ils doivent alors rendre furtif leur canal de transmission, en mettant en œuvre de la sécurité de transmission .

Dans le contexte du problème du prisonnier, [2,41] Alice et Bob sont deux détenus qui communiquent par l'intermédiaire de Wendy, le gardien. Si Wendy soupçonne qu'ils élaborent un plan pour s'échapper, celle-ci s'autorise à mettre fin à la communication entre les deux détenus. De plus, Wendy peut aussi modifier les messages si elle le désire. L'utilisation de messages chiffrés éveillerait les soupçons ; ils seraient de plus, contraints par les autorités à divulguer leur clé de chiffrement. La seule alternative d'Alice et Bob est donc de s'envoyer des messages innocents et de dissimuler l'information compromettante dans ceux-ci. De fait, ils mettent en place un canal de transmission (par l'intermédiaire des messages eux-mêmes) qui n'est pas visible pour Wendy ; ce canal est appelé canal subliminal. La stéganographie permet alors de généraliser les techniques classiques de sécurité de transmission, telles l'étalement de spectre ou l'évasion de fréquence, a tout type de données. Réciproquement, l'étalement et spectre et l'évasion de fréquence peuvent être vus comme des techniques de stéganographie, dissimulant un signal dans de la bande passante ou le spectre des fréquences. Ces techniques visent par ailleurs à rendre furtives les transmissions mais aussi à se protéger contre un attaquant actif qui brouillerait le canal.

D'après [99] Alice et Bob se sont échangés au préalable une clé secrète cryptographique (ou ont accès à un serveur de clés publiques cryptographiques) ainsi qu'une clé secrète stéganographique (ou ont accès à un serveur de clés publiques stéganographiques). Nous appelons dans la suite médium support ou support de couverture le médium qui va contenir le message caché et stégo médium tout médium contenant de l'information cachée. Par abus de langage, il utilise aussi le terme de support et il dit qu'un médium est stégo si c'est un stégo médium et non stégo dans le cas contraire.

La mise en œuvre d'un schéma de stéganographie s'effectue alors en deux étapes distinctes [98].

Pour envoyer un message à Bob, Alice effectue les opérations suivantes :

1. elle compresse son message et le chiffre avec la clé cryptographique,
2. elle génère un support de couverture,
3. l'algorithme de stéganographie sélectionne les sous-parties du support favorables à la dissimulation,
4. il dissimule ensuite aléatoirement, à l'aide de la clé stéganographique, le message chiffré dans les parties favorables,
5. l'envoie du stégo médium par un canal classique [106].

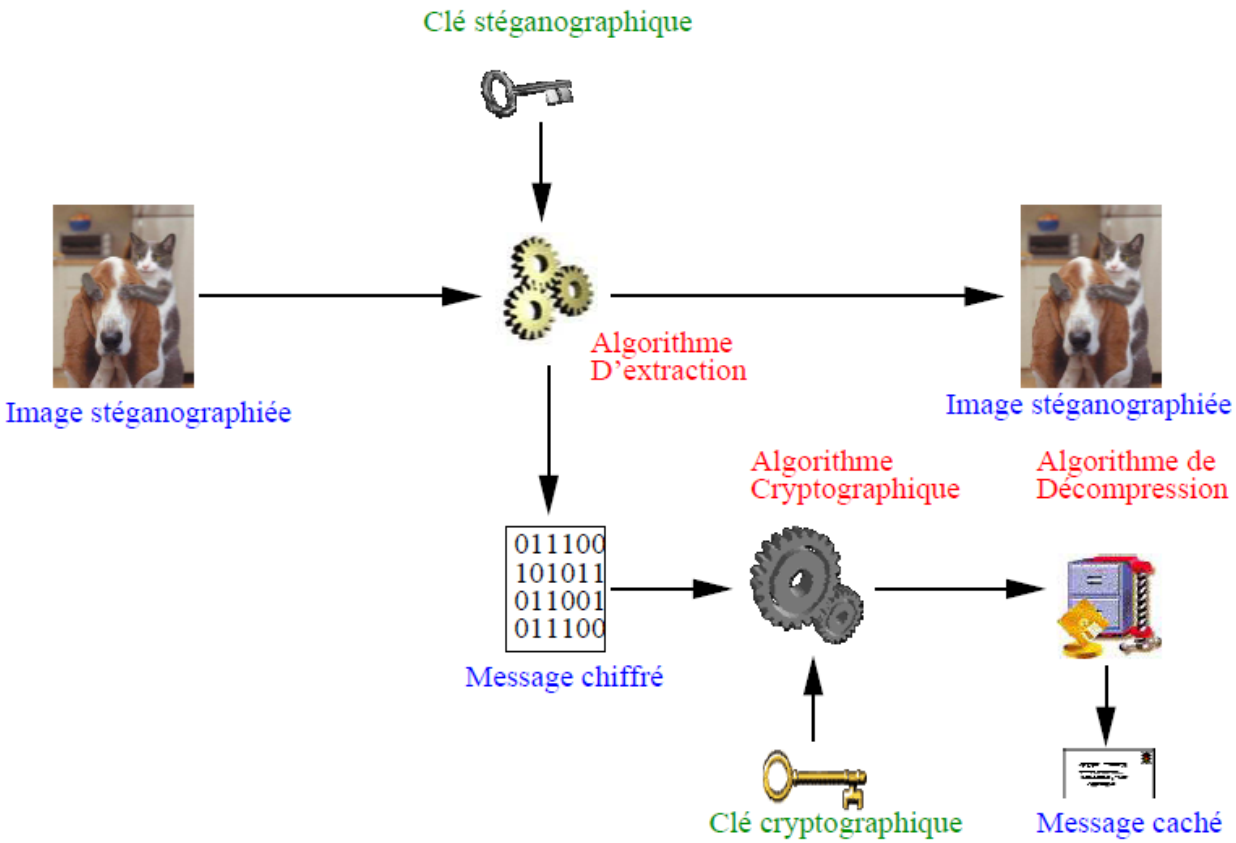


Figure 1.5 : Etape d'extraction pour une image fixe.

1.4.2. La stéganographie numérique (moderne)

Le Saint Graal du stéganalyste est avoir accéder à l'information échangée par Alice et Bob. Tel que, il doit tout d'abord spécifier les stego média des autres, puis extraire l'information dissimulée et enfin cryptanalyser le message chiffré. Selon A. Kerckhoffs [2], l'analyste ne dispose que des spécifications des schémas de stéganographie utilisés par Alice et Bob. Extraire l'information est équivalent à reproduire la suite de pseudo-aléa générée à l'aide de la clé stéganographique sans aucune connaissance ni de cette clé, ni même de la suite. Cryptanalyser le message chiffré est alors équivalent à une attaque a chiffré seul. Enfin, distinguer les stégo média des autres est semblable à trouver au moins une mesure statistique sur les média dont la distribution est différent suivant que le médium est stégo ou non. Aux vues des étapes précédentes, l'avantage soit définitivement acquis au stéganographe et franchir le stéganalyste.

BARBIER.J [86] Suppose de plus, que celui-ci dissimule dans un même support de couverture C deux messages m_1 et m_2 avec les clés stéganographiques respectives k_1 et k_2 pour obtenir le stégo médium ***S***. Suppose aussi qu'il existe un distingueur stéganographique idéal ; c'est- à- dire capable de détecter les stégo média avec une probabilité égale à 1. Une analyse de ***S*** avec ce distingueur indiquera qu'il contient de l'information dissimulée. Confondu, le stéganographe sera contraint de révéler une clé $\boldsymbol{k_i}$ et donc un message $\boldsymbol{m_i}$, i ∈ {1, 2}. Or, l'extraction de $\boldsymbol{m_i}$ consiste en une lecture de ***S*** ; ***S*** étant inchangé après l'extraction, le distingueur classifiera toujours ***S*** comme stégo médium, qu'il contienne plus d'information dissimulée ou non. En d'autres termes, quel que soit le distingueur stéganographique, celui-ci ne peut pas distinguer un stégo médium contenant exactement un message caché, d'un autre possédant plus d'un message dissimulé. Il a traduit cette propriété, *plausible (deniability)*, par *indistingabilité indéniable*. Aussi surprenne que cela puisse paraître, peu d'implémentations stéganographiques offrent cette fonctionnalité. Parmi les trois schémas de stéganographie qu'il a étudié, Outguess [133], F5 [61] et JPHide and JPSeek [68], seul Outguess propose de dissimuler deux messages en même temps, dans le même support et avec deux clés stéganographiques différentes. A la vue de cet exemple, il apparaît naturellement une règle d'or du bon usage de la stéganographie : il faut dissimuler un message sans importance en plus du message à envoyer, ce avec une clé stéganographique différente.

Nous nous sommes attachés principalement à étudier des techniques de stéganographie dédiées aux images fixes et plus particulièrement aux formats non compressés et au format JPEG. Bien que tout vecteur d'information soit potentiellement support d'une technique stéganographique, les images sont le vecteur le plus usité par les algorithmes de stéganographie [173].

1.5 Processus stéganographique

Dans un système stéganographique, il y a principalement deux processus. D'un côté un processus de dissimulation, de l'autre un processus de recouvrement. Un processus de dissimulation simplifié peut être donné par le schéma suivant :

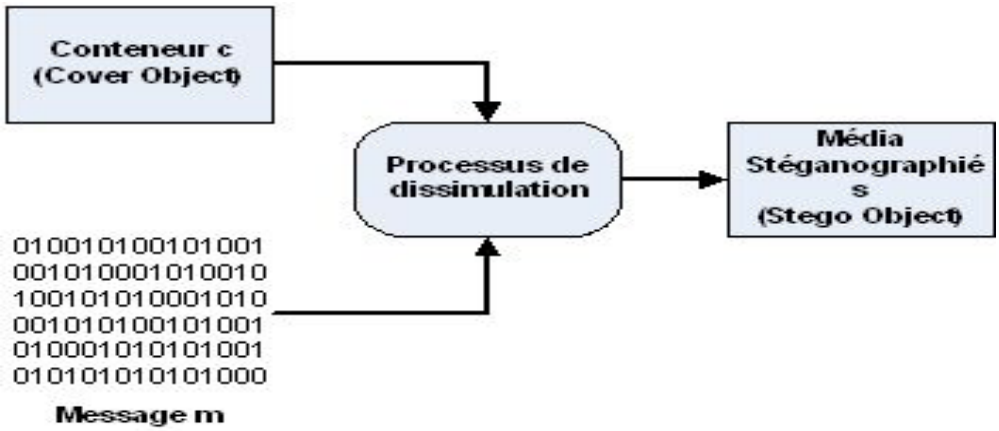


Figure 1.6 : Schéma simplifié de fonctionnement [118].

Il existe trois types de protocoles de stéganographie, correspondant de près à ce qui existe en cryptographie.

- *La stéganographie pure* est un système dans lequel les données secret a dissimulé ne trouve que dans l'algorithme utilisé. La découverte de cet algorithme détruit la dissimulation de la communication. Ceci revient à mettre en place de la "sécurité par l'obscurité".
- *La stéganographie à clé secrète* est similaire à la cryptographie symétrique, l'échange de données confidentielles nécessite, au préalable, l'échange d'une clé secrète que l'on ne partagera que avec notre interlocuteur. Il est donc nécessaire d'avoir un canal sécurisé, ou de rencontrer en personne notre interlocuteur, afin d'être certain que cette dernière ne soit pas compromise. Cette clé aura une influence sur la manière de "cacher" l'information.
- *La stéganographie à clé public,* quant à elle, est similaire à la cryptographie asymétrique. La personne voulant envoyer des données à un autre interlocuteur, sans éveiller de soupçons, utilisera la clé public de ce dernier. La clé public étant à priori connue de tout le monde, il n'y aura pas besoin d'échange préalable "sécurisé". La personne recevant ce message sera le seul à pouvoir en extraire son contenu à l'aide de sa clé privée. Voici un schéma plus complet du processus stéganographique :

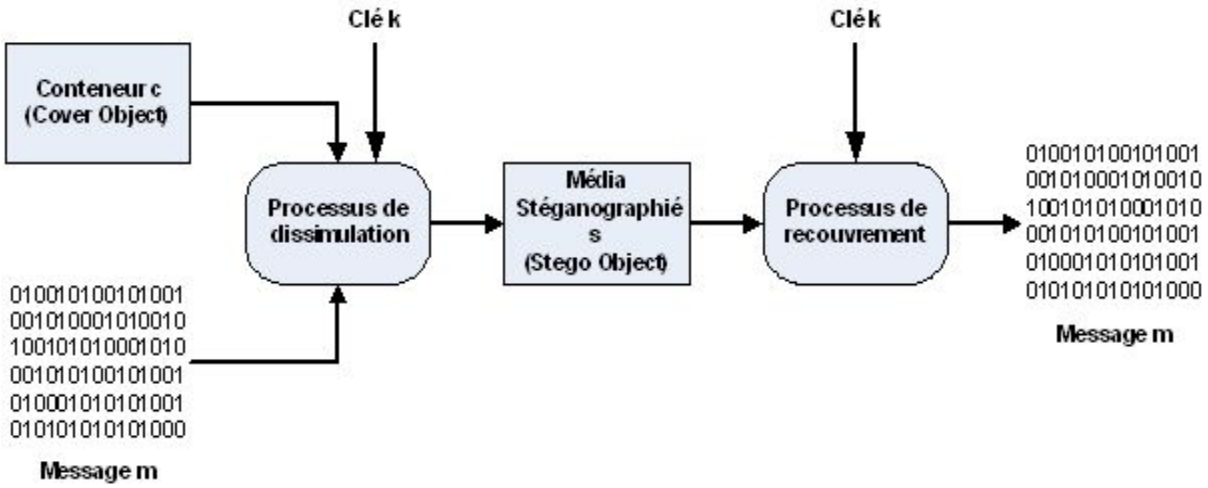
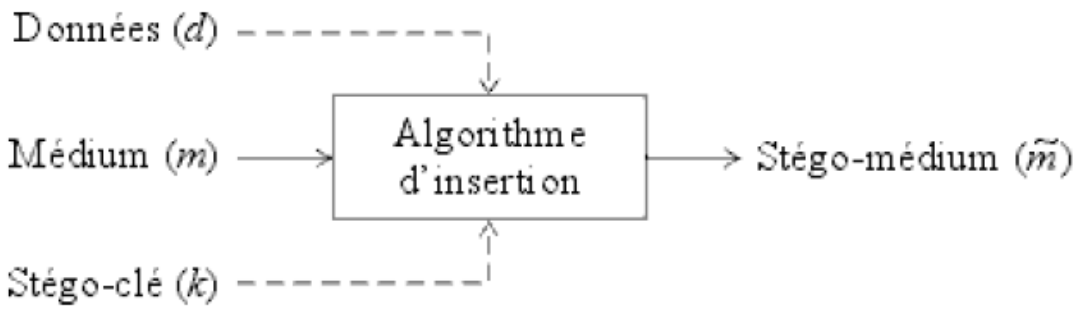


Figure 1.7 : Schéma complet [118].

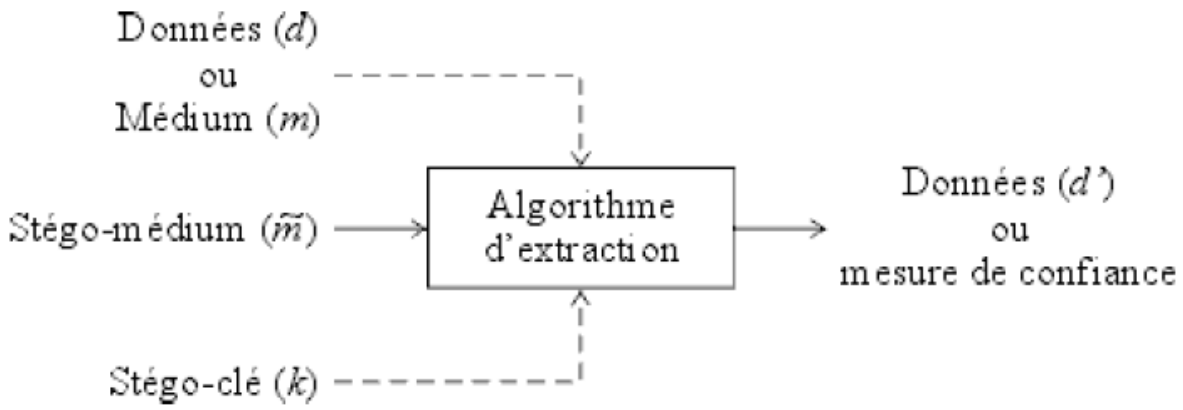
Le processus complet de la stéganographie repose sur deux opérations (figure 1.8) :

1. « La dissimulation », qui consiste à insérer l'information dans le medium comme illustre la figure 1.8.a ;

2. « L'extraction », qui récupère cette information. Le mot détection est également utilisé lorsqu'il s'agit de vérifier la présence d'une information (représentée grâce à un signal, une caractéristique particulière du medium) dans le stégo-medium, sans pour autant vouloir l'extraire comme illustre la figure 1.8.b.



(a) Insertion des données dans le médium



(b) Extraction des données du médium

Figure 1.8 : Principe générale de la stéganogrphie [99].

Trois critères permettent de classer les algorithmes stéganographique : La Capacité, la transparence et la robustesse.

- La Capacité correspond à la masse de données qui peut être insérée dans un conteneur, relativement à la taille de celui-ci.
- La transparence permet de quantifier le bruit généré par le processus de dissimulation, et par la même l'invisibilité de notre message.
- Pour finir, la robustesse spécifie la capacité qu'à notre message stéganographié de rester intacte après que le conteneur ait subit des modifications (filtrage, etc...).

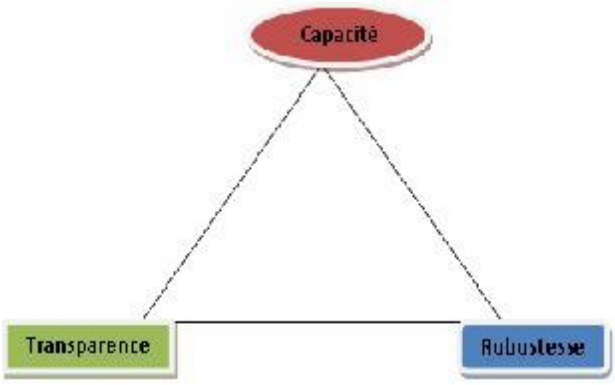


Figure 1.9 : Triangle des caractéristiques.

Ces trois critères ne peuvent pas être maximisés simultanément. Chacun d'entre eux aura une influence sur l'autre. Par exemple, la capacité va en contradiction avec la transparence**.**

Les outils de stéganographie dit naïfs correspondent à la grande majorité des outils disponibles sur internet. Ils cachent les informations dans les conteneurs sans réellement se préoccuper de la facilité à détecter ces données, ni les influences que ces données peuvent avoir sur le conteneur d'un point de vue statistique.

Les outils de stéganographie académique sont quant à eux développé par des équipes de recherches (notamment l'équipe de Fridrich). Leur objectif est de faire évoluer en parallèle stéganographie et stéganalyse , et d'arriver à des algorithmes totalement transparents (pour les méthodes actuelles), afin de pouvoir en déduire des méthodes de stéganalyse encore plus performantes. De récentes recherches portent sur la maximisation de l'espace de dissimulation disponible. Ces outils arriveront peut-être à allier transparence à capacité dans un avenir proche.

En stéganographie, le compromis qui nous intéresse est celui entre la capacité et L'indétectabilité. En effet, on considère que si le message est altéré, il sera réémis. L'objectif du stéganographe est bien d'envoyer le maximum d'information sans qu'un attaquant puisse le détecter. La notion de robustesse est plutôt importante pour le tatouage ou le marquage ; ceux-ci ne rentrant pas dans le cadre de notre étude. Le lecteur intéressé par le marquage d'image pourra trouver une bonne introduction dans les ouvrages [73, 74, 121].

1.5.1. Techniques Stéganographique

Sur Internet, de nombreux logiciels de stéganographie sont disponibles. On compterait à ce jour plus de 200 produits de stéganographie [5].

Tous ces produits ne jouissent cependant pas de la même popularité. Une analyse de chacun de ces logiciels n'étant clairement pas envisageable, se concentrer sur les plus populaires d'entre eux semble une bonne démarche.

Toujours avec pour objectif de mieux cibler le catalogue de produits existants, l'analyse est étendue au format de fichier ainsi qu'à la méthode stéganographique. Cette démarche permet de différentier ce qu'il est possible de faire, et ce qui est vraiment fait.

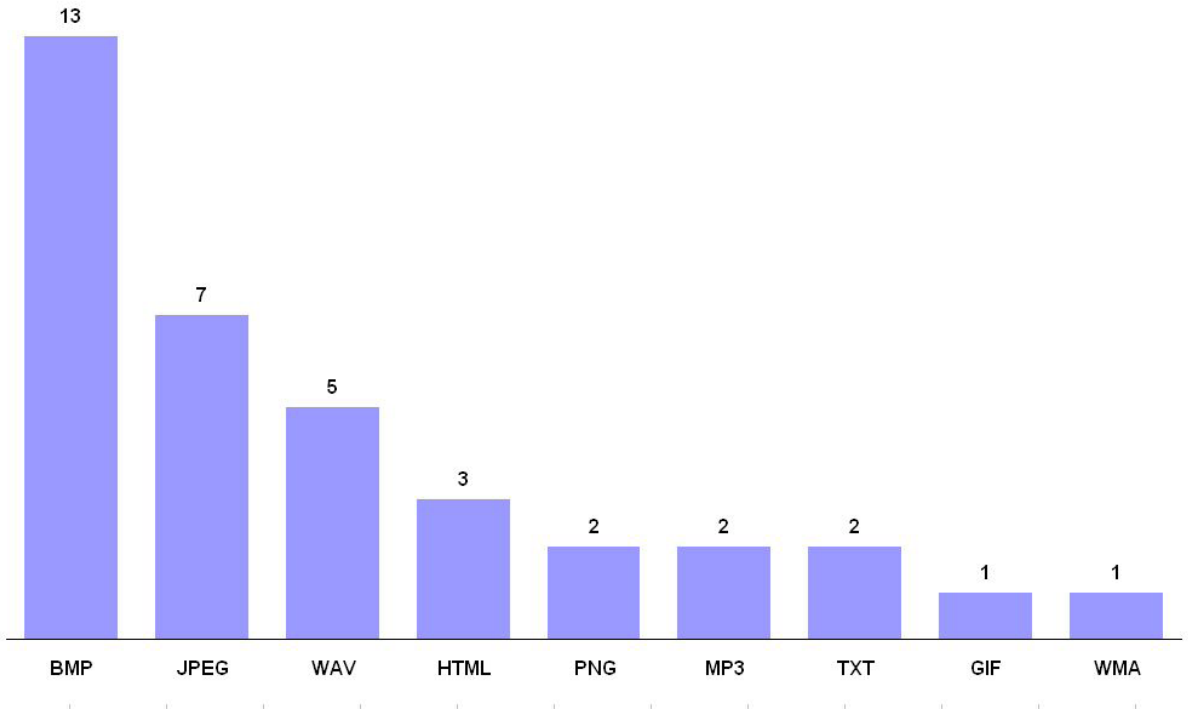


Figure 1.10 : Représentation des formats de fichier dans les produits stéganographique [5].

La figure 1.10 montre un graphique de la représentation des formats principalement utilisés. Les éléments qui peuvent être tirés de ce graphique sont, d'une part, que le format le plus utilisé est le BMP. Cela provient de la relative simplicité du format qui permet un décodage et une modification de l'image très aisée. Le format JPEG/JFIF arrive second. On comprend que les formats d'image restent les plus utilisés. Il n'y a que le format WAV qui soit utilisé dans des proportions significatives. Le graphique fait aussi apparaître des formats tels que le TXT ou l'HTML. Ces formats ne sont cependant pas traités dans ce document [99].

1.6. Les méthodes de dissimulation

Après avoir traité les produits et les formats privilégiés, il est intéressant de pouvoir déterminer quelles méthodes de dissimulations sont les plus largement utilisées par les produits de stéganographie. Un point très important est de pouvoir déterminer la méthodologie utilisée lors du processus de dissimulation. Le document [86] fait une synthèse de l'existant. En se basant sur ce document, l'ensemble des méthodologies utilisées actuellement pourrait être classées en 5 catégories :

– Steganographie dans le domaine spatial

– Steganographie dans le domaine des transformées

– Steganographie basée sur le document (Document Basedsteganography)

– Steganographie basée sur la structure du fichier

– Autres techniques.

1.6.1. Stéganographie dans le domaine spatial

La stéganographie dans le domaine spatial basé sur les modifications de LSB (Least Significant Bit), ainsi que une autre méthode connue sous le nom de BPCS (Bit Plane Complexity Segmentation).

La méthode du BPCS [25] a été proposée comme objectif de fournir une grande capacité de dissimulation. Elle sectionne l'image en petits blocs, qui sont ensuite définis soit comme porteur d'information, soit comme bloc de bruit. La dissimulation se base sur le fait que l'œil humain ne discerne pas bien les modifications intervenues dans des zones fortement bruitées (ou complexes). Ces dernières peuvent être modifiées sans que la perception de l'image ne change. La grande capacité est apportée par la modification possible dans tous les plans de bit. Contrairement à la méthode du LSB, qui est aussi applicable au fichier audio, Le BPCS n'est utilisable que sur des images.

Malgré ses apparentes qualités, Le BPCS n'est pratiquement pas utilisé dans les produits actuels. Il lui est souvent préféré la modification LSB, de par son implémentation très facile. Qtech-HV est le seul produit ayant été trouvé qui utilise cette technique. Il s'agit d'un logiciel développé à l'issue d'une étude menée sur le BPCS.

Dans le domaine spatial, le LSB reste largement la méthode la plus utilisée. Des produits tels qu'Invisible Secret,WbStego4Open et HermeticStego sont autant de logiciels utilisant cette méthode dans des implémentations très différentes.

1.6.2. Stéganographie dans le domaine des transformées

La stéganographie dans le domaine des transformées est majoritairement utilisée dans le cadre de dissimulations effectuées sur le format JPEG/JFIF. En lieu et place d'utiliser les pixels de l'image pour cacher l'information, ce sont les coefficients DCT de la transformée en Cosinus Discret qui sont utilisés. Comme dans le domaine spatial, la méthode principalement utilisée est celle des modifications de LSB, mais les modifications sont, cette fois, appliquées aux coefficients DCT et non plus directement aux valeurs des pixels.

L'énorme avantage de ce type de méthode est la large prolifération du format utilisé. En effet, les images au format JPEG/JFIF sont de loin les fichiers le plus présents sur le World Wide

Web. Tous les jours, des millions d'images JPEG/JFIF sont échangées via e-mail. Un échange de ce type est anodin.

La stéganographie dans le domaine des transformées sur support JPEG semble être une solution de choix. Toutefois, la complexité du format JPEG fait que très peu de logiciels dissimulent réellement les données de cette manière. La plupart des implémentations se contentent d'ajouter les données en fin de fichier.

Jsteg fut le premier produit à implémenter cette méthode. Les produits découlant du domaine académique, tels que F5 ou Outguess, suivent la même voie. Un seul produit commercial semble agir dans le domaine des transformées, il s'agit de SteganosPrivacy (anciennement Security).

1.6.3. Stéganographie basée sur le document

La stéganographie basée sur le document est utilisée pour les formats textuels. La dissimulation se fait à l'aide d'ajout de caractère tel que des espaces ou/et des tabulations en fin de ligne. Cela est utilisé pour coder l'information cachée. Les formats tels que l'HTML ou les fichiers TXT sont utilisés comme conteneur.

1.6.4. Stéganographie basée sur la structure du fichier

La stéganographie basée sur la structure du fichier se sert des espaces non utilisés pour la dissimulation de contenu. Cela est rendu possible par le fait que les décodeurs ne lisent pas les données contenues à ces endroits. Pour la plupart des formats, la dissimulation est effectuée en fin de fichier. Les spécifications de format définissent généralement une suite de bits indiquant la fin du fichier. Similairement, un champ dans l'en-tête du fichier spécifie la taille des données.

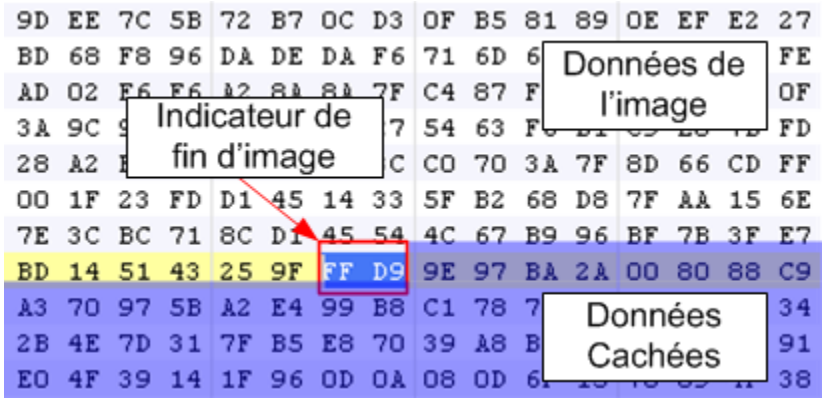


Figure 1.11 – Exemple d'ajout en fin de fichier (JPEG/JFIF).

Une autre approche est d'utiliser des champs spécifiés dans la norme du format. En prenant exemple sur les formats JPEG/JFIF 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/JFIF 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. Dès lors, en fonction de la taille du fichier caché, le bon conteneur devra être choisi pour ne pas éveiller de soupçons (un fichier PDF de 1 Mb « dissimulé » dans un fichier JPEG risque d'être suspect).

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

Remarque Certains logiciels sont faussement placés dans cette catégorie. Ils possèdent un fonctionnement proche, ajout de données en fin de fichier. Toutefois, cette opération est effectuée de manière aveugle. Il n'est donc plus question de stéganographie se basant sur la structure du fichier. Hiderman et Data Stash fonctionnent de cette manière [64].

Les méthodologies de dissimulation ne sont cependant pas figées aux seules catégories énoncées ici, de nombreuses autres existent, mais ne sont pas vraiment représentées dans les logiciels disponibles actuellement. Rien que dans les catégories énoncées plus haut, la représentation des produits n'est pas vraiment homogène. Les modifications dans le domaine spatial, ainsi que la stéganographie basée sur la structure de fichier sont les catégories qui semblent être les plus représentatives de l'offre logiciel actuelle.

Dans le domaine spatial, seul la modification LSB est réellement utilisée. Le BPCS semble à priori prometteur, mais n'est pas utilisé. Il semble que la plupart des acteurs de ce marché souhaitent proposer un logiciel sans vouloir investir beaucoup de temps à la conception de ce dernier. La modification LSB effectuée dans le domaine spatial est très facile à implémenter, contrairement au BPCS qui requiert plus de temps.

Le constat est le même hors du domaine spatial. Si l'on considère les applications effectuées au format JPEG, force est de constater que les applications agissant dans le domaine des

transformées ne sont pas très courant. Une fois de plus, l'écriture d'un tel logiciel requiert le développement (ou la modification) d'un encodeur/décodeur JPEG, tâche qui se révèle très ardue. Alors que le simple ajout de données en fin de fichier ne requiert aucun développement préalable. Mise à part le format BMP, les implémentations existantes dans d'autres formats usent majoritairement de la structure du fichier et non pas des données pour dissimuler un message. Quand cela ne se résume pas simplement à appondre les données à la fin d'un fichier, peu importe sa structure.

En résumé, mise à part les produits venant du monde académique tels que F5, Outguess ou encore Qtech-HV, il semble que peu de sociétés investissent du temps et de l'argent dans le domaine de la stéganographie.

On va détailler quelques techniques stéganographique utilisée couramment dans les images. A noter que certaines de ces techniques sont transposable aux autres domaines que sont l'audio et la vidéo. Selon [5], il est possible d'établir une hiérarchie au niveau des techniques stéganographique.

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 de fusion. Pour ce qui est du reste, ils peuvent être regroupés dans ce qui a trait à la modification LSB.

1.7. La stéganographie pour les images

La stéganographie pour les images binaires [177, 79] se focaliser sur les données, en cache les images en niveaux de gris et en couleur. Éta1nt donné que la composante de luminance d'une image en couleurs est équivalente à une image en niveaux de gris,. Par ailleurs, il est généralement considéré que les images en niveaux de gris sont plus appropriés que la couleur d'images pour cacher des données [6], parce que la perturbation de

corrélations entre les composantes de couleur peuvent facilement révéler la trace de l'intégration. Nous nous concentrons sur les images en niveaux de gris et bien précisément à la stéganographie JPEG.

1.7. 1. La technique LSB

La technique du LSB, signifiant Least Significant Bit, est la technique la plus répandue. Son succès provient d'une grande facilité de mise en œuvre, ce qui permet d'en trouver de nombreuses implémentations.

Sous l'appellation LSB est regroupé tout ce qui a trait à la dissimulation de données par la modification du bit de poids faible d'un élément. Cela va de la valeur d'un pixel, jusqu'à la modification de la valeur d'un coefficient DCT dans le cas de la norme JPEG. Tous se basent sur l'insensibilité du système visuel humain à un faible changement de couleurs [128].

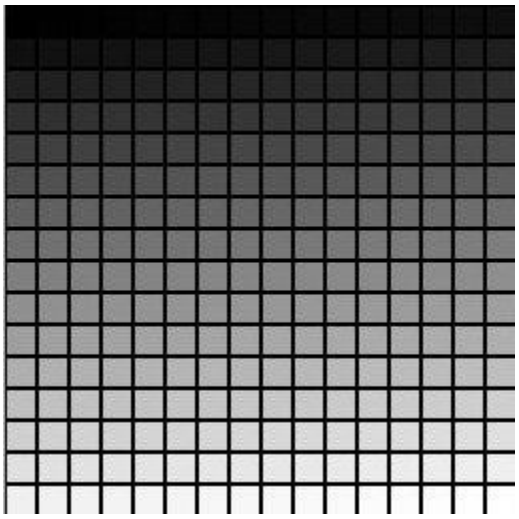


Figure1.12 : mire de 256 niveaux de gris.

Comme cela peut être mis en évidence à l'aide de la figure 1.12, le changement du bit de poids faible correspond à un déplacement horizontal d'une case dans la mire. Aucun changement n'est perceptible.

Plusieurs types de modifications peuvent être effectués sur ces LSB. La plus répandue consiste simplement à remplacer ces bits par les bits du message que l'on souhaite dissimuler. Appelée "LSB Replacement" dans la littérature, cette technique semble à première vue très efficace, elle possède le gros désavantage de modifier de manière significative les statistiques

du conteneur. Un exemple de cette méthode peut être donné à l'aide de la matrice8 C suivante représentant un conteneur de 4 éléments sur 4 [88].

$$C = \begin{bmatrix} 00 & 00 & 10 & 10 \\ 01 & 11 & 10 & 00 \\ 00 & 11 & 10 & 00 \\ 00 & 00 & 00 & 00 \end{bmatrix}$$

Pour simplifier, selon [99] chaque élément est codé sur 2 bits uniquement. Si l'on souhaite dissimuler le message m dans votre conteneur, une première remarque sera sur la taille maximum du message qui sera égal au nombre d'éléments dans la matrice. Cela est vrai pour autant que l'algorithme se limite à la modification du seul bit de poids faible [54] .

$$m_{max} = 1001\,1010\,0011\,1001$$

Dans le cas du conteneur C, la taille maximale est donc de 16 bits. La matrice stéganographié résultant du processus sera

$$S_{replacement} = \begin{bmatrix} 01 & 00 & 10 & 11 \\ 01 & 10 & 11 & 00 \\ 00 & 10 & 11 & 01 \\ 01 & 00 & 00 & 01 \end{bmatrix}$$

En comparant S à C, on remarque que l'opération consiste donc uniquement en une sur écriture du bit de poids faible.

La seconde méthode est appelée "LSB Matching". Elle diffère de ce qui précède par le fait qu'elle ne modifie pas obligatoirement tous les bits de poids faible. Le principe consiste à comparer la valeur du bit de poids faible à la valeur du bit à dissimuler. S'ils correspondent, aucun changement n'est effectué. Dans le cas contraire, une incrémentation/décrémentation de manière aléatoire de la valeur de l'élément de 1 sera effectuée. Cela aura pour incidence de codé la valeur désirée au niveau du LSB.

En reprenant la même matrice C que précédemment,

$$C = \begin{bmatrix} 00 & 00 & 10 & 10 \\ 01 & 11 & 10 & 00 \\ 00 & 11 & 10 & 00 \\ 00 & 00 & 00 & 00 \end{bmatrix}$$

Dans laquelle le message m_{max} suivant est à dissimuler.

$$m_{max} = 1001\ 1010\ 0011\ 1001$$

Le résultat de l'application de cette méthode à la matrice de base C sera cette fois

$$S_{matching} = \begin{bmatrix} 01 & 00 & 10 & 01 \\ 01 & \mathbf{00} & 11 & 01 \\ 00 & 10 & 11 & \mathbf{11} \\ 01 & 00 & 00 & \mathbf{11} \end{bmatrix}$$

Cette matrice est un des résultats possible, le choix entre l'incrémentation et la décrémentation étant fait de manière aléatoire. En gras sont mis en évidence des erreurs pouvant provenir du processus. Par exemple, l'incrémentation d'une valeur 11 aura pour conséquence sont passage à 00. L'écart des valeurs étant trop élevé, le résultat de l'opération risque d'être visuellement décelable. L'algorithme doit donc veiller à ce que ce genre de situation ne soit pas autorisée (autorisé uniquement l'incrémentation lorsque la valeur est la plus petite possible par exemple).

Comparativement à la méthode précédente, cette dernière est moins sensible aux analyses statistiques. Du faite de l'incrémentation/décrémentation aléatoire des valeurs des éléments, cette méthode n'ajoutera pas les mêmes distorsions statistiques que le "LSB Replacement".

Dans les deux cas, le choix de l'image hôte est un point essentiel dans l'optique d'obtenir la meilleure transparence possible. Les images contenant peu de couleurs sont à proscrire.

Certains experts recommandent l'utilisation d'image en niveau de gris comme meilleurs V conteneurs. Il conseille l'utilisation d'images non compressées.

Par la suite vont être détaillées les deux plus grands domaines d'application de la technique du LSB que sont, d'une part, le domaine spatial et de l'autre, le domaine de la transformée en cosinus discret.

A. Domaine Spatial

Dans le domaine spatial, la dissimulation du message est directement effectuée au niveau du codage des pixels. Une image peut être représentée à l'aide d'une matrice de pixel. Chaque pixel est représenté à l'aide de 1 à 32 bits. Ce nombre dépend de la représentation des couleurs utilisées. Les plus utilisées sont cependant :

– Sur 8 bits

– Sur 24 bits

Sur 8 bits, deux méthodes de codage sont utilisées. Dans le cadre d'une image en niveaux de gris, chaque pixel code directement le niveau de gris approprié. Dans celui d'une image couleurs, utiliser le mécanisme d'image indexée. Au sein de chaque pixel est codé une valeur correspondant à l'index de la couleur a utilisé dans la palette de couleurs définie.

Pour ce qui est du codage sur 24 bits, utilise généralement le RGB (Red Green Blue) pour la représentation des couleurs. Chaque pixel est codé à l'aide d'un triplé de byte spécifiant chacune des composantes principales. Dans ce type de représentation, il est possible de dissimuler 3 bits par pixel (1 par composante) sans aucun impact visuel.

Figure1.13 : A gauche, image originale. A droite, image contenant un PDF de 32 Kb dissimulé à l'aide d'Invisible Secrets 4.

Les fichiers généralement utilisé sont au format BMP et GIF. Cette technique est très utilisée car elle facile à mettre en œuvre. Elle permet de directement ajouter les données à l'image, sans avoir à passer par un mécanisme de compression/ décompression comme ce serait le cas pour la modification des coefficients DCT.

Crée sa propre implémentation devient très facile, cela permet de s'affranchir de l'utilisation d'un produit existant et par la même occasion, d'éviter la détection de signature étant attribuée à ce programme (technique encore très utilisée dans les outils de détection). De plus, il est possible de vraiment contrôler la manière dont seront dissimulées les données au sein du conteneur. Il sera possible de facilement implémenté des mécanismes permettant de déjouer les tentatives d'analyse statistiques.

Les images au format JPEG sont, de loin, les plus répandue sur Internet. Ainsi, les outils permettant la dissimulation de données dans le format JPEG constitue une des meilleures solutions offertes. Le simple fait d'envoyé une image au format BMP peut attirer l'attention, et de ce fait, mettre à néant tous les efforts de dissimulation de données. La prochaine section va en détailler l'idée générale du format JPEG.

1.7.2 Transformée en Cosinus Discret (DCT)

Comme souligné dans la section précédente, les images au format JPEG [28] représentent la grande majorité des images circulant sur le réseau. Une image envoyé à son collègue en utilisant ce format est devenu quelque chose d'anodin. Dans ce contexte, ce format semble être un conteneur de choix pour des communications secrètes. La dissimulation d'information dans des formats de compression à perte se révèle, cependant, plus difficile. Ce type de compression utilisant les mêmes données redondantes (n'ayant aucun impact sur la perception du média) que celle utilisée dans le processus de dissimulation.

Afin d'assurer un maximum de transparence, les modifications doivent avoir lieu dans le domaine des Transformées en Cosinus Discret [86] et non dans le domaine spatial. D'un point de vue compression cela permet d'éliminer les hautes fréquences (saut brusque de couleur) qui ne sont que difficilement décelable par l'oeil humain.

En ce qui concerne la stéganographie par modification des coefficients DCT, elle intervient après la phase de quantification de l'algorithme de compression (voir figure 1.14).

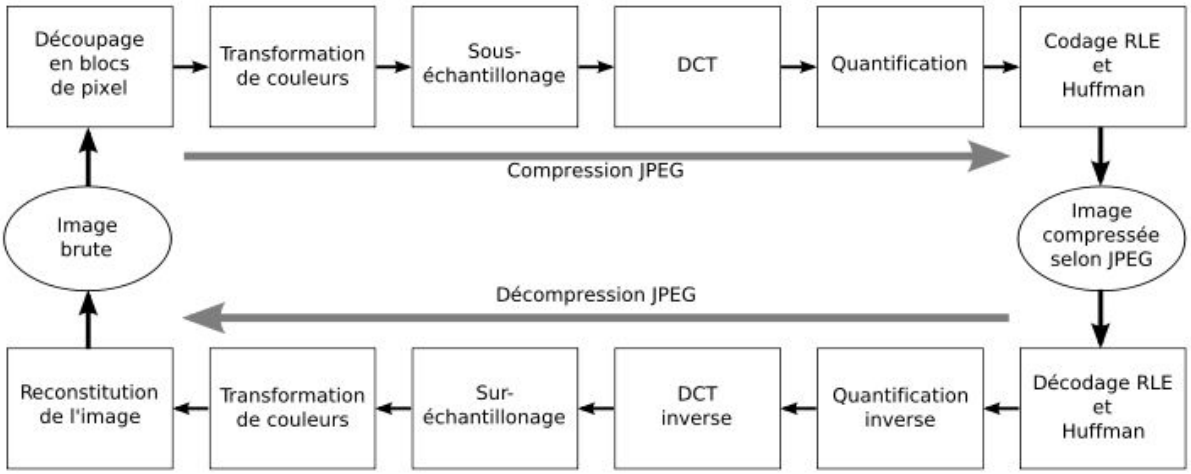


Figure 1.14 : Schéma Bloc du processus de compression/décompression JPEG [28].

Pendant ce processus, la plupart des coefficients étant ramenés à zéro, tout effort de dissimulation intervenant avant la quantification serait rendu inutilisable. Un autre facteur à prendre en compte, et de ne pas modifier les propriétés de compression. Après quantification, les matrices de coefficients passent par un algorithme de compression sans perte, le codage de Huffmann. Afin de ne pas modifier les informations, aucun coefficient étant égal à 1 ou 0 ne devra être modifié. Si l'on remplace un coefficient à 1 par 0, cela engendrera un meilleurs taux de compression, ce qui n'est pas souhaité.

De cela découle un des gros inconvénients de cette méthodologie. Une matrice de coefficients étant principalement composée de 0, et ceux-ci ne pouvant pas être modifiés, la capacité du conteneur s'en retrouve fortement réduite. De plus, contrairement à la modification dans le domaine spatial, implémenté une telle technique se révèle plus ardu. Il est en effet nécessaire de coder tout le compresseur JPEG, ce qui n'est pas une tâche aisée. La dissimulation dans les coefficients DCT ne se révèle pas non plus performante d'un point de vue transparence. Quelques méthodes de stéganalyse dans le domaine spatial ont rapidement été adaptées au domaine DCT [110].

A. Modification des coefficients DCT

Les images JPEG utilisent la Transformée en Cosinus Discret (DCT) pour accomplir la compression. Les données compressées sont stockées en tant qu'entier ; cependant, les calculs pour le traitement de la quantification exigent des calculs en virgule flottante ce qui donne des arrondis. L'erreur introduite par l'arrondi définit le caractère de compression avec perte du

format JPEG. L'information est cachée dans l'image JPEG en modulant les choix d'arrondi dans les coefficients DCT. A l'encapsulation on suit le schéma de la figure 1.15.

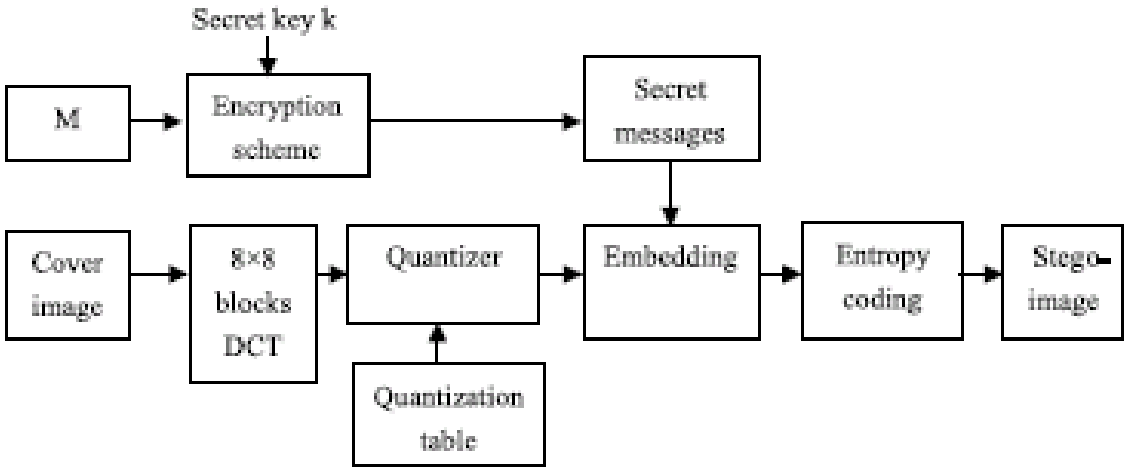


Figure 1.15 : Insertion d'information avec une image au format JPEG.

A l'extraction du message on suit le schéma de la figure 1.16.

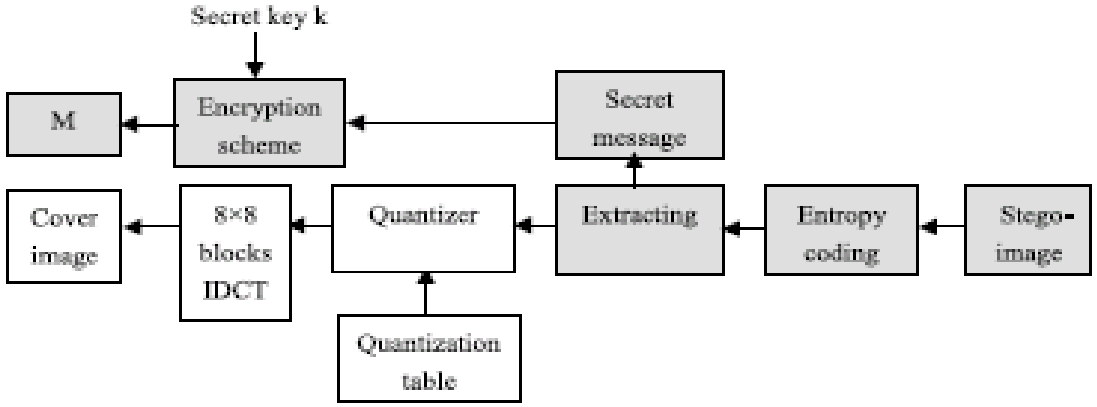


Figure 1.16 : Extraction de l'information.

E. Koch et J. Zhao proposent l'utilisation de la transformée en cosinus discret dans le schéma suivant :

1. Sélection et extraction de blocs de 8 x 8 pixels (les emplacements dépendent d'une clé secrète) ;

2. Pour chaque bloc :

(a) modification de tous les pixels en fonction du bit à inscrire : leur valeur est augmentée ou diminuée selon que le bit vaut 0 ou 1 ;

(b) normalisation des valeurs des pixels, de sorte qu'elles soient toute comprises dans un intervalle fixé (par exemple [-127..127]) ;

(c) application de la transformée en cosinus discret ;

(d) choix d'un certain nombre de coefficients ;

(e) modification de ces coefficients de telle sorte qu'ils vérifient une certaine relation d'ordre ;

(f) application de la transformée en cosinus discrète inverse ;

3. Réintégration des blocs dans l'image.

La clé secrète réside dans l'emplacement des blocs, mais aussi dans le choix des coefficients que l'on modifie. Cette technique permet d'inscrire environ dix fois plus d'informations que la technique de régions de couverture et bit de parité.

Algorithme Outguess

Développer par Niels Provos, [84], cette algorithme avait comme objectif de passer totalement inaperçu lors d'une analyse statistique des 2. Il travaille sur les fichiers de type JPEG en effectuant un sur écriture des LSB au niveau des coefficients DCT [85]. Cependant, pour ne pas modifier les propriétés de compression, il modifie uniquement les coefficients étant différent de 1 ou 0.

Afin de paraître invisible lors d'analyse statistique du premier ordre (analyse des histogrammes de valeur DCT), l'algorithme opère en deux étapes :

En première passe, il modifie les LSB des coefficients DCT de manière pseudo-aléatoire.

Ceci est défini à l'aide d'une clé.

En deuxième passe, il parcourt les coefficients DCT non modifiés afin d'adapter leur valeur de tel sorte que l'histogramme des coefficients après modification soit égal à celui du fichier d'origine.

Afin de pouvoir être sûr de retrouver l'histogramme d'origine après la manipulation, il effectue un calcul de la taille maximal des données à dissimuler en fonction de l'image. Ceci est, bien sûr, effectué avant le début des opérations.

Figure1.17 : a) image originale. B) image contenant le fichier outguess.hstéganographié avec Outguess [83].

Comme il peut être remarqué sur la figure 1.17, aucune différence n'est visuellement apparente. Fridrich [83] a défini une méthode permettant de détecter de manière viable les contenus stéganographié à l'aide d'Outguess.

1.8 Conclusion

La stéganographie numériques, lorsque il est adaptée aux données numériques (images, son,..). Elle suit depuis des années 90 un démarrage corrélé à celui d'Internet ;

L'image est l'une des supports les plus largement utilisé par la stéganographie. La raison de l'importance de ce type de support dans le domaine de dissimulation secrète réside dans les différents types d'images numériques. Le chapitre qui suit décrit la stéganographie appliquée aux images JPEQ.

Chapitre 2

Stéganographie adaptée aux images du format JPEG

2.1 Introduction

Dans ce chapitre, nous nous intéressons à la stéganographie appliquée aux images JPEG. Ce format est l'un des vecteurs d'information les plus usités et les plus présents sur Internet. La stéganographie souhaitant dissimuler l'existence même de l'information qu'il veut transmettre va naturellement se tourner vers les supports les plus représentés afin de « noyer » son message caché dans la masse. Les images fixes sont donc les média de couverture les plus prisés par les logiciels de stéganographie. De nombreux formats d'image sont disponibles sur Internet, mais là encore, deux d'entre eux semblent être majoritaires.

Les formats non compressés permettent d'échanger des images sans dégradation. Mais de nombreuses applications nécessitent des images de qualité supérieure (images médicales, photographie de qualité, photos satellites . . .), et les fichiers associés aux formats sont néanmoins de taille beaucoup plus importante. Parmi les formats d'images compressées, le JPEG [65] est bien sûr le plus répandu et s'est imposé comme standard de facto. Le format JPEG est en effet un bon compromis entre la qualité et la taille du fichier. Il est de plus supporté par la majorité des applications et laisse facilement le choix à l'utilisateur de la qualité qu'il souhaite obtenir.

La plupart des algorithmes de stéganographie sont adaptés pour les formats non compressés et le format JPEG. Dans un premier temps, nous décrivons le domaine spatial, (l'image non compressée) ainsi que l'algorithme de stéganographie pour lequel nous présentons une stéganalyse au chapitre 3. Nous détaillons ensuite finement le format de compression JPEG afin de comprendre les mécanismes sous-jacents et les contraintes imposées pour utiliser les fichiers JPEG comme média de couverture. Les descriptions du format JPEG et des algorithmes sont centrales dans notre étude car elles permettent non seulement de se familiariser avec les algorithmes attaqués mais aussi d'appréhender la méthodologie que nous avons adoptée pour notre stéganalyse.

2.2 La Stéganographie dans le domaine spatial

2.2.1 Les images non compressées

Les images fixes non compressées apparaissent dans de nombreux formats, notamment BMP, Raw, X Pixmap . . .. 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 . . .). L'image non compressée est composée d'une succession de points

appelés pixels. Elle est alors en noir et blanc, en niveaux de gris ou en couleurs selon le nombre de bits nécessaires à coder chaque pixel [155].

Pour une image en noir et blanc, chaque pixel est codé sur 1 bit ; la valeur 0 pour un pixel noir et la valeur 1 pour un pixel blanc. Pour une image en niveau de gris un octet permet de coder les 256 niveaux de gris que peut prendre le pixel. Les pixels des images couleurs sont codés sur au moins 24 bits sous forme de coordonnées dans un espace de couleurs. Il existe plusieurs façons de coder la couleur et donc plusieurs espaces de couleurs. Le choix d'un espace de couleurs dépend essentiellement des applications mises en œuvre, comme par exemple la photographie, la télévision, l'impression, etc. Le lecteur intéressé par une description exhaustive des différents modèles de représentation de la couleur pourra se référer à [30]. Un des espaces de couleur le plus usité pour les images fixes est l'espace RVB. Dans cet espace, chaque couleur possède trois composantes qui correspondent respectivement à une intensité de rouge (R), de vert (V) et de bleu (B). Chaque composante est additionnée pour donner la couleur finale. L'espace RVB est un espace en 3 dimensions et peut se représenter par un cube RVB, comme l'illustre la figure 2.1.

Figure 2.1 : représentation d'image en noir et blanc, niveaux de gris et couleurs.

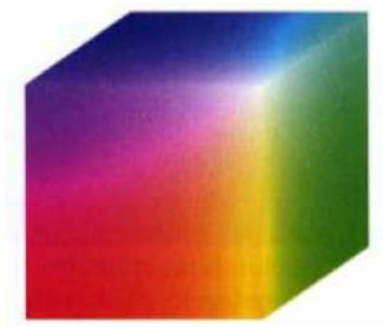


Figure2.2 : Cube RVB.

Il est néanmoins possible de stocker des images couleurs avec des pixels codés sur moins de 24 bits. Par exemple, pour stocker des images couleur sur 16 bits nous devons constituer un dictionnaire de 2^{16} entrées, appelé *palette*. A chaque pixel de 16 bits est associé un entier de 16 bits qui fait référence à une entrée de la palette et à chaque entrée est associé un triplé (*r, v, b*) de 24 bits. La palette doit évidemment être codée et stockée dans le format de l'image. Les fichiers GIF, par exemple, utilisent une palette de 256 entrées. L'image de la figure 2.2 possède une palette de 16 couleurs.

2.2.2 La Stéganographie LSB

La stéganographie LSB (Least Significant Bit) consiste a dissimuler l'information dans des bits de poids faibles d'un support. Cette technique est un cas particulier de stéganographie +/-k, qui incrémente ou décrémente les valeurs du support de +/-k , la stéganographie LSB adaptée aux images fixes non compressées est l'une des premiéres techniques stéganographiques et peut-etre meme l'une des plus employées encore aujourd'hui.

Malheureusement, la stéganographie LSB est sujette a de nombreuses attaques, tout comme la stéganographie +/-k. Parmi les inombrables stéganalyses, nous pouvons notamment citer la stéganalyse RS due à J. Fridrich [89], qui permet non seulement de détecter l'usage de stéganographie LSB mais aussi d'estimer la longueur du message, les attaques du type $\chi 2$ dues à A. Westfeld et A. Pfitzmann [204], l'analyse par paires de S. Dumitrescu et al. [100] et améliorée par P. Lu et al. [134] ou d'autres toutes aussi efficaces proposées par S. Lyu et H. Farid [161] ou A.D. Ker [100].

Figure 2.3 : Image couleur avec une palette de 16 couleurs.

Dans le cas d'une image non compressée codée sur 24 bits, chaque pixel est décrit par trois octets.

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

…

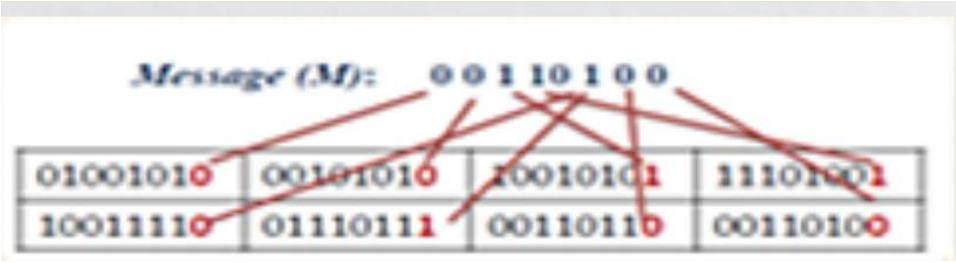


Figure 2.4 : Stéganographie LSB pour une image non compressée

Les algorithmes d'insertion et d'extraction sont résumés dans les figures C.5 et la figure C.6.

La notion *de capacité d'un algorithme stéganographique* est très proche de celle donnée par Shannon [158]. En effet, si l'on considéré le support comme un canal de transmission, la quantité d'information que l'on peut dissimuler est donc bornée. Dans cet esprit, un algorithme de stéganographie peut être vu comme un algorithme de codage et la capacité du support est donc soumise à la borne de Shannon. Une approche de la stéganographie sous l'angle de la théorie de Shannon a été proposée par C. Cachin [43, 80, 41] et dans ce modèle,

R. Chandramouli et N.D. Nemon [148, 149] ont défini la notion de capacité. En pratique, un même support possède une capacité différente selon l'algorithme de stéganographie utilisé.

Afin de pouvoir comparer la quantité d'information que peuvent dissimuler les algorithmes de stéganographie, selon [99] il utilise le taux stéganographique, c'est- à -dire rapport entre la taille du message à dissimuler et la taille du support. Ce taux peut s'exprimer en pourcentage ou en nombre de bits de message dissimulés par pixel (bpp). Par la suite, ila fait le choix de l'exprimer en pourcentage. La figure 2.5 met en évidence les pixels affectés par l'insertion d'un message par MBPIS. Les images supports (ligne du haut) sont comparées avec les stégo images (ligne du milieu) pour différents taux stéganographiques. Les images de la dernière ligne, correspondent à la différence entre le support et le stégo médium. Les pixels blancs représentent les pixels inchangés, ceux de couleur apparaissent pour des pixels changés par MBPIS selon la composante de même couleur.

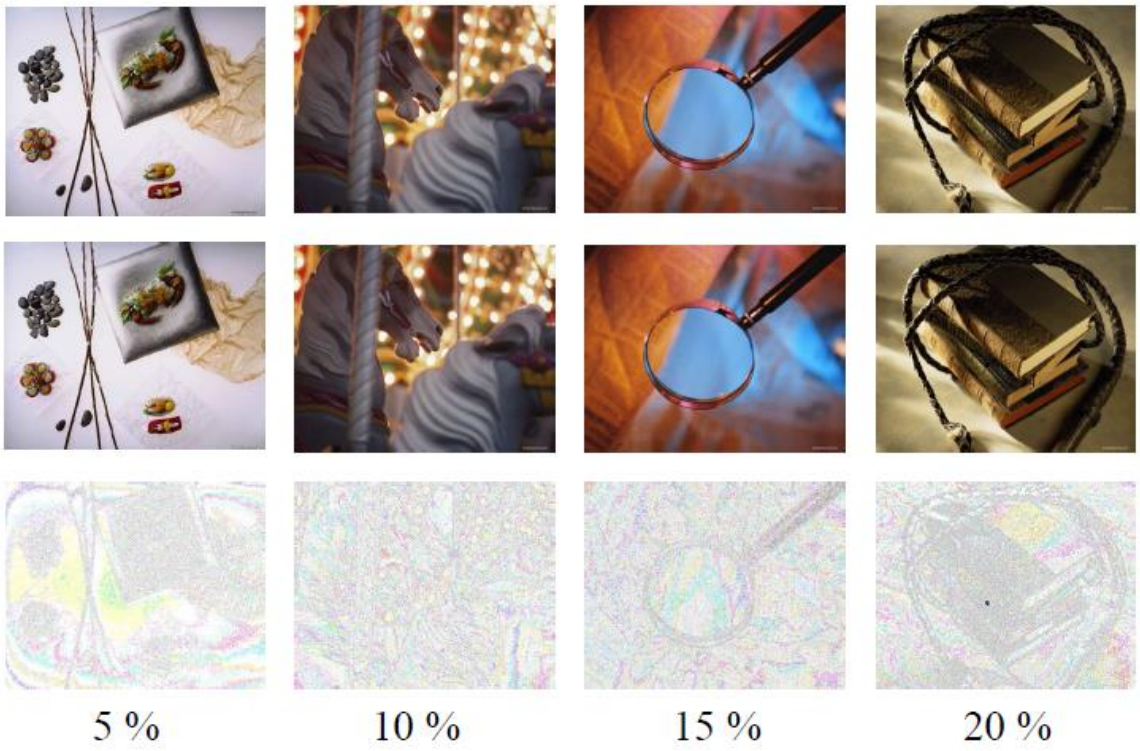


Figure 2.5 : Images stéganographie par MBPIS pour différents taux [155].

2.3 Le format JPEG dans la stéganographie

JPEG est l'acronyme de « Joint Picture Experts Group ». Ce groupe d'experts s'est formé en 1986 sous l'impulsion de l'ISO (International Standards Organisation) et de l'ITU (International Telcommunication Union) afin de travailler à l'élaboration d'un standard de compression pour les images fixes en nuances de gris ou en couleurs. Ses travaux ont débouché sur deux standards de compression : la norme T.81 pour l'ITU et la norme 10918-1 pour l'ISO. Par abus de langage, JPEG désigne aujourd'hui ces standards internationaux de compression.

Ce chapitre présente succinctement les grandes étapes qui composent la compression JPEG, son objectif est de permettre la compréhension des techniques de stéganographie et de stéganalyse adaptées au format JPEG. Les puristes nous en excuseront et pourront retrouver l'ensemble des informations et précisions concernant le format JPEG, notamment dans les normes suscités mais aussi dans [98, 30]. Le lecteur pourra aussi se référer à une description très complète et en français du format JPEG [110]. La compression JPEG s'organise suivant les cinq grandes étapes qui sont : **1** : le changement de l'espace des couleurs, **2** : la

transformation en cosinus discrète (DCT), **3** : la quantification, **4** : le codage RunLengthEncoding (RLE) et **5** : la compression de Huffman comme illustre la figure 2.6.

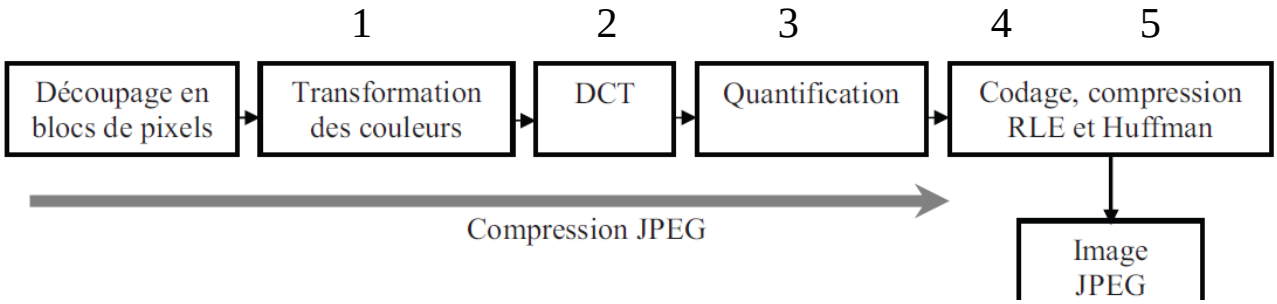
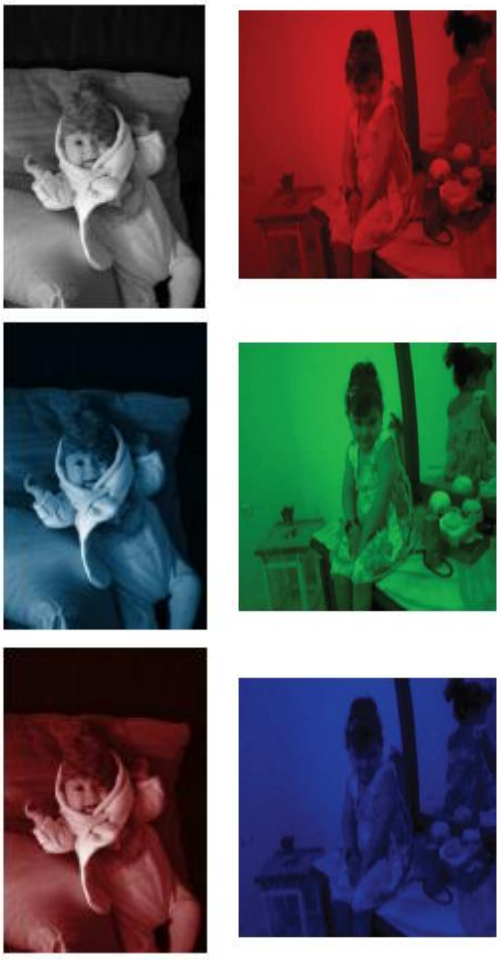


Figure 2.6 : Schéma de compression JPEG [21].

2.3.1 L'étape 1 : Changement de l'espace des couleurs

Soit une image I que l'on veut compresser au format JPEG. Chaque pixel p_i, est représenté par un triplet (R_i, V_i,B_i) dans l'espace RVB. La première 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_i,Cb_i,Cr_i), ou Y_i désigne la luminance, c'est- à- dire l'intensité lumineuse, Cb_i la chrominance bleue, c'est- à -dire l'intensité de la couleur bleue et Cr_i la chrominance rouge, c'est- à -dire l'intensité de la couleur rouge.



(a) (b)

Figure 2.7 : Décomposition suivant les composantes YCbCr (a) et RVB(b).

Le changement d'espace de couleurs se justifie par le fait que l'espace YCbCr est très proche du fonctionnement de l'œil humain. De plus, ce dernier est très sensible aux variations de luminance et très peu sensible aux variations de chrominance. De ce fait, on pourra effectuer des modifications sur les composantes Cb et Cr afin de compresser l'information visuelle et cela, sans que l'œil ne détecte la différence. Pour ce faire, les pixels sont regroupés en blocs de 4×4 pixels. Les quatre valeurs de chrominance bleue sont remplacées par leur moyenne ; la même transformation est effectuée sur les quatre valeurs de chrominance rouge. L'œil ne perçoit pas les changements effectués. Le gain de stockage est de 50% et la transformation appliquée est non réversible, la compression est dite avec perte. Cette transformation est appelée sous-échantillonnage et illustrée par la figure 2.8.

Chacune des valeurs Y, Cb, Cr est un nombre codé sur P bits, compris entre 0 et 2^P-1, où P est appelé *précision*. Les valeurs Y, Cb et Cr sont alors ramenées sur l'intervalle $[2^{P-1}+1; 2^{P-1}-1]$, par une translation de (-2^{P-1}).

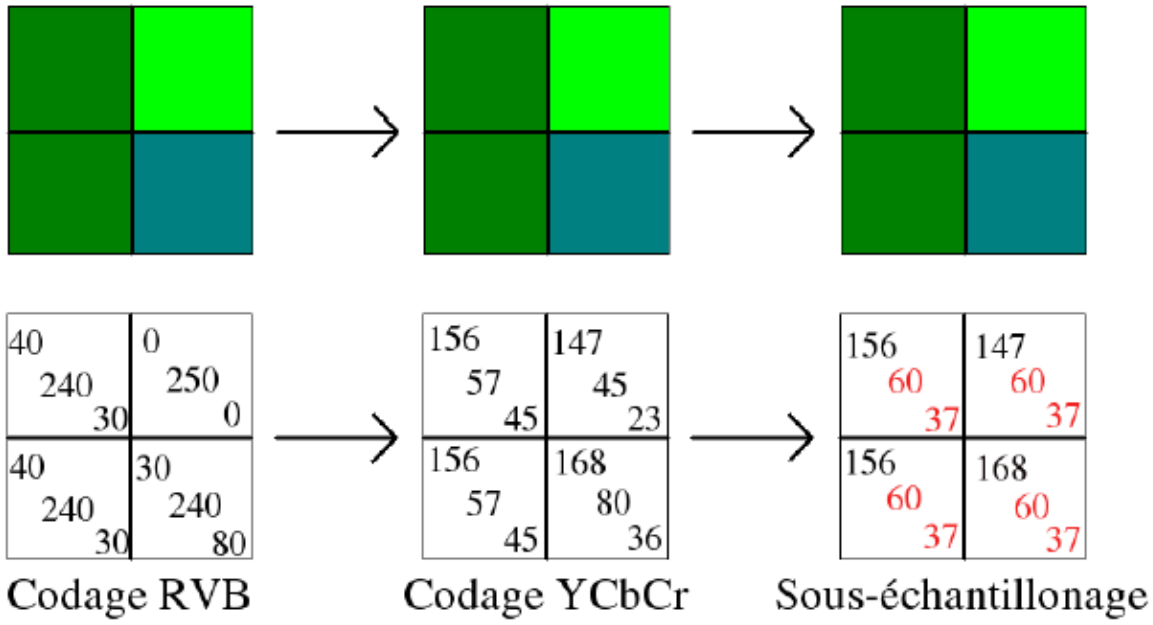


Figure 2.8 : Etape de sous-échantillonnage.

2.3.2 L'étape 2 : Transformation DCT

L'œil est sensible aux basses fréquences et peu sensible aux hautes fréquences. Le JPEG tire avantage de cette sensibilité, selon [50] on passe tout d'abord dans le domaine fréquentiel et on applique une transformée de Fourrier discrète. Des approximations sont effectuées sur les hautes fréquences ; elles ne sont donc pas aperçues par l'œil humain. Pour ce faire, chaque composante Y, Cb et Cr est découpée en blocs de 8 × 8 valeurs indexée de 0 à 7 de haut en bas et de gauche à droite comme illustré par la figure 2.9.

Chacun de ces blocs appartenant au domaine spatial est ensuite transformé en un bloc de 8×8 valeurs dans le domaine fréquentiel par la transformation DCT. Un coefficient DCT S_{uv} de coordonnées (u, v) dans le domaine fréquentiel s'exprime en fonction des 64 valeurs du bloc dans le domaine spatial s_{xy} de coordonnées (x, y) à partir de la formule

$$S_{uv} = \frac{1}{4} C(u) C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} s_{xy} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \quad (2.1)$$

Avec

$$C(0)=\frac{1}{\sqrt{2}} \text{ et } C(u)=1 \text{ si } u\neq 0$$

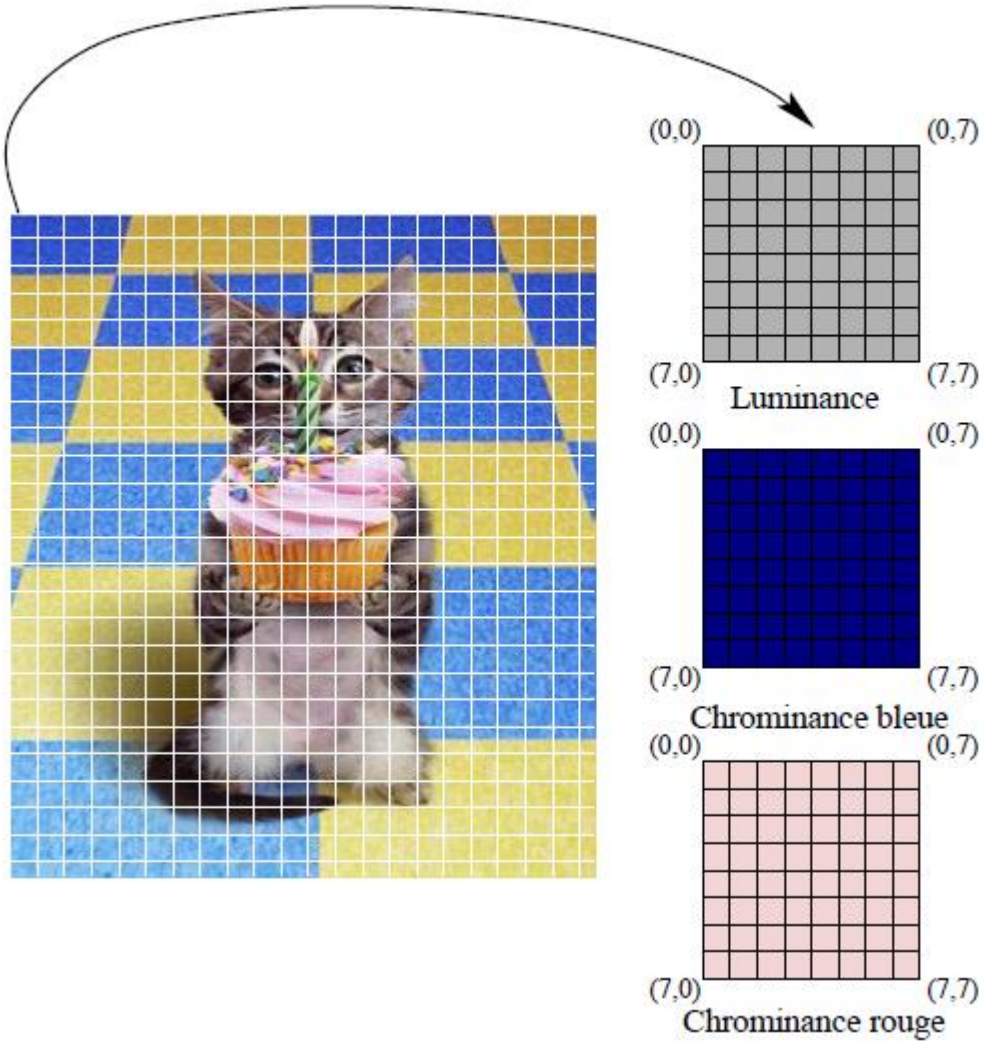


Figure 2.9 : Découpage en blocs de 8 × 8 valeurs.

Cette transformation est en fait la partie réelle de la transformée de Fourrier discrète. La transformée en cosinus discrète inverse, (IDCT) permet de retrouver, lors de la décompression, les blocs dans le domaine spatial à partir des coefficients DCT, à l'aide de la formule

$$s_{xy} = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)S_{uv}\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \quad (2.2)$$

Parmi les coefficients DCT, on distingue le coefficient $S_{0,0}$, aussi appelé coefficient DC, des autres que l'on nomme coefficients AC. Le coefficient DC est le coefficient des basses fréquences, et qui porte la majorité de l'information. Ce coefficient étant généralement le plus grand ; on ne stocke que sa différence avec le coefficient DC du bloc précédent. Le tableau 2.1 illustre un bloc de coefficients DCT.

235,6	-1	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0,6	1,3
0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1	1
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

Tab 2.1 : Exemple d'un bloc de coefficients DCT de la luminance [21].

Le coin en haut à gauche des blocs DCT contient les valeurs de basses fréquences, tandis que les hautes fréquences se situent dans le coin bas à droite. Dans un bloc de coefficients DCT, les droites d'équation u + v = cte regroupent les coefficients pour une fréquence donnée.

2.3.3 L'étape 3 : Quantification

La quantification est une étape importante de la compression JPEG. C'est lors de cette étape que l'information est la plus dégradée. Chaque bloc de coefficients DCT est divisé par une table de quantification. Cette division s'effectue coefficient à coefficient et le résultat est arrondi par défaut. Les tables de quantification sont construites à partir d'un facteur de qualité Q et de tables de référence représentées dans le tableau 2.2.

Les tables de quantification tab quant sont calculées à partir des tables de référence *tab-ref* (une pour la luminance et une pour les chrominances) à partir de la formule

$$tab\_quant(i,j) = \left\lceil \left(tab\_ref(i,j) \times c(Q) + 50 \right) / 100 \right\rceil \ \forall i, j = 0...7 \quad (2.3)$$

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	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Tab.2.2 : Tables de référence pour la luminance (à gauche) et la chrominance (à droite) [21].

Où *c(Q)* se déduit à partir du facteur d+3e qualité de la manière suivante.

$$C(Q) = \begin{cases} \dfrac{5000}{Q} & \text{Si } Q < 50, \\ 200 - 2Q & \text{Sinon} \end{cases} \quad (2.4)$$

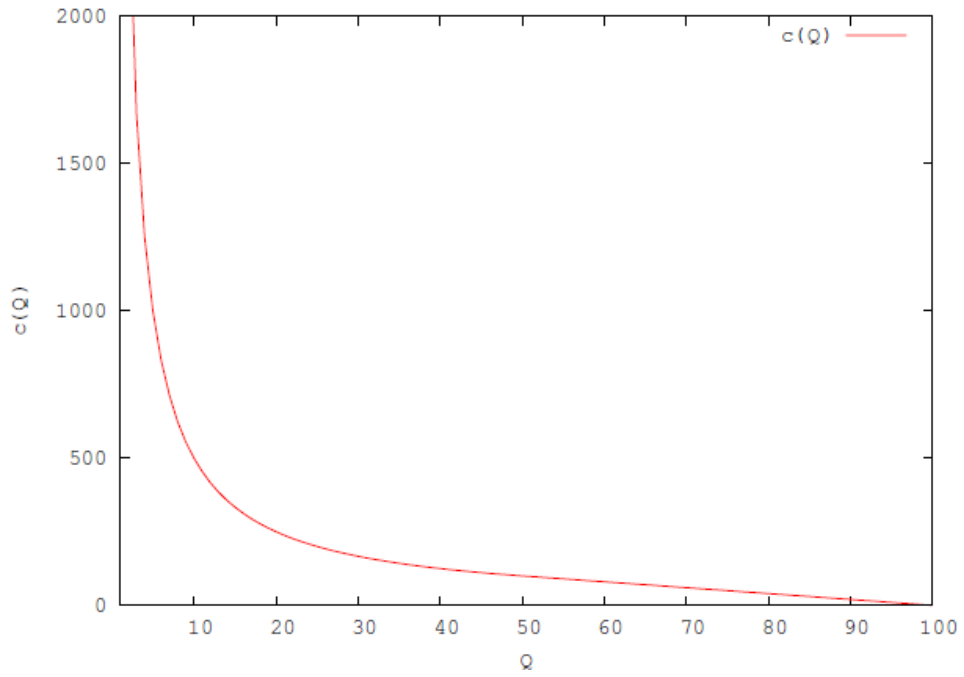


Figure 2.10 : *c(Q)* en fonction du facteur de qualité.

Le facteur de qualité prend des valeurs comprises entre 1 et 100, la valeur 1 correspondant à une faible qualité, c'est-`a-dire de dégradation maximale et la valeur 100 a la qualité la plus forte, c'est-`a-dire sans dégradation. En effet, si Q = 100, alors les coefficients des tables de quantification sont tous égaux a 1 d'après l'équation (2.4) ; les coefficients DCT sont juste arrondis lors de l'étape de quantification.

La quantification a pour effet de favoriser les basses fréquences, c'est-`a-dire celles qui contiennent le plus d'information. En effet, les coefficients les plus faibles se trouvent en haut à gauche des tables et les coefficients les plus grands en bas à droite. De la même manière, les coefficients DCT de la luminance sont favorisés par rapport à ceux des composantes de chrominance.

En conclusion, la quantification introduit majoritairement des coefficients DCT quantifiés a 0 dans les hautes fréquences et les composantes de chrominance. Le tablea.2.3 nous donne un aperçu de la proportion de coefficients DCT quantifiés à 0.

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Tab. 2.3 – Exemple d'un bloc de coefficients DCT quantifiés pour la luminance [21].

2.3.4 L'étape 4 : Codage RLE

L'étape de quantification introduit beaucoup de zéros dans les hautes fréquences. Un codage particulier, le Run Length Encoding (RLE) permet de compresser sans perte, des données qui comportent majoritairement des longues plages de symboles identiques.

Pour préparer au codage RLE et regrouper au maximum les 0, le format JPEG prévoit le regroupement des coefficients AC quantifiés suivant une séquence dite Zig-Zag. Cette

séquence de parcours des blocs DCT quantifiés est représentée sur la figure 2.11. Elle consiste à regrouper tout d'abord les coefficients AC quantifiés d'une même fréquence puis de les ordonner selon les fréquences croissantes. Dans l'exemple du tableau 2.4, la séquence

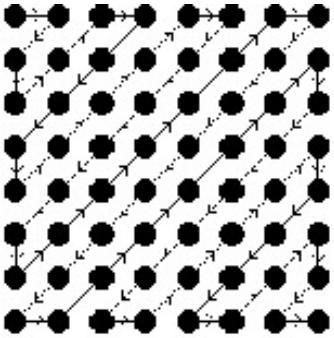


Figure 2.11– Séquence Zig-Zag.

Le codage RLE, représente chaque coefficient AC quantifié par un triplet (RL,S,V), où :
1. RL de 0 à 16, codé sur 4 bits, compte le nombre de 0 précédant le coefficient à coder,
2. S de 1 à 10, codé sur 4 bits, est le nombre de bits nécessaires pour coder le coefficient,
3. V est la valeur du coefficient codée sur S bits.
La valeur V est codée en prenant le complément à 2 binaire et en ne gardant que les S bits de poids faible comme l'illustre la table 2.4. Enfin, à la sortie du codage RLE, une suite binaire de longueur variable est obtenue comme représentée dans le tableau 2.5.

Valeur	Conversion	Complément à 2	Bits conservés
-7	(-7-1)=-8	1111000	000
-6	(-6-1)=-7	1111001	001
-5	(-5-1)=-6	1111010	010
-4	(-4-1)=-5	1111011	011
4		0000100	100
5		0000101	101
6		0000110	110
7		0000111	111

Tab. 2.4 – Exemple de valeurs codées sur 3 bits [21].

Valeurs	14	0	4	-8
Codage décimal	0 :4 :14	-	1 :3 :4	0 :4 :7
Codage binaire	0000 :0100 :1110	-	0001 :0011 :100	0000 :0100 :0111

Tab. 2.5 – Exemple de codage de 4 coefficients DCT quantifiés [109].

Selon [99] l'étude des distributions des triplets (RL,S,V) montre que les valeurs V semblent distribuées aléatoirement, tandis que certaines paires RL et S apparaissent plus fréquemment que d'autres. Le format JPEG prévoit alors un codage entropique pour compresser sans perte les paires RL et S. Suivant les modes du format JPEG, ce codage entropique peut être un codage de Huffman ou un codage arithmétique. Le codage de Huffman étant majoritairement utilisé, nous ne détaillerons que celui-ci. Le lecteur intéressé par le codage arithmétique pourra se référer à [101].

Les coefficients DC sont les termes de plus basse fréquence ; généralement non nuls, ils sont codés différemment des coefficients AC. De grande amplitude, seule la différence entre deux coefficients DC successifs est codée. Cette différence est ensuite codée par une paire (S,V) avec S de 0 à 11, codé sur 4 bits représentant le nombre de bits pour coder V et V la valeur de la différence codée sur S bits. Seule la valeur S sera ensuite codée par l'algorithme de Huffman [44].

2.3.5 L'étape 5 : Codage de Huffman

Le principe de ce codage est la création d'un arbre dont les feuilles sont les valeurs à coder. Un poids est associé à chacune de ces feuilles, et il correspond à la fréquence d'apparition de leurs valeurs ,et appelés aussi *codage entropique.*
De plus, celui-ci est dit préfixé, c'est-à-dire que chacun des mots du code ne peut être le début d'un autre mot du même code. Dans ce paragraphe, nous décrivons l'algorithme de construction du code de Huffman dans la figure c.14 illustré par un exemple adapté de [101]. Pour plus de détails, le lecteur pourra se référer à [101].

La figure 2.12 montre le déroulement de l'algorithme de Huffman pour la séquence « abracadabra » sur l'alphabet A = { a, b, c, d, r }. Le code de Huffman pour la séquence *S* est représenté dans le tableau 2.6.

Symbole		code
a		1
b		01
r		000
c		0010
d		0011

Tab. 2.6 – Code de Huffman pour S=« abracadabra ».

Le décodage de l'algorithme se fait en lisant les lettres du code jusqu'à trouver un mot de code présent dans le dictionnaire. Le code étant préfixé, aucune ambigüité sur le mot de code n'est possible et le décodage est univoque. Le dictionnaire étant construit en fonction des données à coder, il doit être fourni au décodeur.

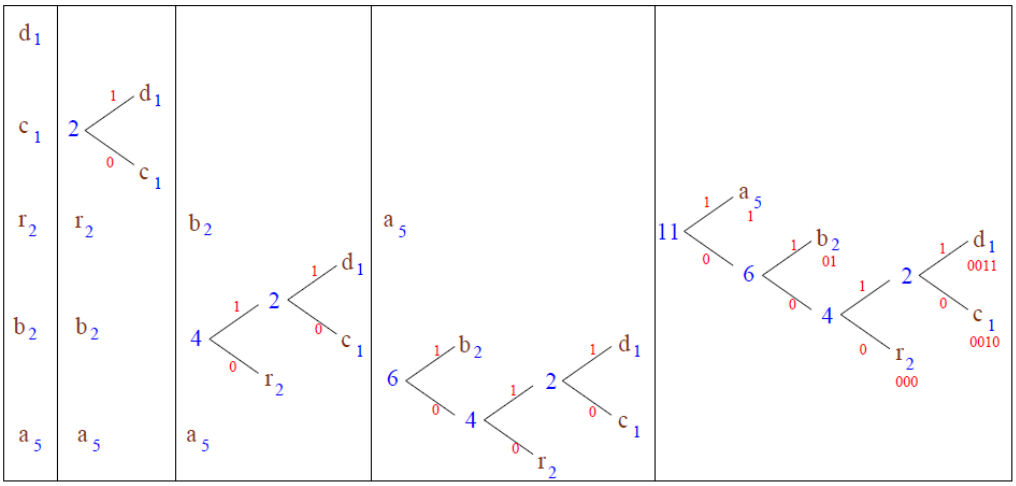


Figure 2.12– Algorithme de Huffman pour *S*=« abracadabra ».

La dernière étape du format JPEG est un codage entropique de Huffman de la valeur *S* de la différence des coefficients DC et des paires (RL,S) pour les coefficients AC. Deux cas sont alors possibles. Soit les dictionnaires sont transmis dans l'entête du fichier JPEG pour le décodage, soit ce sont les numéros de dictionnaires prédéfinis par la norme. Ces dictionnaires

prédéfinis ont été déterminés expérimentalement par le groupe d'experts JPEG et sont en moyenne très performants. La plupart des applications manipulant le format JPEG utilisent les dictionnaires prédéfinis.

Un exemple d'histogramme est donnée dans la figure 2.13 Notons qu'il existe plusieurs améliorations de ce test et que de nombreux travaux sont menés dans ce domaine mais ils font appels à des notions compliquées d'analyse statistique. Citons entre autre les travaux de Jessica Fridrich et de son équipe, qui ont développé une méthode stéganalytique sur les deux logiciels considérés comme les plus fiables en matière de stéganographie appliquée à des fichiers au format JPEG : OutGuess [152] et F5.

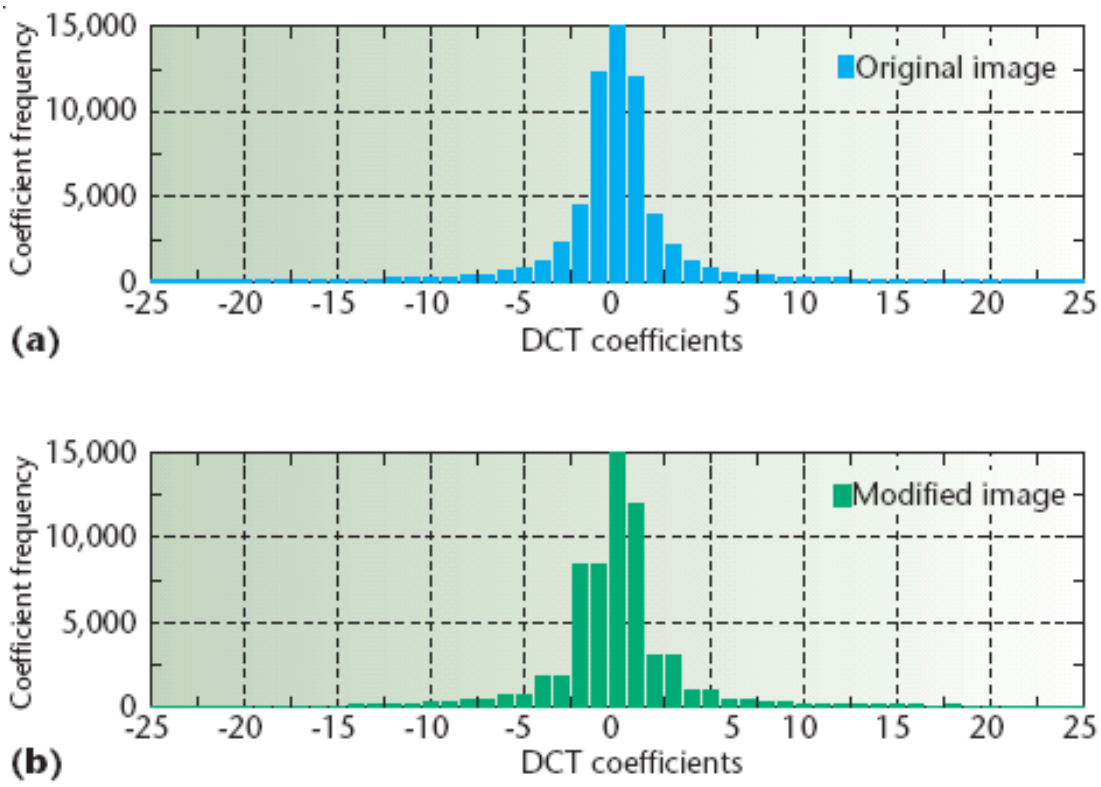


Figure 2.13 – Histogramme des fréquences.

2.4 Stéganographie adaptée au format JPEG

Les étapes présentées précédemment peuvent se regrouper en deux étapes. La première étape prend en entrée une image non compressée, appartenant au domaine spatial et transforme celle-ci en coefficients DCT quantifiés, appartenant au domaine fréquentiel.

Cette première transformation n'est pas bijective, elle correspond à une étape de compression avec perte d'information. La seconde étape prend en entrée des coefficients DCT du domaine fréquentiel et les transforme en une suite de données binaires dans le domaine fréquentiel compressé. Le format JPEG ne laisse donc pas le choix quant aux données qui peuvent recevoir l'information à dissimuler. En effet, la première étape étant avec perte, nous ne pouvons pas insérer de l'information au cours de celle-ci sans risque de ne pouvoir l'extraire. D'autre part, modifier ne serait-ce que quelques bits dans les données binaires du fichier JPEG impliquent nécessairement un décodage de Huffman suffisamment erroné pour que l'image stéganographiée ne ressemble en rien à une image naturelle. La seule possibilité est alors de dissimuler le message à l'aide des coefficients DCT.

Lorsque les premiers algorithmes de stéganographie tels Jsteg ou Outguess [70] dans sa version première, ont été spécifiés, une attaque proposée par A. Westfeld et A. Pfitzmann [10] a mis en évidence des déviations statistiques triviales sur l'histogramme des coefficients DCT quantifiés. Par exemple, comme l'illustre les figures 2.14 et 2.15, Jsteg a tendance à égaliser des paires de coefficients DCT quantifiés. A partir d'un simple test du $\chi 2$, ils ont alors mis au point des détecteurs stéganographiques permettant de discriminer les stégo média des supports de couverture.

Des rustines ou de nouveaux algorithmes ont alors été proposés, tenant compte des nouvelles contraintes imposées par cette attaque. Nous présentons dans ce paragraphe, trois algorithmes de stéganographie dédiés au format JPEG conçus pour préserver les statistiques du premier ordre des coefficients DCT quantifiés. Ces algorithmes nous servent au chapitre 3 à illustrer les méthodes de stéganalyse.

Figure 2.14– Image chinois.jpg [99].

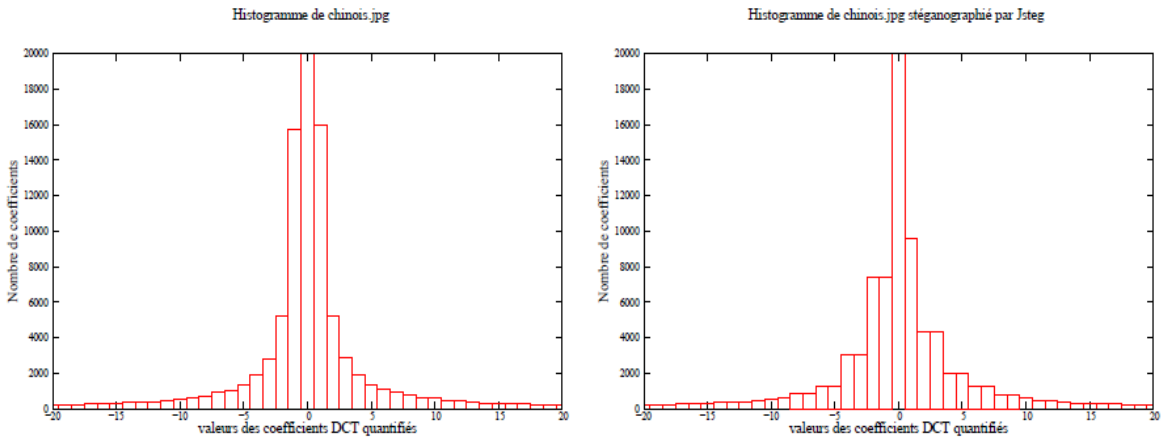


Figure 2.15– Modifications de l'histogramme de chinois.jpg par Jsteg [99].

2.5. Logiciels Stéganographiques

2.5.1 Outguess

Nom :Outguess0.2

Licence :Open source

Classification : DCT embedding

Format d'image : JPEG ,PNM

Plateforme :Unix, Windows

URL :http ://www.outguess.org

Outguess [133] est l'un des tout premiers algorithmes de stéganographie dédié aux images et issu de la communauté scientifique. Sa version première, (0.13), de 1998. Outguess dissimule l'information dans les LSB des coefficients DCT quantifiés du format JPEG. Il utilise RC4 pour chiffrer le message et sélectionner aléatoirement les coefficients DCT et propose d'utiliser un codage de parité.

Dans un premier temps, le message M de longueur l à dissimuler est chiffré avec l'algorithme de chiffrement flot, RC4 [45], en un message M'. Outguess propose en option l'utilisation

d'un code de parité aprés le chiffrement. Dans un deuxième temps, l'algorithme simule l'insertion de M′ avec plusieurs Vecteurs d'Initialisation (IV), IV_i. L'IV qui minimise le nombre de changements dans les LSB des coefficients DCT est alors retenu.

Outguess insère tout d'abord un premier registre de 32 bits constitué de la concaténation de l'IV et de la longueur du message, chacun codé sur 16 bits. Le GPA est ensuite réinitialisé avec l'IV. L'insertion s'effectue en forçant les LSB des coefficients DCT choisis a la valeur des bits a insérer. L'ordre des coefficients est fixé par la suite des indexes des coefficients DCT (pos_i) définie par

$$\begin{aligned} pos_0 &= 0, \\ pos_i &= pos_{i-1} + R\big(x(i)\big) \end{aligned} \qquad (2.5)$$

Où R(x(i)) est une valeur tirée aléatoirement dans [1, x(i)] et

$$x(i) = \begin{cases} x(i-1) & \text{si } i \mod 8 \neq 0, \\ 2 \times \dfrac{\text{nombre de coefficients DCT non utilisés}}{\text{nombre de bits restant à insérer}} & \text{sinon.} \end{cases} \qquad (2.6)$$

A la fin de l'étape d'insertion du message chiffré, Outguess corrige dans une seconde étape les distorsions introduites sur l'histogramme des coefficients DCT. Pour ce faire, chaque modification est compensée par une modification inverse. Une modification par insertion LSB change un coefficient DCT 2i en 2i + 1 et inversement. Si chaque inversion est corrigée alors l'histogramme est globalement inchangé. L'objectif des corrections apportées par Outguess est de maintenir identiques les écarts $f_{2i+1} - f_{2i}$ entre deux valeurs adjacentes 2i et 2i + 1 de l'histogramme, avant et après l'insertion tout en restant proche de f_{2i+1} et f_{2i}. Pour chaque valeur f_i de l'histogramme un nombre maximum de changements T_i est toléré au cours du processus de correction. Si ce seuil est dépassé alors l'algorithme essaie de compenser la modification courante. A la fin, Outguess essaie de corriger les modifications résiduelles. L'algorithme **corriger** (pos, val) consiste à trouver un coefficient DCT d'index inférieur à pos de valeur adjacente à val, non déjà utilisé pour l'insertion ou la correction et à inverser son LSB.

D'autre part, N. Provos associe à chaque coefficient DCT une valeur de détectabilité, D qui n'est pas définie dans [134] mais qui apparaît clairement lorsque l'on regarde de plus près les sources de Outguess [133]. D est donné par la relation

$$D(DCT) = \begin{cases} 1 & \text{Si } DCT \leq 16, \\ 0 & \text{Si } 16 < DCT < 240, \\ -1 & \text{Si } DCT \geq 240. \end{cases} \qquad (2.7)$$

La détectabilité de l'insertion est la somme des détectabilités des coefficients DCT modifiés par la dissimulation du message moins la somme des détectabilités des coefficients DCT modifiés par la correction statistique. La distorsion introduite par Ougtuess est alors définie comme la somme du nombre de coefficients modifiés et de la détectabilité de l'insertion. Une autre optimisation qui apparaît aussi dans les sources consiste à utiliser prioritairement les coefficients les moins détectables pour la correction statistique. Le processus de correction est résumé dans la figure C.19. Les algorithmes d'insertion et d'extraction sont résumés aux figures C.20 et C.21. Le lecteur doit néanmoins avoir à l'esprit que l'IV est un paramètre public et peut donc être choisi comme étant celui qui entraîne la distorsion la plus faible. D'autre part, les algorithmes Enc(.) et Dec(.) correspondent aux applications de codage et de décodage par un code correcteur d'erreurs. Ces applications sont soit l'identité (absence de codage correcteur) soit un code parité.

La figure 2.16 représente l'évolution de l'histogramme de la figure 2.17 tandis que la figure 2.18 illustre la différence des histogrammes avant et après l'insertion par Outguess. D'aprés la figure 2.19, les pixels modifiés se répartissent sur l'ensemble de l'image et préférentiellement autour des zones non homogènes, sur les contours.

2.5.2 F5

Nom :F5
Licence :Open source
Classification : DCT embedding
Format d'image : JPEG
Plateforme :Dos, Windows
URL :http ://wwwrn.inf.tu-dresden.de/ westfeld/f5.html

F5 fut créé par des chercheurs du monde académique au code source libre ,F5 est un algorithme de stéganographie +/-1 sur les coefficients DCT quantifiés, a été présenté par A.Westfeld [203, 202] en 1999. F5 conjugue différentes techniques pour conserver l'histogramme des coefficients DCT quantifiés proche de celui d'une image naturelle, d'une part et minimiser le nombre de changements à effectuer sur les coefficients DCT, d'autre part.

Afin de préserver l'histogramme des coefficients DCT quantifiés, l'information est insérée en fixant le LSB du coefficient DCT porteur à la valeur du bit du message à insérer. Si le LSB du coefficient et le bit du message sont identiques, le coefficient est laissé inchangé. Dans le cas contraire, la valeur absolue du coefficient est décrémentée de 1. De plus, les histogrammes des images naturelles présentent plus de coefficients non nuls impairs que pairs. Pour conserver cette propriété, les coefficients DCT positifs et impairs ainsi que les coefficients négatifs et pairs codent un bit de message de valeur 1, les autres codant un bit de message à 0, comme l'illustre la figure 2.20. De même, la symétrie devant être conservée, les coefficients DCT égaux à 0 sont donc ignorés. D'autre part, les coefficients 0 étant ignorés, le récepteur ne peut faire la différence entre un 0 de l'image et un 0 produit par le changement d'un -1 ou 1. Si un tel changement survient lors de l'insertion, le bit du message est réinséré de nouveau dans le prochain coefficient DCT. On parle alors d'effondrement.

D'autre part, pour minimiser les changements introduits, une randomisation et un codage par syndrome sont mis en œuvre. En utilisant un GPA, les coefficients DCT sont mélangés avant l'insertion. Cela a pour effet d'uniformiser la répartition des changements sur l'ensemble des coefficients DCT. Sans cette étape de randomisation, les changements sont localement concentrés sur les premiers coefficients. L'emploi du codage par syndrome en stéganographie a initialement été proposé par R. Crandall [63] en 1999. Depuis , l'utilisation de code

Figure 2.16 : Image tahiti.jpg.

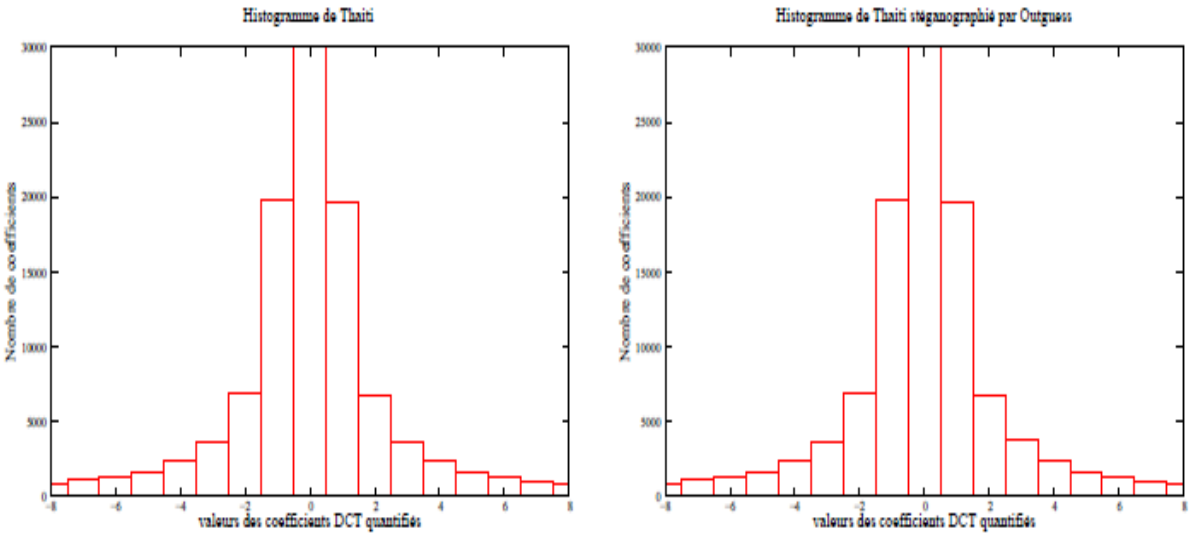


Figure 2.17 : Modifications de l'hitogramme de tahiti.jpg par Outguess.

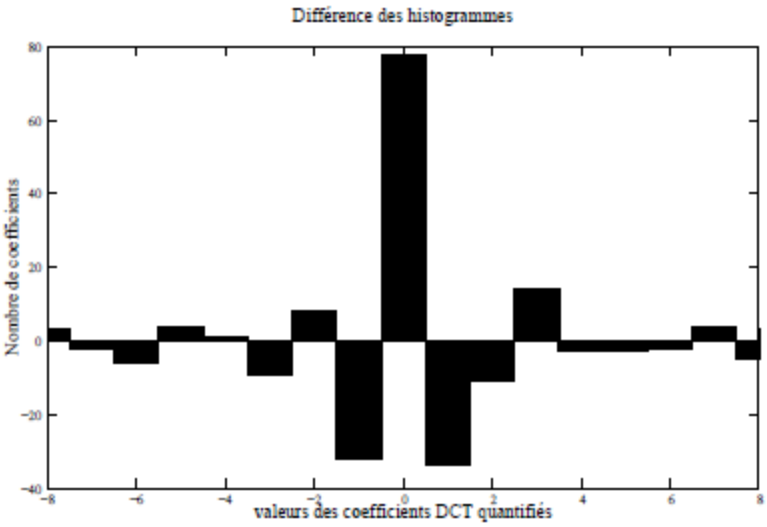


Figure 2.18 : Différence des histogrammes avant et après insertion.

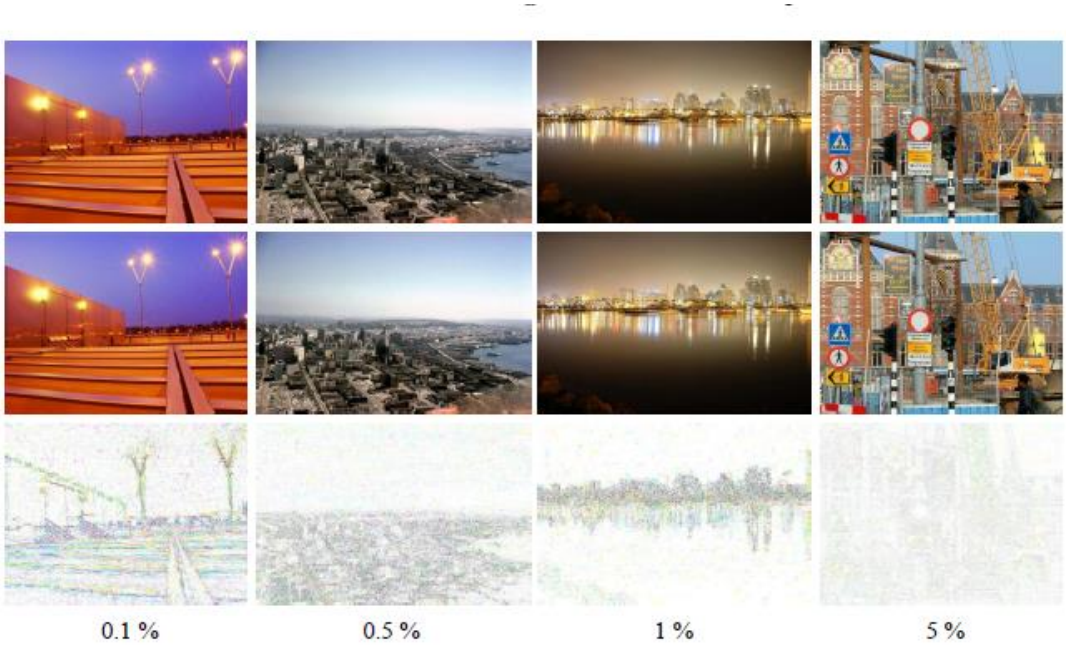


Figure 2.19– Images stéganographiées par Outguess pour différents taux stéganographiques.

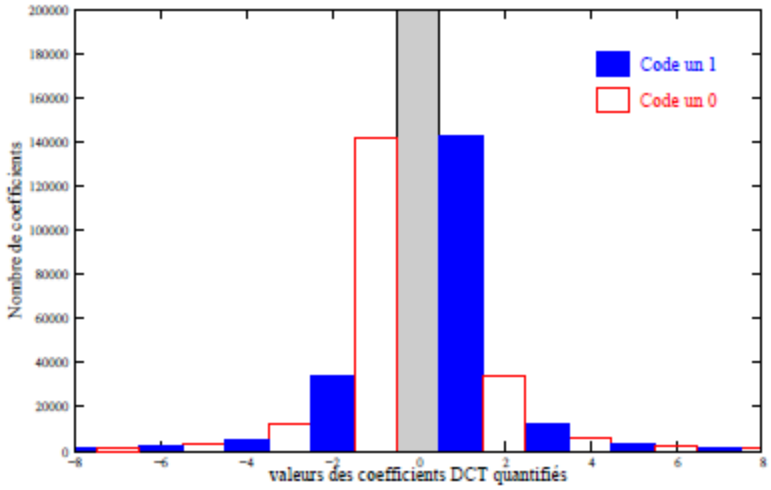


Figure 2.20– Codage des bits du message par les coefficients DCT.

parfaits et de codes MDS (*Maximum Distance Separable*) a été largement étudiée. Pour plus de détails le lecteur pourra se référer à [90, 94, 95, 93, 54, 42, 130].

Tout le problème est donc de retrouver e à partir de son syndrome. En règle générale, mettre en évidence une telle application est un problème difficile. Le lecteur intéressé par la théorie des codes pour plus de précisions concernant les choix de codes et l'existence ou non d'algorithmes de d´décodage pour des codes donnés.

Le principe du codage de syndrome est d'associer a un message à dissimuler *m*, de longueur $r = (n - k)$ symboles de F_q, en utilisant un vecteur *v*de *n* symboles du support de couverture. Pour ce faire, *v*est transformé par l'algorithme d'insertion en *v'* vecteur de *n*symboles, tel que $S(v') = m$. En d'autres termes, on cherche v' dont le syndrome est exactement le message que l'on souhaite dissimuler. D'autre part, nous voulons que le nombre de modifications pour passer de *v*a v' soit minimum , $(v' - v)$ de poids faible.
Les algorithmes d'insertion et d'extraction, εmb et εxt, du schéma de stéganographie vérifient alors

$$\varepsilon mb(v,m) = v + \Delta(m,v) = v' \text{ avec } w(\Delta(v,m)) \le \rho, \qquad (2.8)$$

$$\varepsilon xt(v') = S(v') = m$$

Où ρ est le nombre maximum de modifications autorisées pour *n* symboles du support. Notons *D* l'application qui, à un syndrome *s*, associe un vecteur de poids borné de syndrome

$$D : F_q^{(n-k)} \rightarrow F_q^n \qquad (2.9)$$

$y \rightarrow D(y)$

telle que *S(D(y)) = y* et *w (D(y)) ≤ ρ*. De plus, nous appelons *rayon de couverture d'un code C*, noté *ρ(d)*, l'entier défini par

$$\rho(C) = \min\left\{\rho\{S(y)\, w(y) \leq \rho\} = F_q^{(n-k)}\right\} \qquad (2.10)$$

Tout comme D, le rayon de couverture est en général difficile à évaluer. Afin de minimiser les distorsions introduites lors de l'insertion, nous aurons plutôt tendance, lorsque les contraintes le permettent, à utiliser des codes pour lesquels le rayon de couverture est minimum. De tels codes sont appelés codes parfaits. C. Fontaine et F. Galand [35] préconisent préférentiellement des codes MDS lorsque certains symboles doivent impérativement demeurer inchangés au cours du processus d'insertion et qu'il est impossible de les déterminer à l' avance. Les codes MDS sont des codes pour lesquels la distance minimale est maximale. Pour répondre à nos contraintes, nous prenons

$$\Delta(v,m) = D\left(m - S(v)\right) \text{avec } d_H\left(v, D\left(m - S(v)\right)\right) \leq \rho \qquad (2.11)$$

Où d_H est la distance de Hamming. Le problème du codage de syndrome est alors équivalent au problème de décodage du syndrome de poids borné, qui est en général un problème difficile.

L'algorithme F5 utilise des codes parfaits, les codes de Hamming. Ces codes sont paramétrés par *s*, leur longueur vaut $n = 2^s - 1$ et leur dimension $k = n - s$. De plus, l'algorithme de décodage est connu ainsi que son rayon de couverture qui vaut 1. Le taux stéganographique est inférieur à $s/(2^s - 1)$ et pour dissimuler un message de *s* bits, au plus un seul bit parmi n est modifié. En fonction de la taille du message et du support, F5 choisit le paramètre s adapté. Le code est défini par sa fonction syndrome,

$S : F_2^n$

$$y \rightarrow S(y) = \bigoplus_{i=1}^{n} y_i . i \qquad (2.12)$$

où *i* est sous sa forme binaire, codé sur *s* bits. Soit à insérer le message *m* de *s* bits dans le vecteur support *v* de *n* bits, nous ´évaluons alors

$$m - S(v) = m \oplus S(v) \qquad (2.13)$$

et $j = D(m-S(v))$ est défini comme étant l'entier correspondant à $m-S(v)$. Si $j = 0$ alors $m = f(v)$ et aucune modification n'est nécessaire, dans le cas contraire le bit $v_j$doit être inversé. Pour ce faire, la valeur absolue du coefficient *DCT* de $LSBv_j$ est décrémentée. Si un effondrement apparaît, $v_j$est remplacé par v_j+1, v_j+1 par v_j+2, etc ... et le *LSB* du prochain coefficient *DCT* non nul est inséré en v_n. Les algorithmes d'insertion et d'extraction sont résumés dans les figures C.27 et C.28.

La figure 2.30 représente l'évolution de l'histogramme de la figure 2.29 tandis que la figure 2.31 illustre la différence des histogrammes avant et après l'insertion par F5. D'après la figure 2.32 les pixels modifiés se répartissent sur l'ensemble de l'image et préférentiellement autour des zones non homogène sur les contours.

2.5.3 JPHide and JPSeek

Nom :JPHIDE - JPSEEK
Licence :Freeware
Classification : DCT embedding
Format d'image : JPEG
Plateforme :Unix, Windows
URL :http ://linux01.gwdg.de/ alatham/stego.html

JPHide and JPSeek [3] est un algorithme de stéganographie implémenté par A. Latham en 1999 selon deux versions, 0.3 et 0.5. La version 0.5 intègre en plus un algorithme de compression du message à dissimuler.

Le message à dissimuler est tout d'abord chiffré avec l'algorithme Blowfish [116]. La clé cryptographique est un mot de passe tronqué à 120 caractères. Dans un premier temps, les 8 premiers coefficients DCT sont réduits modulo 256 et concaténés en un bloc de 64 bits. Le mécanisme de dérivation de clé est initialisé avec le mot de passe et le premier bloc est chiffré. Dans un deuxième temps, la taille du message à insérer est codée sur les 24 bits

premiers bits du bloc chiffré ; les 40 restants constituant un vecteur d'initialisation (IV). Ce bloc de 64 bits est ensuite concaténé au message chiffré avant l'insertion.

Figure 2.21 – Image anim.jpg.

Steghide dissimule le message dans les bits de poids faible des coefficients *DCT* mais il s'autorise aussi à modifier ponctuellement certains bits du plan *1*. D'autre part, les coefficients -1, 0 et 1 bénéficient d'un traitement particulier. Enfin, l'ordre de parcours des coefficients *DCT* est fixé par une table statique et n'est pas randomisé. En revanche, avec une certaine probabilité dépendant de la longueur du message à dissimuler et de la taille de l'image, le coefficient *DCT* courant est ignoré. Ces coefficients sont choisis à l'aide du *GPA*[135,7].

La figure 2.24 représente l'évolution de l'histogramme de la figure 2.23 tandis que la figure 2.27 illustre la différence des histogrammes avant et après l'insertion par StegHide. D'après la figure 2.22, les pixels modifiés se répartissent sur l'ensemble de l'image et préférentiellement autour des zones non homogènes, sur les contours.

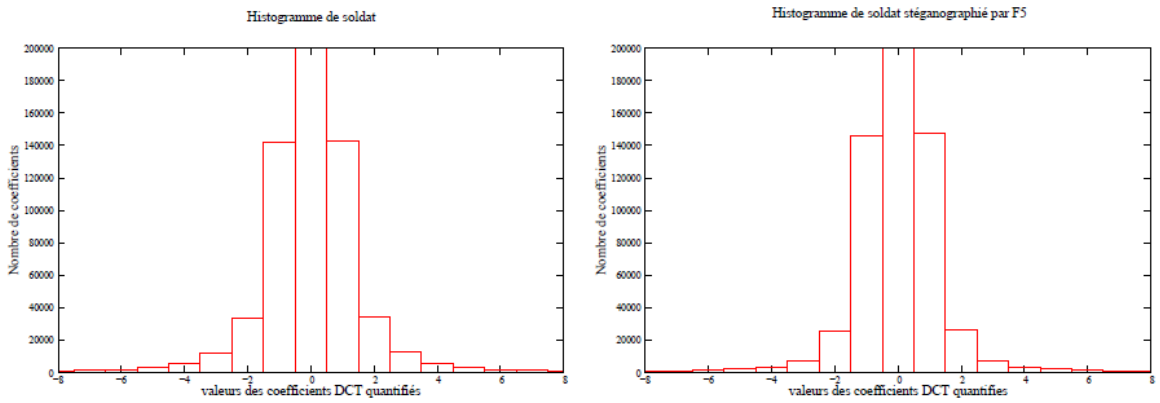


Figure 2.22 :l'histogramme avant l'insertion

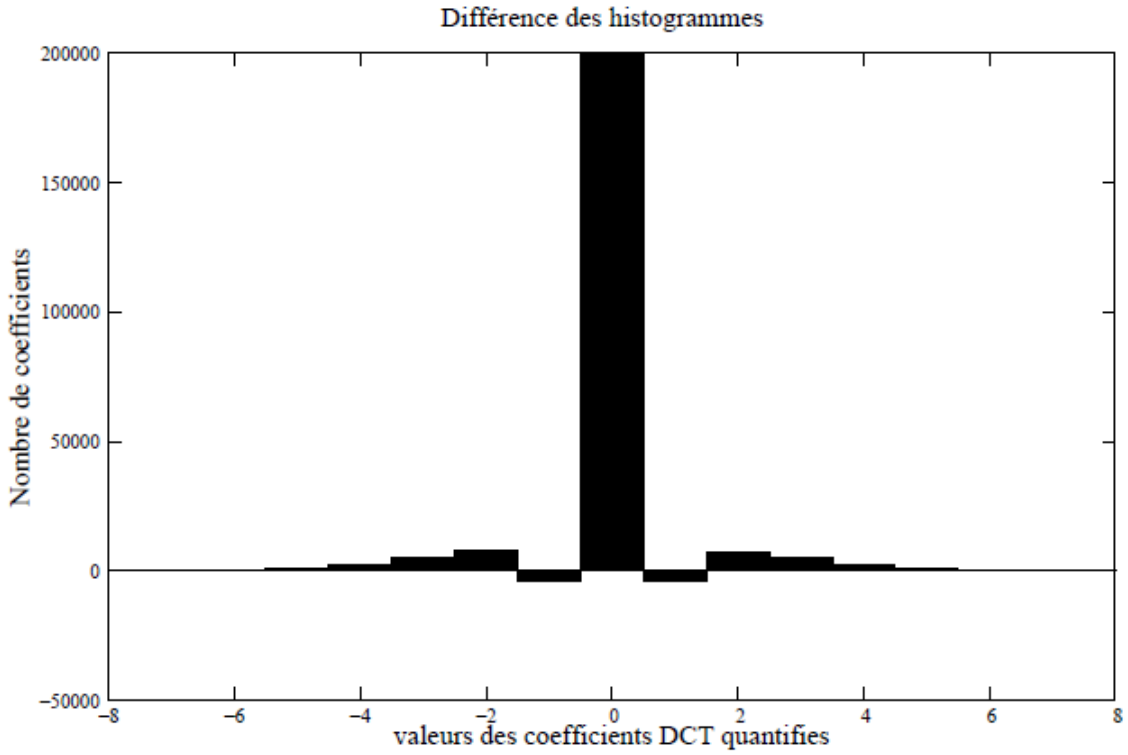


Figure 2.23 : Différence de l'histogramme de l'image anim.jpg après l'insertion.

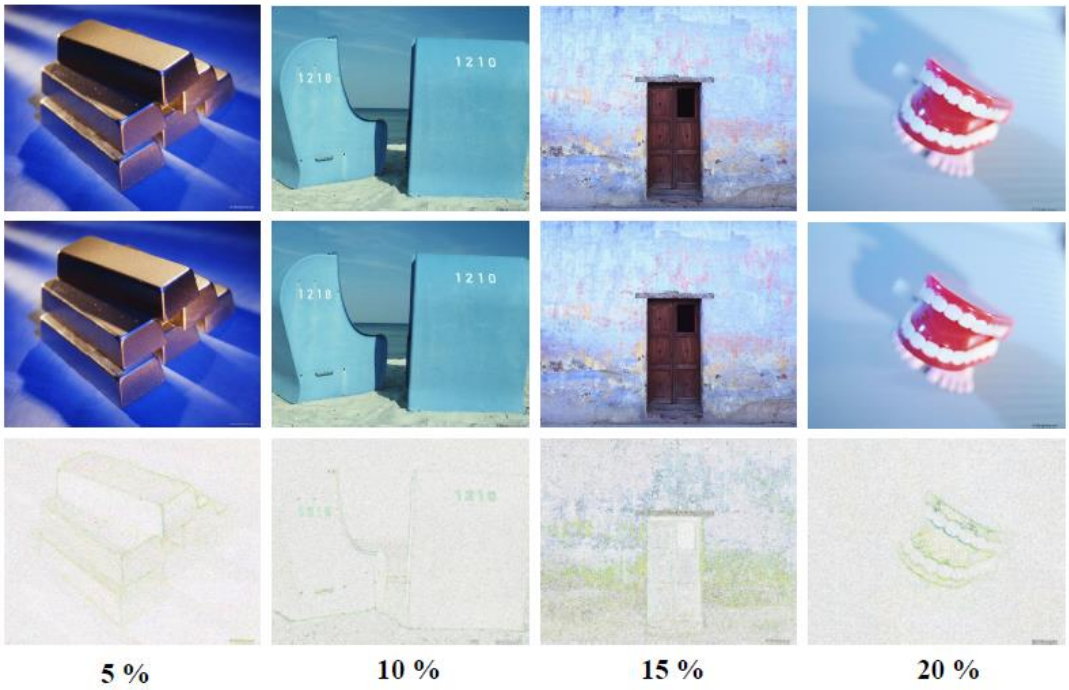


Figure 2.24 : Images stéganographiées par F5 pour différents taux [155].

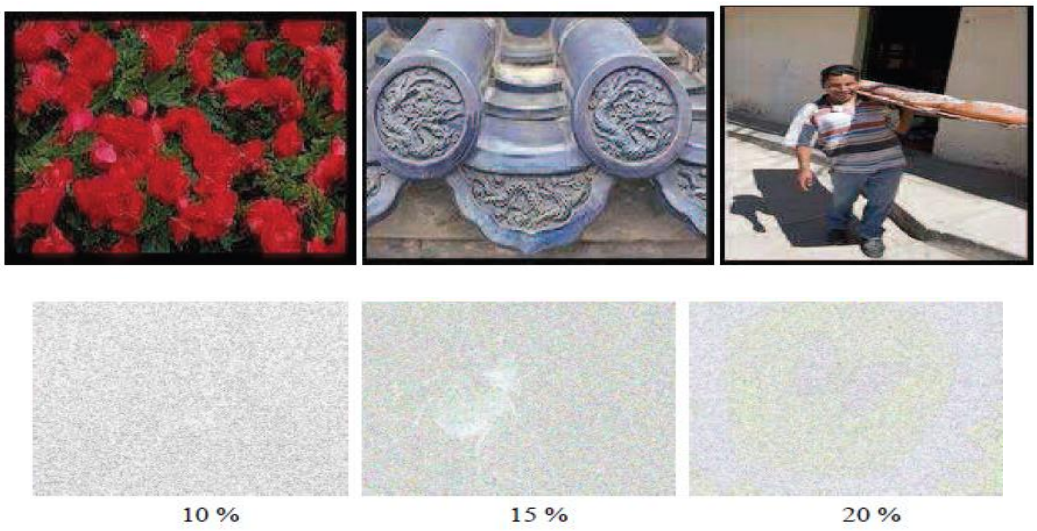


Figure 2.25 – Images stéganographiées par StegHide pour différents taux.

Figure 2.26– Image pic21.jpg.

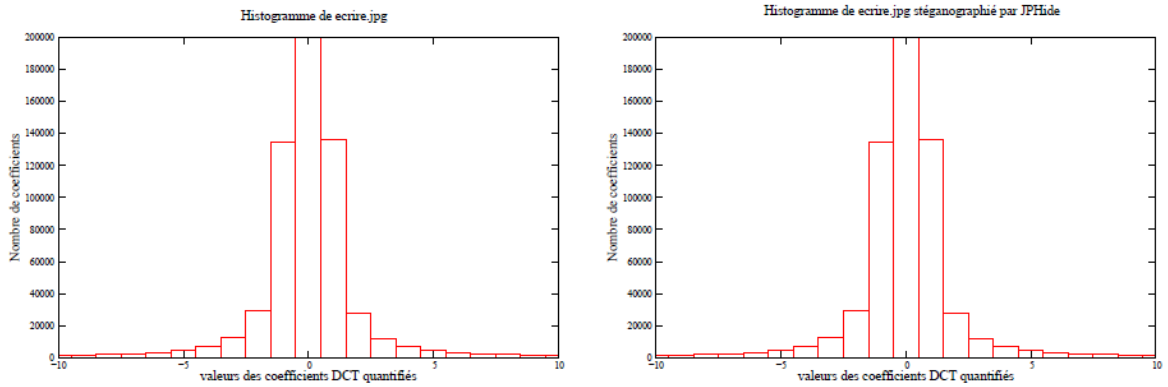


Figure 2.27– Modifications de l'histogramme de l'image pic21.jpg par StegHide.

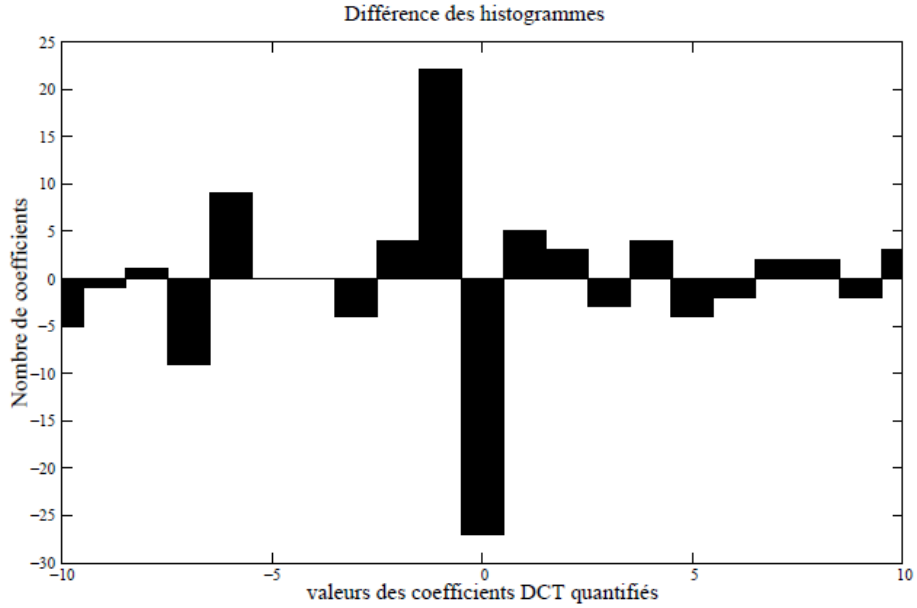


Figure 2.28– Différence de l'histogramme de l'image pic21.jpg après et avant insertion.

Indépendamment de l'intérêt scientifique, l'étude de techniques de stéganalyse, c'est- à -dire des techniques visant à détecter la présence d'information cachée, a un impact certain dans le domaine de recherche de preuves informatiques, dans la lutte contre la pornographie infantile [29, 16, 115].

2.6 Conclusion

Dans ce chapitre nous avons présenté, dans une première partie, quelques concepts de base des images numériques, pour lesquels les techniques stéganographiques s'appuient. On distingue deux 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.

Chapitre 3

Le modèle d'attaquant pour la stéganalyse

3.1 Introduction

L'objectif principal de la stéganographie est de dissimuler un message secret dans un médium de couverture 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.

C'est le cas par exemple 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 Attaque d'un schéma de stéganographie

Le but de cryptographie est de récupérer le message, ayant été préalablement crypté, sans connaissance de la clé, mais la stéganalyse (ou analyse stéganographique) n'a pas comme objectif initial d'extraire les données dissimulées à l'aide d'un algorithme stéganographique. Elle consiste à la détection de la présence des données cachées.

Dans le cas de la stéganographie par modification d'un médium dit empirique[3] (tel que les images numériques naturelles), la stéganalyse revient en pratique à vérifier la statistique du support intercepté, pour déterminer si elle est ou non altérée par un algorithme particulier.
De manière plus formelle, pour un support donnée $x=(x_1, ...,x_n)$, le problème de détection de message secret peut-être représenté comme un test entre deux d'hypothèses :

$$\begin{cases} H_0 : x \sim P_c \\ H_1 : x \sim P_s \end{cases}$$

H_0 : Le support x ne contient pas de message caché (cover)
H_1 : Le support x contient un message caché (stego) (3.1)

Le stéganalyste, représenté par la gardienne Eve dans *le problème des prisonniers* doit donc décider entre ces deux hypothèses pour juger si oui ou non le médium est stéganographié.
Il existe trois types de stéganalyse se différenciant par les objectifs recherchés et les moyens utilisés :

- *la stéganalyse à gardien passif* : se contente uniquement de décider si oui ou non le support intercepté est porteur de message caché. En d'autres termes, le rôle du stéganalyste (la gardienne Eve) se limite uniquement à un test d'hypothèses H_0, H_1 (eq 3.1).
- *la stéganalyse à gardien actif* : peut également faire le test d'hypothèse (eq 3.1), mais en plus elle a pour but d'empêcher la communication de données secrètes. Pour ce faire, le stéganalyste va essayer d'apporter quelques modifications sur le médium intercepté (compression, filtrage. . . etc) dans le but de détruire le message caché s'il existe ;
- *la stéganalyse à gardien malicieux* : va plus loin que la stéganalyse à gardien passif ou actif. L'objectif du stéganalyste, pour ce type d'analyse, est de comprendre la technique stéganographique utilisée, et même extraire le message secret. Une fois extrait, le stéganalyste peut contourner le message secret pour ses propres fins. Il peut même réintroduire un autre message falsifié.

3 : un médium de couverture est dit empirique si son modèle statistique est partiellement ou totalement inconnu [57].

Pour ce type de stéganalyse, a proposé un protocole de stéganographie à clé publique luttant contre la modification et la classification du message secret.

L'attaquant doit savoir si le document contient des données cachées. Il existe plusieurs formes d'attaques selon les moyens dont dispose l'attaquant [100]:

- **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. Les Méthodes de la Stéganalyse

Plusieurs méthodes peuvent être utilisées pour la détection des images stéganographiées. La plus simple profite du 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 utilise 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 [151].

3.3.1 Stéganalyse Universelle

Dans la stéganalyse universelle, la communication du message secret est également réalisée à travers une seule image. Le stéganographe considère que Eve la gardienne ne connaît ni l'algorithme de stéganographie utilisé, ni la quantité de bits insérés, ni la clé stéganographique. Eve connaît seulement la distribution des images sources.

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 ?* ».

a) 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), [132].

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 [96].

Figure 3.3 : Image Lena originale.

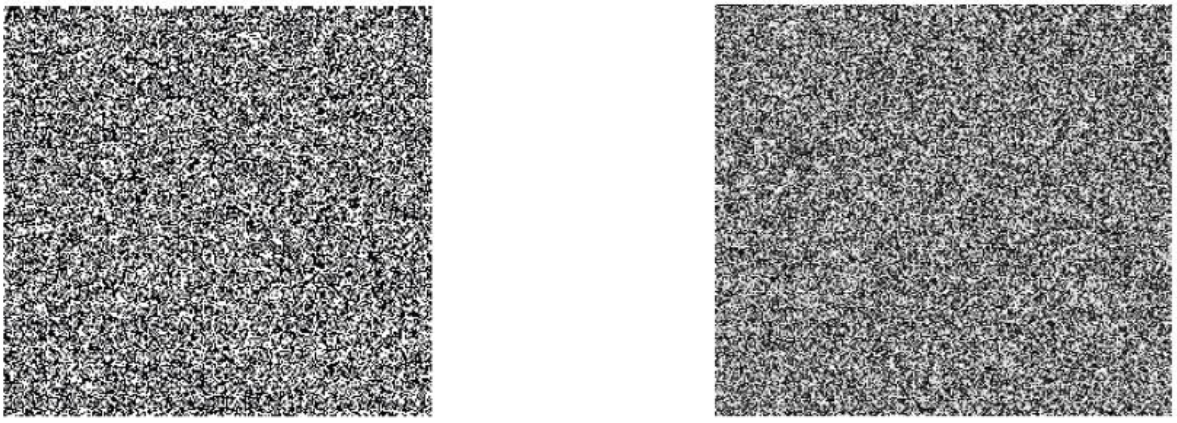


Figure 3.4.: Dernier plan de bit avant et après insertion de l'image Lena.

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

b). Stéganalyse basé 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. L'analyse statistique à base X^2 est présentée par [10], le schéma de Memon basée sur les paires de pixels [155] et le schéma proposée par Fridrich basée sur les groupes de pixels [89]. Pour les autres schémas [108], [160], c'est le même principe que celui définit dans [10], mais la différence intervient dans le choix des sous-ensembles.

c). Méthode de Stéganalyse dans le domaine spatial

Kobsi *et al.* dans [157] 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 ([70]). Une extension de cette analyse est donnée dans [181] qui proposent d'étudier les variations de certaines corrélations entre variables statistiques existant entre les différents plans de bits (7ème et 8ème 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 ([162], [101]) 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.

Fridrich présente dans [72] 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).

3.3.2 Stéganalyse Spécifique

La stéganalyse ciblée, également appelée *spécifique*, a pour principe d'essayer de déterminer les faiblesses de sécurité d'un algorithme particulier, en étudiant son "*implémentation*" et/ou ses "*failles statistiques*", pour pouvoir identifier la présence d'un message caché, par cet algorithme, dans un médium donné. Pour ce faire, le stéganalyste, qui a connaissance au préalable de l'algorithme de dissimulation (il cible un algorithme stéganographique particulier), génère un ensemble de supports stéganographiés avec le même algorithme pour 1) comprendre et analyser les différentes étapes de l'algorithme, et 2) comparer la statistique des images de couverture qu'il a à sa disposition avec celles qui ont été générées. À travers cette opération, le stéganalyste tente de déterminer les points caractérisant ainsi que les faiblesses de l'algorithme ciblé, pour pouvoir discriminer les images *stego* des images *cover*. On peut donc dire que la stéganalyse ciblée se base sur l'identification des caractéristiques spécifiques, qui distinguent un algorithme stéganographique donné des autres algorithmes.

Si les mesures dépendent des algorithmes que nous essayons de détecter, la stéganalyse est dite *spécifique*. Par exemple, J. Barbier *et al.* dans [Bar 26] proposent une stéganalyse dédiée aux algorithmes Outguess [70], F5 [61] et JPHide and JPSeek [161] 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 ([101], [94], [124]) proposent une méthode de stéganalyse spécifique aux algorithmes Outgeuss et F5. Ces attaques reposent sur la modification d'une certaine donnée statistique macroscopique de l'image dans le processus d'insertion.

a). L'analyse par calibration

La méthode de stéganalyse par calibration est une attaque ciblée, qui a été initialement conçue pour faire face à l'algorithme F5 de [172] utilisant le principe de stéganographie par correspondance (LSB Matching (±1)) sur les images JPEG .Le principe de cette méthode d'analyse, proposé par [101], consiste à estimer l'histogramme des coefficients DCT de l'image de couverture (l'image originale sans aucune modification) à partir de l'image stéganographiée. Pour ce faire, les auteurs proposent de procéder à une décompression de

l'image JPEG dans le domaine spatial, suivie d'un décalage en bloc de 4 pixels en colonne et en ligne, puis une recompression de l'image, ce qui permet d'avoir une estimation de l'histogramme DCT très proche de l'originale. Une fois obtenue, l'histogramme DCT estimé est comparé à celui de l'image stéganographiée interceptée, ceci en utilisant la méthode des moindres carrés. Cette méthode de stéganalyse permet non seulement de détecter la présence du message caché, mais aussi d'estimer sa taille. Par la suite, dans la littérature, cette méthode a été revisitée et adaptée pour d'autres attaques dédiées aux images JPEG [94,3].

b). L'analyse RS

L'analyse RS est une méthode de stéganalyse ciblée dédiée aussi à l'algorithme de stéganographie par substitution des LSB dans le domaine spatial. Le principe de cette méthode, proposée par [130], repose sur la classification des pixels, selon leur variation, en groupes distincts.

L'analyse RS s'appuie sur l'ajout de bruit, de ce fait elle est n'est pas adaptée pour des images fortement bruitées. Par ailleurs, cette méthode de stéganalyse est plus efficace sur des images modifiées de manière aléatoire. Si les modifications stéganographiques sont adaptées aux média de couverture (insertion dans des régions ciblées moins détectables), l'analyse RS est moins efficace. Pour résoudre ce genre de problème, il est fréquent d'utiliser la méthode RS en complément de la méthode X^2. Cette combinaison de méthodes est efficace pour une large palette d'algorithmes stéganographique procédant par substitution des LSB.

c). L'analyse du X^2

La méthode de stéganalyse X^2, développée par [10], est une attaque ciblée générale, qui peut être employée pour la détection de tout algorithme d'insertion utilisant le principe de stéganographie par substitution. Cette attaque, pionnière dans le domaine, repose sur le test statistique du X^2 pour la détection du message secret.

Le principe de cette méthode s'appuie sur le fait que pour une dissimulation par substitution des LSB, les caractéristiques statistiques de l'image originale ont tendance à être altérées considérablement. En partant du principe que les bits du message à dissimuler sont uniformément distribués, les auteurs mettent en évidence que les fréquences d'apparition des paires de valeurs (PoV) d'une image modifiées sont quasi identiques, contrairement à une image naturelle. Autrement dit, la sur écriture des LSB réduit l'écart de fréquence entre des nuances de gris adjacentes au sens LSB.

La méthode X^2 a été initialement utilisée pour la détection des méthodes stéganographiques LSB opérant dans le domaine spatial (spécifiquement au départ pour l'algorithme EzStego sur les images GIF [10]). Plus tard dans la littérature, [128] ont adapté cette méthode aux algorithmes stéganographiques par substitution opérant dans le domaine transformé sur des images JPEG (tel que l'algorithme Jsteg [169]). Pour appliquer l'analyse X^2 sur les images JPEG, les auteurs utilisent le même concept décrit ci-dessus, mais cette fois ci sur les coefficients DCT.

L'analyse X^2 n'est efficace que quand il s'agit de détection de données cachées de manières séquentielle, ou lorsque le chemin d'insertion est connu d'avance. Dans le cas d'utilisation d'une séquence pseudo-aléatoire pour l'insertion du message, la méthode X^2 ne donne pas de bons résultats. Pour résoudre le problème de l'insertion aléatoire, [128] proposent également, dans leur publication, d'adapter cette méthode pour la détection de contenu caché manière aléatoire. Pour cela, ils appliquent le test de X^2 sur une fenêtre de taille plus petite que l'image, qui se déplace au fur et à mesure.

3.4 Les principaux scénarios de la stéganalyse

En stéganalyse, il existe plusieurs scénarios possibles. Ces scénarios définissent un certain nombre de règles et d'hypothèses, sur ce que le stéganalyste connaît du processus de dissimulation établi par le stéganographe. Dans cette section nous présentons quelques-uns des différents scénarios.

3.4.1 Stéganalyse à clairvoyance

Kerckhoffs [2], stipulant que la sécurité d'un schéma ne doit pas tenir dans le fonctionnement du système mais dans la clé uniquement. Dans ce scénario, le stéganographe (Alice et Bob) considère que la gardienne (Eve) dispose de tous les éléments du schéma stéganographique, à l'exception de la clé secrète utilisée lors de l'insertion. Autrement dit, on considère que la distribution des images sources, P_C, est connue, c-à-d que Eve dispose de suffisamment d'images sources sans message, pour en déduire une distribution similaire à celle utilisée par le stéganographe. Nous supposons également que Eve connaît l'algorithme de stéganographie utilisé, ainsi que le *payload* (la quantité de bits α) inséré. Autrement dit, Eve connaît également la distribution des images stéganographiées P_S. Enfin, Eve ne connaît pas la clé stéganographique. Par ailleurs, la communication du message secret est réalisée à travers une seule image.

Du côté de la gardienne Eve, le problème de stéganalyse, dans ce scénario, se ramène donc à un simple test consistant à vérifier si la distribution de l'image interceptée est celle d'une image de couverture ($x \sim P_C$), ou celle d'une image stéganographiée ($x \sim P_S$) (revoir les hypothèses H0 et H1 dans l'Eq. 3.1).

Du côté du stéganographe, la stéganalyse à clairvoyance est le scénario plus difficile. En effet, pour le stéganalyste, il est plus facile d'élaborer une attaque efficace dans ces conditions. Ces dernières années, beaucoup de travaux ont été consacrés à ce scénario, que ce soit en stéganographie [107] [125] ou en stéganalyse [167] [168] [169].
Les travaux présentés dans ce manuscrit considèrent également ce premier scénario de stéganalyse à clairvoyance.

3.4.2 Stéganalyse à payload inconnu

Le scénario de stéganalyse à payload inconnu est très similaire à celui de la stéganalyse à clairvoyance, à l'exception du *payload* qui est inconnu. Dans ce scénario, La communication du message secret est effectuée à travers une seule image. Le stéganographe considère que le stéganalyste (Eve la gardienne), connaît la distribution des images sources, l'algorithme de stéganographie utilisé, mais ne connaît ni le *payload* $\propto$ inséré, ni la clé stéganographique utilisée pour l'insertion. De façon plus formelle, le problème de stéganalyse dans ce scénario, revient donc à vérifier si la quantité de bits insérés est supérieure ou non au seuil de détection $\propto_0$, pour juger de la présence ou non d'un message caché :

$$\begin{cases} H_0 : \alpha \simeq 0 \\ H_1 : \alpha \geq \alpha_0 \end{cases} \tag{3.2}$$

Où $\propto_0$ est un seuil de détection (également appelé taux d'insertion critique) au-delà duquel l'image interceptée est considérée comme porteuse d'informations cachées. Pour ce scénario, il existe deux approches de résolution possibles : 1) *la stéganalyse quantitative* dont le principe repose sur l'estimation de la taille du message dissimulé [60] [125] ou bien 2) *l'approche CFAR* (pour *Constant False-Alarm Rate*) dont le principe est de fixer lors de l'apprentissage le taux de faux positif au minimum, pour pouvoir ensuite lors des tests détecter les images stéganographiées avec un *payload* inconnu [105].

3.4.3 Stéganalyse avec cover-source mismatch

En stéganalyse avec *cover-source mismatch*, Eve, la gardienne ne connaît que partiellement, ou pas du tout, l'origine et la distribution des images de couverture sources.
En effet, en pratique, les images utilisées lors de l'apprentissage par le steganalyseur ne viennent pas de la même source que les images utilisées lors des tests. Dans un tel cas de figure, il est important de déterminer ce qui caractérise une distribution *cover*, et de se libérer de la dépendance forte à la distribution *cover* lors de l'apprentissage. Dans cet esprit, [167] évoquent la possibilité de réaliser l'apprentissage sur une base contaminée, mais cela ne semble pas donner de bons résultats. [66], dans le cadre de la stéganalyse à clairvoyance avec *cover-source mismatch* et connaissance d'une base test, proposent d'ajouter à la base d'apprentissage une base de test filtrée pour être moins sensible au *cover-source mismatch*. [174] attaquent le problème de manière détournée en utilisant une approche de classification par ensemble de classifieurs Perceptron. Pour capturer la variété de types des images et pour modéliser la distribution des images de couverture, l'apprentissage de leur classifieur est effectué sur une très grande base de l'ordre de millions d'images. [102] proposent une autre approche, meilleure que celle de Lubenko et Ker ; pour lutter contre le *cover-source mismatch*. Pour cela, ils ont adapté l'ensemble de classifieurs FLD de [106]. Leur approche ne nécessite pas l'emploi d'une très grande base d'images.

3.4.4 Stéganalyse par mise en commun

La stéganalyse par mise en commun, ou *Pooled Steganalysis* en anglais [24], est un scénario difficile et réaliste, qui ajoute en plus une dimension temporelle au problème initial. Dans ce scénario, on s'approche de l'idée d'un stéganalyseur de trafic automatique qui analyse ce qui passe sur le réseau de communication. Deux cas de figures sont possibles pour ce scénario. Le premier cas consiste à voir, sur le trafic de communication, un seul acteur 2 malintentionné qui peut parfois envoyer des données numériques contenant des informations secrètes cachées. Le deuxième cas consiste à avoir plusieurs acteurs qui communiquent sur le réseau et échangent des données numériques différentes. Certains de ces acteurs, dont l'intention est malhonnête, peuvent parfois utiliser la stéganographie pour s'échanger des informations secrètes. Dans les deux cas de figures, le stéganographe a la possibilité d'envoyer plusieurs images stéganographiées avec des messages différents, ou diviser un seul message long sur plusieurs images stéganographiées. Le stéganalyste (la gardienne Eve) quant à lui ne possède

aucune information sur le schéma stéganographique utilisé par l'acteur (ou les acteurs), mais peut analyser la nature des différentes images circulant sur le trafic, sur un temps étendu. Le stéganalyste peut ainsi essayer d'apprendre à distinguer entre une distribution naturelle (*cover*) et une distribution suspecte (*stégo*). Dans le cadre de la stéganalyse par mise en commun à plusieurs acteurs, récemment [112] [111] ont proposé le premier classifieur pratique. Leur classifieur basé sur une approche par acteur, regroupe les observations d'images de chaque acteur en nuage de points, puis calcule ensuite la distance entre les différents acteurs. L'acteur dont le comportement diffère des autres, i.e. celui qui s'éloigne du groupe, est jugé comme étant suspect.

3.5 Un problème de discrimination

Nous prenons maintenant la place du stéganalyste et selon [151] intercepte en aveugle des média numériques échangés entre Alice et Bob. Avant de préciser l'attaquant, il faut définir formellement ce qu'est un schéma de stéganographie à clé secrète.

Définition 1 *Un système de stéganographie a clé secrète* $\sum$ *est la donnée d'un ensemble C de supports de couverture et de trois algorithmes polynomiaux, dont*

- *Un algorithme probabiliste* $\boldsymbol{K}$ *de génération des clés. Il prend en entrée un paramètre de sécurité* 1^k *et renvoie une clé secrète* K_s.
- *Un algorithme probabiliste* $\boldsymbol{\varepsilon}$*mb d'insertion. Il prend en entrée la clé* K_s*, un message clair* $m \in \boldsymbol{M} \subset \{0, 1\}$*, un support de couverture* $I \in C$ *et renvoie un stégo médium I' de* $\boldsymbol{S}$*, l'ensemble des stégo média.*
- *un algorithme déterministe* ε*xt d'extraction. Il prend en entrée la clé privée* K_s*, un médium I'et renvoie le message clair m si I'appartient à* $\boldsymbol{S}$ *ou* $\perp$*, un message d'erreur sinon.*

Remarque 1 :

Tout comme en cryptographie, cette définition s'adapte aisément à un schéma de stéganographie à clé publique. Néanmoins, tout ce qui suit reste vrai ou se transpose sans difficulté pour des schémas à clés publiques. Dans un contexte plus général, cette transposition n'est en générale pas aussi directe et s'effectue souvent au prix de quelques adaptations et beaucoup de précautions.

Dans ce contexte, nous jouons le rôle d'un attaquant passif qui ne modifie pas les informations échangées entre Alice et Bob. L'objectif est, dans un premier temps, de différencier, parmi les média échangés, les supports de couverture des stégo média. Plus simplement, étant donné un médium *I*, nous devons répondre à la question « *I* est-il un stégo médium? ». Ce problème est équivalent à distinguer un support de couverture $c \in \mathbf{C}$, connaissant P_C d'un stégo médium $s \in \mathbf{S}$ connaissant P_S. La sécurité d'un schéma de stéganographie peut être définie à l'aide de n'importe quelle distance définie sur l'ensemble des distributions de probabilité définies sur un même support, comme par exemple la distance de Kullbak-Liebler proposée par C. Cachin.

Définition 2 *Soit* $\sum$ *un schéma de stéganographie. Nous appelons attaque par discrimination contre* $\sum$*, la donnée d'une fonction*

$$V : \mathrm{I} = C \cup S \to v_1 \times ... \times v_n$$
$$I \to V(I) = \left(V_1(I), ..., V_n(I)\right) \qquad (3.3)$$

où V_i *est une variable aléatoire définie sur* I *a valeurs dans* V_i *et calculable en temps polynomial.*

Selon la définition 2, une attaque par discrimination contre un système de stéganographie est équivalente à la donnée de *n* mesures, coordonnées d'un vecteur statistique. Une attaque est efficace si elle permet de distinguer **C** de **S** ou de façon équivalente V (**C**) de V (**S**). L'objectif d'un attaquant passif est de mettre en évidence une attaque V par discrimination efficace contre $\sum$ afin de répondre à la question « *I* est-il un stégo médium? ».
Clairement une attaque est efficace si et seulement si elle vérifie au moins l'un des deux critères suivants. Il existe $i \in [1, n]$ tel que $P_{Vi}(C) \neq P_{Vi}(S)$ ou il existe $i \in [1, n]$ tel que

$$P_{V_i \mid \{V_k, k \neq i\}}(c) \neq P_{V_i \mid \{V_k, k \neq i\}}(s) \qquad (3.4)$$

Le premier critère implique que $I \longrightarrow V_i(I)$ est aussi une attaque par discrimination efficace, ce qui n'est pas le cas pour le second. Si $V_i(I)$ n'est pas une attaque efficace, cela signifie que nous devons aussi observer les autres coordonnées de *V* pour obtenir de l'information sur *I* à partir de V_i. Une attaque efficace permet alors d'obtenir des distributions que l'on peut distinguer, encore faut-il construire un distingueur qui exploite une différence de distributions marginales, P_{Vi} , ou conditionnelles, $P_{v_i \mid \{v_k, k \neq i\}}$, évaluées sur les ensembles **C** et **S**.

Définition 3 *Soit ∑ un schéma de stéganographie et V une attaque par discrimination contre ∑. Nous appelons distingueur compatible avec V, la donnée d'une fonction définie par*

$$\mathrm{DV} : V_1 \times \cdot \cdot \cdot \times V_n \longrightarrow \{0, 1\},$$

Calculable en temps polynomial. Par convention 0 est associé à non stégo et 1 à stégo.

Celle-ci nous permet alors d'énoncer la définition de la stéganalyse par discrimination, qui semble être implicitement la plus usitée dans la communauté.

Définition 4 *Soit ∑ un schéma de stéganographie. Nous appelons stéganalyse par discrimination contre ∑ tout couple (V,D_V), où V est une attaque contre ∑ et D_V un distingueur compatible avec V .*

La définition formelle de stéganalyse par discrimination reflète parfaitement le comportement de l'adversaire réel. En effet, un attaquant effectif procède en deux étapes distinctes. Il doit tout d'abord mettre en évidence un ensemble de mesures dont au moins l'une des distributions marginales ou conditionnelles diffère sur les média de couverture et sur les stégo média, comme par exemple les coefficients RS . Dans un second temps, il doit mettre au point un distingueur qui va décider si un médium est stégo ou non en fonction des mesures effectuées. Il existe différents procédés pour construire des distingueurs, notamment les machines a support de vecteurs, une analyse en composante principale ou bien une analyse discriminante de Fisher. Le choix des mesures relève de l'expérience du stéganalyste tandis que la conception d'un distingueur relève de techniques statistiques dédiées aux tests d'hypothèses. De même, il convient de définir la sécurité d'un schéma contre une stéganalyse par discrimination.

Définition 5 *Soient ∑ un schéma de stéganographie et (V,D_V) une stéganalyse par discrimination contre ∑. On dit que ∑ est ε-sûr contre (V,D_V) si et seulement si*

$$D\left(P_{(D_v \circ v)(S)}\right) \leq \varepsilon \qquad (3.5)$$

De plus si ε = 0 alors le système est qualifié de parfaitement sûr contre (V,D_V).

La proposition suivante relie la sécurité contre une stéganalyse par discrimination donnée et la sécurité inconditionnelle classique.

Proposition.1 *Soit ∑ un schéma de stéganographie. Alors*

∑ est ε-sˆur ⇒ ∑est ε-sûr contre (V,D_V), ∀ (V,D_V) stéganalyse par discrimination.

Preuve :

D'après [10, p. 46], pour toute *f* fonctions définie sur *I* alors

$$D\left(P_{f(c)},P_{f(s)}\right)\leq D\left(P_C,P_S\right) \qquad (3.6)$$

La sécurité inconditionnelle d'un algorithme de stéganographie est une donc une borne de sécurité sur l'ensemble des stéganalyses par discrimination. De plus, si un schéma de stéganographie ∑ est parfaitement sûr alors cela signifie que ∑ est parfaitement sûr contre toute stéganalyse par discrimination. Dans ce cas, cela implique qu'il n'existe pas de stéganalyse par discrimination efficace contre ∑.

3.6 Modèles d'attaquants

Dans ce paragraphe, nous présentons tout d'abord, le modèle de sécurité *d'indistingabilité* issu du monde de la sécurité des primitives cryptographiques. Ce modèle s'adapte aisément à l'évaluation des schémas de stéganographie et modélise de façon réaliste l'attaquant passif réel. Nous déclinons ensuite ce modèle en deux modèles d'attaquant stéganographique, *l'attaquant spécifique* et *l'attaquant universel* selon que l'algorithme attaqué est connu ou non. Du point de vue du concepteur, un schéma de stéganographie est évalué sous l'angle de sa sécurité. Pour montrer qu'un schéma est ε-sûr dans un modèle d'attaquant donné, celui-ci doit montrer que ∀ (V,D_V), stéganalyse par discrimination, le schéma est ε-sˆur contre (V,D_V). En revanche, en nous plaçant du côté de l'attaquant, nous cherchons à quantifier préférentiellement l'insécurité du schéma étudié. Dans ce cadre, il faut mettre en évidence une borne minimum sur l'insécurité, c'est-`a-dire, trouver une attaque efficace contre le schéma et un distingueur adapté.

3.6.1 Modèle d'indistingabilité

Le modèle présenté vise à prouver la sécurité d'un schéma de cryptographie sous l'angle de l'indistingabilité. Pour ce faire, il met en jeu des attaquants adaptatifs ou non. Pour un algorithme de chiffrement, l'objectif de cet attaquant est de pouvoir distinguer des messages chiffrés connaissant les messages clairs. Au paragraphe précédent, nous avons montré que la stéganalyse par discrimination se réduit à un problème de discrimination statistique, le modèle d'indistingabilité est parmi les modèles de sécurité cryptographique, celui qui est le plus à même de fournir les briques nécessaires à la modélisation de l'attaquant réel.

3.6.2 Attaquant spécifique

L'attaquant classique utilisé implicitement dans toutes les stéganalyses effectives de la littérature est un attaquant IND-PA contraint, car il ne choisit plus m. En effet, lors d'une phase d'apprentissage, l'attaquant génère lui-même sa clé secrète, ses messages, ses supports de couverture et essaie d'obtenir de l'information sur les distributions P_C et P_S. Il construit enfin un distingueur pour ces deux distributions. Dans la phase de challenge, il intercepte les média échangés entre Alice et Bob et doit deviner pour chaque médium intercepté s'il est stégo ou non stégo. Pour un attaquant réel, O_1 et O_2 ne renvoient rien. De plus, lors du challenge, puisqu'il intercepte en aveugle les média, il ne peut imposer le message caché ; celui-ci est donc choisi par la challenger. Nous définissons ainsi un attaquant classique qualifié de spécifique car il cible un schéma de stéganographie particulier, dans le respect des principes de Kerckhoffs.

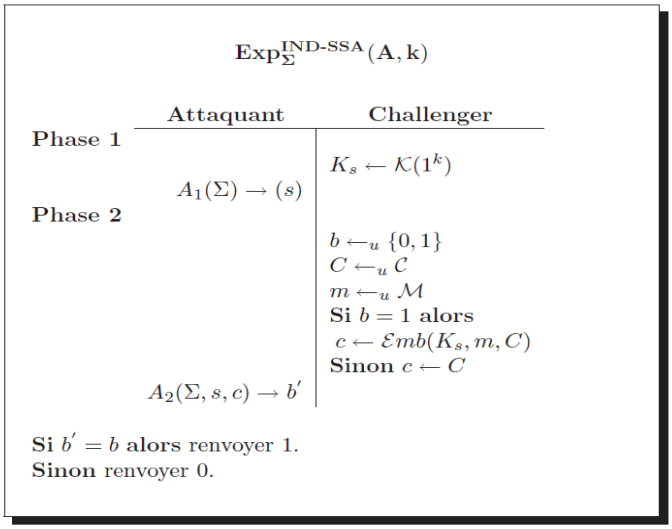


Figure 3.5 : Expérience en indistingabilité avec un attaquant A de type IND-SSA contre ∑ [21].

Cet adversaire est un attaquant IND-SSA (Specific Steganalysis Attack). Le jeu qui lui est associé est résumé dans la figure 3.5.

Dans ce jeu, l'adversaire n'est pas restreint à une stéganalyse précise, il peut utiliser n'importe laquelle d'entre elles. En pratique, nous cherchons à évaluer l'efficacité d'une stéganalyse par discrimination donnée et à relier celle-ci a l'insécurité du schéma attaqué.
Pour ce faire, nous introduisons l'attaquant IND-SSA(V,D_V) qui ne peut utiliser que la stéganalyse par discrimination (V,D_V) pour apprendre de l'information lors de la phase 1 et pour répondre au challenge. Cette attaquant est noté $(A^1_{(V.D_v)}, A^2_{(V.D_v)})$ Le jeu lui correspondant est résumé dans la figure 3.6 page suivante.

3.6.3 Attaquant universel

En stéganographie, on considère souvent un attaquant encore plus faible et ne respectant pas les principes de Kerckhoffs [150]. Bien évidemment, cet attaquant n'est jamais pris en compte lors de la phase de conception car beaucoup trop faible. Néanmoins, cet adversaire simule des attaques par discrimination qui ne dépendent pas de l'algorithme et qui mettent en évidence des faiblesses intrinsèques à une classe de schémas stéganographiques.
L'objectif est de mesurer le caractère générique d'une attaque. Pour ce faire, l'attaquant dispose de tous les schémas de stéganographie $\sum_i$, sauf un, $\sum_j$. Durant la phase d'apprentissage, il essaie d'obtenir de l'information sur les distributions P_C et $P_{Si\neq j}$, en espérant que P_{Si} ne soit pas trop différente d'au moins un P_{Sj} . Le challenge est alors réalisé avec $\sum_j$. L'attaque par discrimination possède ce caractère générique. Cela signifie en d'autres termes, qu'en entraînant un distingueur sur des schémas de stéganographie connus, nous sommes potentiellement capables de détecter des schémas inconnus. Cet attaquant IND-USA (Universal Steganalysis Attack) est qualifié d'universel [94].
Le jeu lui correspondant est résumé dans la figure 3.6. Dans ce jeu, l'adversaire n'est pas restreint à une stéganalyse précise, il peut utiliser n'importe laquelle d'entre elles. En pratique, nous cherchons à évaluer l'efficacité d'une stéganalyse par discrimination donnée et à relier celle-ci a l'insécurité du schéma attaqué. Pour ce faire, nous introduisons l'attaquant IND-USA(V,D_V) qui ne peut utiliser que la stéganalyse par discrimination (V, V_D) pour apprendre de l'information lors de la phase 1 et pour répondre au challenge. Cet attaquant est noté $(A^1_{(V.D_v)}, A^2_{(V.D_v)})$

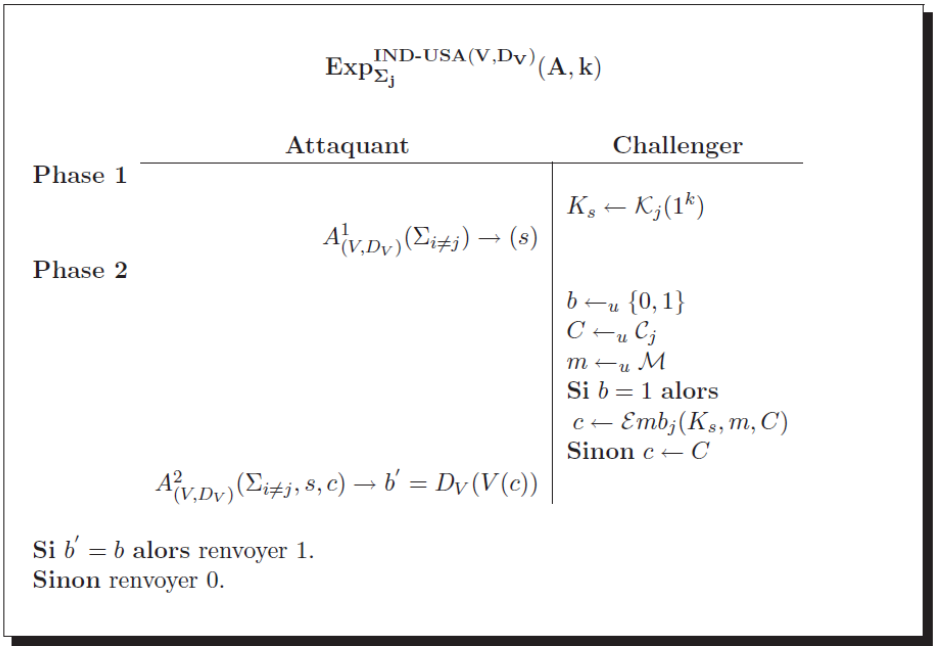


Figure 3.6 – Expérience en indistingabilité avec un attaquant A de type IND-USA(V,D_V) contre $\sum_j$ [28].

3.7 La stéganalyse sous d'autres angles

Récemment dans la littérature, on observe l'apparition de nouvelles approches en stéganalyses, qui ne peuvent être ni classées comme des attaques ciblées ni comme des attaques aveugles. Ces approches, qui abordent la stéganalyse sous un nouvel angle, peuvent être regroupées en deux grandes familles, l'une basée sur la théorie de la décision et l'autre sur la théorie des jeux :

3.7.1 La stéganalyse du point de vue de la théorie de la décision

Comme mentionné précédemment, l'objectif principal de la gardienne Eve, lors de la stéganalyse, est de pouvoir décider entre les deux hypothèses H_0 (support non stéganographié) et H_1 (support stéganographié). Autrement dit, décider si une image interceptée x={x_1, ...,x_n} est distribuée selon une loi P_C définissant l'hypothèse nulle H_0 ou bien selon une loi P_S définissant l'hypothèse alternativeH_1 (voir l'Eq. 3.1). Le problème principal est alors de formuler une définition statistique précise de l'hypothèseH_0 (l'image x interceptée ne contient pas un message dissimulé). Pour atteindre cet objectif, les méthodes dites "*de détection statistiques*" reposent sur la théorie de la décision, et utilisent pour la décision des outils purement statistiques. Pour choisir entre les deux hypothèses (H_0 et H_1), ces méthodes de

stéganalyse établissent d'abord un modèle paramétrique définissant la nature et la statistique des images de couverture, ensuite, se fixent un critère d'optimalité donné pour choisir le test statistique le plus adéquat au problème de la détection. La décision finale du détecteur statistique est obtenue par le biais d'une fonction qui associe à l'image analysée x la classe à laquelle elle appartient (−1 pour une image de couverture et +1 pour une image stéganographiée).

À la fin du processus de décision, la qualité du test statistique établi est définie par le nombre d'erreurs commises. Pour cela, les méthodes de détection statistiques utilisent les probabilités d'erreur. Lorsque l'image analysée est déclarée comme étant stéganographiée par le détecteur δ, alors que ce n'est pas le cas, on parle de fausse-alarme. La probabilité associée à cet événement est alors la probabilité de faux positif (notée P_{FP}). Dans le cas contraire, si une image stéganographiée n'est pas détectée, on parle de non-détection et la probabilité associée à cet événement est la probabilité de faux négatif (notée P_{FN}). Enfin, pour ce genre d'approches statistiques la puissance du test statistique δ est défini par la probabilité de détection notée β(δ).

Rappelons au passage que la construction de tout test statistique nécessite d'abord d'établir un modèle paramétrique définissant la nature des images de couverture non modifiées, puis de choisir un critère d'optimalité adéquat au problème de détection associé (test Bayésien, test minimax, ou test de de Neyman-Pearso...). Dans ce contexte, il existe plusieurs travaux qui ont traité ces deux problématiques. Le lecteur intéressé par plus de détails pourra se référer aux références suivantes [57] [177] .

3.7.2 La stéganalyse du point de vue de la théorie des jeux

Le problème de stéganalyse/stéganographie peut également être abordé sous l'angle de la théorie des jeux. Notons que dans ce cadre, l'insertion et la stéganalyse ne sont plus totalement déterministes. De manière générale, la théorie des jeux est une approche très intéressante, lorsqu'il s'agit de modéliser la stratégie de chacun des participants d'un jeu compétitif. Elle permet de prendre en compte le comportement de deux (ou plusieurs) opposants qui doivent adapter leurs stratégies en fonction d'hypothèses sur le comportement des autres adversaires dans le jeu. Le principe général de cette approche est de considérer le scénario étudié comme un problème d'optimisation, où chaque participant tente de maximiser ses gains et minimiser ses pertes dans cette compétition. Si nous pouvons modéliser le problème tel qu'il existe *un équilibre de Nash*, alors chaque joueur dispose d'une stratégie

optimale, et aucun des joueurs ne peut changer sa stratégie sans affaiblir sa position personnelle par rapport aux autres [122]. Pour un contexte de stéganographie/stéganalyse, [53] présente les différents acteurs du jeu comme étant : l'environnement, le stéganographe , le stéganalyste, et le juge (ou le maître) du jeu.

Par la suite, toujours dans le même esprit, [31] ont développé la première méthode pratique, basée sur la théorie des jeux, pour l'insertion adaptative d'un message secret.

3.8 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 différentier 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 qui suit présente quelques outils d'apprentissage et de classification appliquée à la stéganalyse universel.

Chapitre 4

Méthode d'apprentissage pour la Stéganalyse

4.1 Introduction

Pour laquelle le stéganalyste dispose au préalable d'une large base de données (dans notre cas une base d'images), et tel que la classe de chacun de ses éléments est connue d'avance (classe *cover,* ou classe *stego*). Les images utilisées doivent être de la même dimension, lors de cette phase, le stéganalyste procède d'abord à l'extraction des caractéristiques de chacun des média composant la base d'images. Ensuite il choisit un classifieur donné et règle ses paramètres (Par exemple, le taux de fausses alarmes est fixé au minimum), pour discriminer le plus précisément possible les deux classes d'objets, à partir des caractéristiques extraites. À la fin de cette première phase, le détecteur est opérationnel, et peut alors être utilisé pour la classification.

De cette description, il ressort deux choix importants lors de la conception d'une attaque aveugle par apprentissage. Le premier choix crucial étant les caractéristiques utilisées, qui doivent être pertinentes pour la discrimination des classes. Le deuxième choix est celui du classifieur (SVM linéaires ou non-linéaires, Réseaux de neurones, discriminants linéaires de FISCHER, ...ou autres), qui doit être efficace lors de la classification.

4.2. Les méthodes d'apprentissage automatique

Après avoir extrait les caractéristiques pertinentes pour la discrimination, l'étape suivante est le choix puis le régalage du classifieur pour la détection. Dans cette section, nous présentons quelques outils d'apprentissage et de classification appliquée à la stéganalyse universel.

4.2.1 SVM

Le Séparateur à Vaste Marge, également appelé machine à vecteur support (en anglais Support Vector Machine (SVM)), est un outil pour la classification et l'apprentissage super-visé, introduit par V. Vapnik en 1995 [122] [24]. Cet outil de classification puissant, qui découle de la théorie statistique de l'apprentissage, repose sur une théorie mathématique solide. Le principe général de son fonctionnement consiste à trouver un classifieur, ou une fonction de discrimination, dont la capacité de généralisation est la plus grande possible.

Les Machines à Vecteurs de Supports (SVM) [1, 12] permettent d'obtenir une frontière (La frontière représente dans un espace à X dimensions la séparation entre les différentes classes). (Optimale) pour séparer les données en deux classes stego et cover. Toutefois, certaines données ne sont pas linéairement séparables. Dans ce cas-là, il est nécessaire d'utiliser un espace de redescription (c'est-à-dire un espace de dimension supérieure). La

frontière (ou hyperplan) sera ensuite recherchée dans ce nouvel espace de dimension supérieure. Pour se faire, il faut utiliser le kernel trick, c'est-à-dire une fonction noyau permettant de généraliser l'approche linéaire [153]. La classification par SVM fonctionne très bien dans le cas binaire, classification cover ou stego. Pour nos expérimentations, nous utiliserons un noyau non linéaire décrit par Ivans Lubenko et Andrew D. Ker de type gaussien[12]. La complexité du noyau gaussien est quadratique. Avec l'utilisation de libsvm la complexité varie de $\mathbf{O\ (d.N^2)}$ à $\mathbf{O\ (d.N^3)}$ en fonction de l'utilisation de la mémoire cache, avec d la dimension des vecteurs caractéristiques et N le nombre de données d'entrainement [47].

a). SVMs binaire

Initialement conçu pour la discrimination binaire (classification à deux classes), l'objectif d'un classifieur SVM est de chercher l'hyperplan de marge optimale qui, lorsque c'est possible, classe ou sépare correctement les données tout en étant le plus éloigné possible de toutes les observations. Autrement dit, chercher l'hyperplan séparateur qui maximise la distance « marge » entre les deux classes (−1,+1), de sorte à minimiser la probabilité de mauvaise classification d'un élément qui ne serait pas dans le bon ensemble. Dans le cas de la stéganalyse, bien évidemment les données à classifier sont les vecteurs caractéristiques des images, et les deux classes (−1,+1) représentent respectivement la classe *cover* et la classe *stégo*. La Figure 4.1 illustre un exemple de discrimination linéairement séparable pour un classifier à marge maximale. Sur la Figure4.1 l'hyperplan séparateur optimal est représenté par la droite.

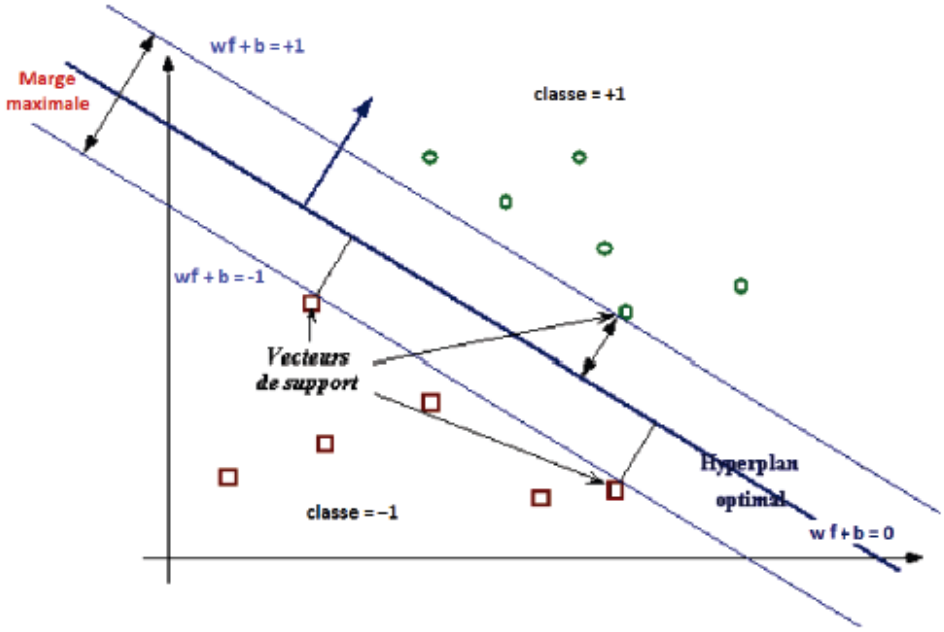
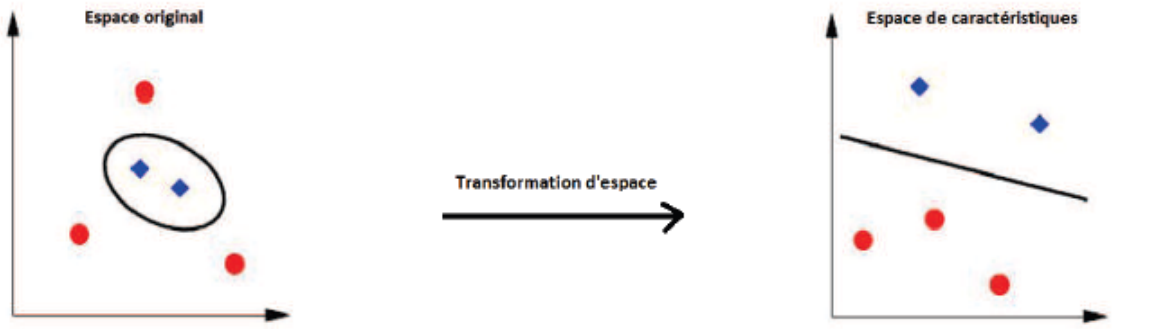


Figure 4.1 – Illustration du principe de fonctionnement d'un SVM binaire dans le cas d'un problème linéairement séparable.

L'approche par SVM consiste à transformer le problème de séparation non linéaire dans l'espace de représentation initial, en un problème de séparation linéaire dans un nouvel espace de représentation (espace de re-description) de plus grande dimension. Autrement dit, amener l'espace caractéristique initial des données, vers un autre espace caractéristique de dimension plus grande (voir l'exemple sur la Figure 4.2). Plus la dimension de l'espace de re-description est grande, plus la probabilité de trouver un hyperplan séparateur optimal entre les observations est importante [68,55]. Cette transformation d'espace est effectuée grâce à une fonction particulière appelée «Noyau » (en anglais *Kernel*). Il existe différents types de noyaux, dont certains couramment utilisés : polynomiale, gaussien, sigmoïde, ou laplacien. Une fois le noyau choisi, la fonction objective à optimiser peut alors être calculée comme suit:



(a) Cas non linéairement séparable (b) Cas linéairement séparable

Figure 4.2 : Exemple illustrant le principe de résolution, pour un SVM, dans le cas où les données sont non-linéairement séparables [55].

$$\hat{\alpha} = \arg\max_{\alpha} \left(\sum_{i=1}^{I'} \alpha_i - \frac{1}{2} \sum_{i=1}^{I'} \sum_{j=1}^{I'} \alpha_i \alpha_j c_i c_j K\left(f_i, f_j\right) \right) \qquad (4.1)$$

avec $I' > I$ la dimension de l'espace de re-description, les $\propto i$ (resp. $\propto j$) des multiplicateurs de Lagrange satisfaisant les contraintes $\propto i > 0, \sum_{i=1}^{I'} \alpha_i c_i = 0$, et tel que K (f_i,f_j) une fonction noyau définie par :

$$K : \begin{array}{l} \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \\ f_i, f_j \to k\left(f_i, f_j\right) \end{array} \qquad (4.2)$$

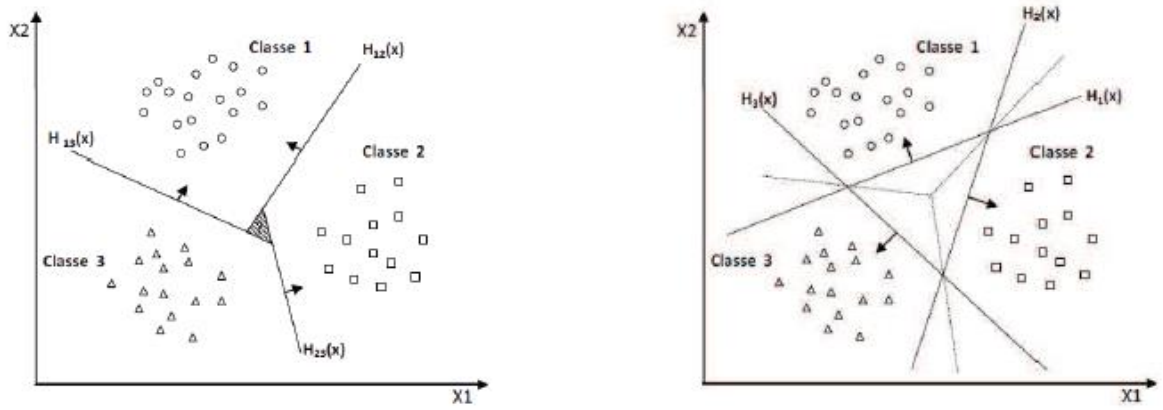
La fonction de décision quant à elle devient :

$$\phi\left(f\right) = \sum_{i=1}^{I'} \alpha_i c_i k\left(f_i, f\right) + b \qquad (4.3)$$

b). SVM multiclasses

Dans le cas de classification multiclasses, on ne dispose plus de deux classes, mais de plusieurs classes. L'objectif est donc d'affecter une nouvelle observation à l'une des plusieurs classes. Autrement dit, la décision n'est plus binaire et l'utilisation d'un seul hyperplan séparateur n'est plus suffisante. Pour les machines à vecteur support qui sont à la base des classifieurs binaires, la résolution du problème multiclasses consiste à réduire le problème initial à une composition de plusieurs hyperplans biclasses permettant de tracer les frontières

de décision entre les différentes classes. En d'autres termes, décomposer l'ensemble des observations en plusieurs sous-ensembles représentant chacun un problème de classification binaire. Une fois réalisé, la décision finale de la classe d'un élément est effectuée grâce à un processus hiérarchique. Dans cet esprit, il existe plusieurs travaux proposant différentes approches de décomposition : une classe-contre-une autre, une classe contre-reste,...etc (voir la Figure 4.3).



(a) Approche une classe contre une autre (b) Approche une classe contre le reste

Figure 4.3 – Exemple illustrant quelques approches de décomposition pour un classifieur SVM multiclasses [36].

Notons au passage qu'il est envisageable d'utiliser les SVMs muticlasses pour le scénario de stéganalyse universelle avec ou sans *cover-source mismatch*. Parmi les travaux qui ont exploré cette piste, on retrouve [163,131].

c). SVM monoclasse (OC-SVM)

Dans les machines à vecteur support binaires ainsi que les SVMs muticlasses présentés précédemment, nous avons toujours deux classes : une classe négative représentant la classe des images de couverture, et une classe positive représentant la classe des images stégo. De telles informations ne sont pas toujours disponibles en stéganalyse. À titre d'exemple, pour les scénarios de stéganalyse universelle avec ou sans *cover-mismatch*, il est très coûteux, voire impossible, pour la gardienne Eve de construire une base d'images stégo couvrant tous les cas possibles pour son apprentissage. Dans un tel cas, il est souhaitable d'avoir un modèle de décision permettant de distinguer les images de couvertures originales des autres images

modifiées (les images aberrantes ou *outliers*). Pour cela, Eve doit avoir à sa disposition un modèle statistique assez complet pour décrire la nature statistique des images de couverture.

Pour la classification SVM mono classe, il est supposé que seules les images de la classe *cover* sont disponibles. L'objectif est donc de trouver une frontière qui sépare les images de couverture du reste de l'espace. Autrement dit, trouver un hyperplan optimal qui sépare les observations de la classe cible "les images de couvertures" des observations aberrantes "les images stégo". La figure 4.5 représente, en deux dimensions, un exemple de problème de classification pour un SVM monoclasse.

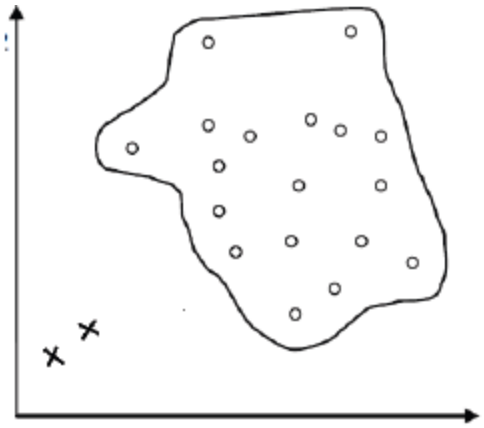


Figure 4.4 – Illustration du principe de fonctionnement d'un classifieur SVM monoclasse.

4.2.2. Average Perceptron.

Pour gérer les grands jeux d'apprentissage, une solution consiste à utiliser un classifieur linéaire. Comme le montrent Ivans Lubenko et Andrew D. Ker [13] en 2012, l'approche par perceptron est très prometteuse. Un perceptron ou neurone, est un classifieur online permettant de séparer linéairement les données [3]. Le neurone est consisté d'un vecteur poids, c'est-à-dire d'un vecteur $w \in \mathbb{R}^d$. Ce vecteur représente la séparation entre les images cover et stego, initialement ce vecteur vaut $\vec{0}$ (vecteur nul).

L'apprentissage s'effectue par mises à jour successives. La mise à jour comprend le vecteur caractéristique $f_i \in \mathbb{R}^d$ avec $i\{0..N\}$, N étant le nombre d'images dans la base de

données et le type de l'image $l_i \in L$. L'algorithme classe le vecteur $\mathbf{f_i}$, le résultat est noté *z*. Pour cela, on analyse la position du point $\mathbf{f_i}$ par rapport à la séparation **w**, voir formule 4.5 :

$$z = sign(w.f_i) \qquad (4.5)$$

Avec **sign** la fonction retournant le signe.

Une représentation *2D* de cette classification est présentée dans la figure 4.5. En effectuant la projection de la formule (4.5), nous avons représenté les deux possibilités. Sur la figure 4.5 de gauche, on voit que la projection de f est positive sur w, comme le label est 1 il n'y aura aucune mise à jour. La figure 4.5 de droite représente un deuxième scénario où la projection est négative sur w, par conséquent il y aura une mise à jour.

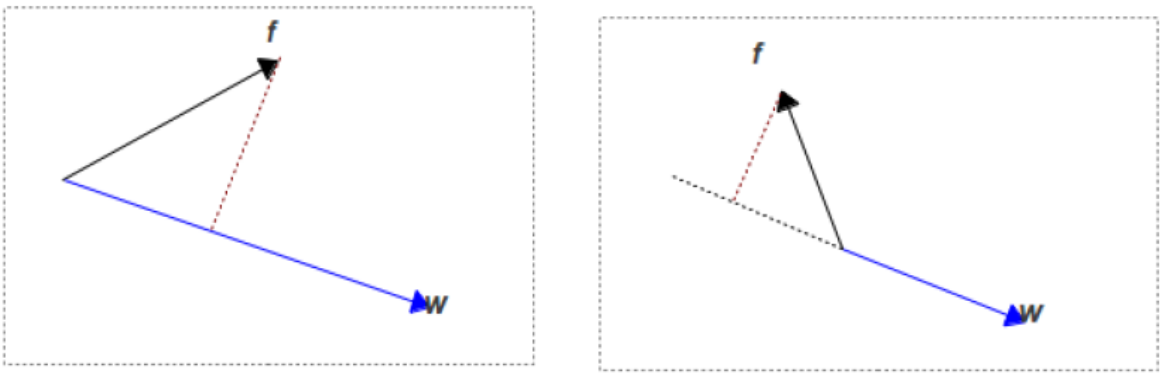


Figure 4.5 : Soient w le vecteur poids du perceptron et **f** le vecteur caractéristique à classer possédant un label $\mathcal{L}$ de 1.

Deux cas sont possibles : 1- l'algorithme de classification a correctement placé $\mathbf{f_i}$: aucune mise à jour n'est effectuée. 2- Sinon il faut modifier le poids du neurone, comme l'indique la formule 4.6.

$$\begin{cases} \text{Si } \; l_i = z \; \text{ alors aucune mise à jour} \\ \text{Si } \; l_i \neq z \; \text{ alors } \; w = w + l_i.f_i \end{cases} \qquad (4.6)$$

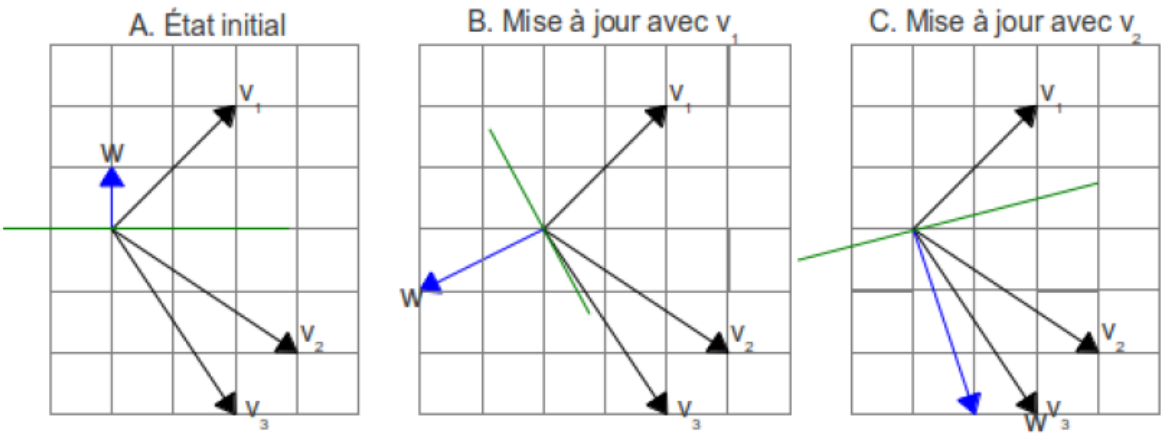


Figure 4.6 : Dans cette illustration, v1, v2, v3 sont des vecteurs quelconques avec respectivement -1, 1 et 1 comme classe L. w est le poids du neurone étudié, la droite en vert représente la droite orthogonale à w.

L'évolution d'un perceptron est schématisée en deux dimensions dans la figure 4.6 : la figure 4.6.B représente l'évolution de la figure 4.6.A après la mise à jour de v1. La figure 4.6.C représente la mise à jour v2 réalisée à partir de la figure 4.6.B. Notons que sur la figure 4.6.B v3 n'apportera aucune mise à jour car son label serait prédit.
Ce classifieur permet sur les grands ensembles de données de créer une frontière linéaire. Lors de la phase d'apprentissage, cet algorithme repose sur un vecteur initial, appelé vecteur poids, noté w = $\vec{0}$, sur un vecteur représentant la somme des poids, noté w^{sum} et sur un vecteur moyen, avec N_{app} le nombre d'apprentissages que le perceptron a réalisé.

Le perceptron prédit le label à partir d'un vecteur caractéristique $\mathbf{f_i}$, auquel on veut associer un label $l_i \in \{cover = 1, stego = -1\}$. La prédiction notée *z* est calculée selon l'équation 4.5. Le calcul de *z* se fait par un produit scalaire entre le vecteur caractéristique d'entrée $\mathbf{f_i}$, et le vecteur moyen des poids, c'est-à-dire $\mathbf{w}^{avg}$. En fonction du signe de ce produit scalaire, on détermine de quel côté de l'hyperplan orthogonal à $\mathbf{w}^{avg}$ se trouve $\mathbf{f_i}$ et on en déduit la classe du vecteur (stego ou cover).

Lors de l'apprentissage, pour chaque vecteur caractéristique $\mathbf{f_i}$, de classe réelle li, si la prédiction est correcte, aucune mise à jour n'est effectuée sur le vecteur poids w, sinon le vecteur **w** est mis à jour, voir équation 4.6. Une fois la mise à jour effectuée, on met à jour le vecteur $\mathbf{w}^{sum}$, puis le processus recommence avec un nouveau vecteur caractéristique en entrée

jusqu'à la stabilisation du vecteur de poids. Ce vecteur de poids représente la séparation entre les différentes classes, dans notre cas il s'agit de la frontière entre les images stego et cover.

Comme l'a prouvé Novikoff [3], tout perceptron converge après un nombre fini d'itérations si les données sont linéairement séparables. Une autre version du perceptron existe : le batch iterated perceptron [13], celui-ci consiste à répéter k fois les j premières mises à jour, k et j étant des paramètres fixés par l'utilisateur. Ce processus permet d'accélérer la convergence. La complexité de chaque mise à jour est *O(d),* ce qui équivaut pour l'apprentissage total à une complexité de *O(d.N).*

4.3 Ensemble classifieur

Une autre approche, proposée par Kodovsky en 2011 consiste à utiliser un ensemble de classifieur [104, 9]. Le fonctionnement est simple et consiste à obtenir un vote de m classifieurs de faible complexité.

L'ensemble classifieur s'apparente donc à une forêt d'arbres décisionnels, c'est-à-dire un classifieur qui exécute de multiples sous-classifieurs sur des données légèrement différentes. Deux paramètres sont à définir : le nombre L représentant le nombre de classifieurs faibles et la taille d_{red} de chaque sous-échantillonage du vecteur caractéristique **f** (utilisé en entrée de l'ensemble classifieur). Nous définissons, chaque classifieur faible comme étant une fonction h_i avec i $\in \{0..L\}$ telle que :

$$h_i(f): f_{red} \to L$$

$$\mathbb{R}^{d_{red}} \to \{-1,1\} \qquad (4.7)$$

La décision finale $\mathcal{S} \in \mathcal{L}$ par vote majoritaire de l'ensemble classifieur, pour un vecteur **f**, est définie dans l'équation 4.8 :

$$S = sign\left(\sum_{i=0}^{L}\left(h_i(f)\right)\right) \qquad (4.8)$$

Chaque classifieur de base voit son apprentisssage s'effectuer sur un unique sous-échantillon de **f**, l'apprentissage se fait indépendamment des autres classifieurs. La figure 4.8 résume le processus global d'apprentissage de l'ensemble classifieur. Notons que ces classifieurs faibles peuvent être un neurone ou n'importe quel autre type de classifieur.

Dans leur article Kodovský et Fridrich [169] proposent une méthode d'automatisation du choix de L et d_{red}. Pour cela, ils utilisent une technique bootstrap, qui consiste à apprendre sur

un certain nombre d'échantillons $N_{simulation}$ de la base de données. Pour la suite 67% de $N_{simulation}$, noté $N^{a}_{simulation}$ est utilisé pour l'apprentissage et 33%, noté $N^{b}_{simulation}$ est utilisé pour les tests. Dans un premier temps, ils définissent l'erreur dite out-of-bag, noté **EOOB** qui représente pour chaque couple d'images x_i et y_i avec i $\in \{0..N^{b}_{simulation}\}$ la moyenne de la somme des erreurs de chaque classifieur faible. Notons que cette formule est ici modifiée par rapport à l'article afin d'homogénéiser les notations.

$$E_{OOB} = \frac{1}{2.N^{b}_{simulations}} \sum_{i=0}^{N^{b}_{simulations}} \left(1 - \frac{h(y_i)}{2}\right) \qquad (4.9)$$

La première étape consiste à déterminer le nombre L de classifieur faible.

Dans l'article [156], les auteurs constatent que pour un nombre fixé d_{red}, l'erreur *out-of-bag* converge asymptotiquement lorsque que L croît. Il sufit de prendre la valeur minimale de L tel que l'erreur classifieur soit au plus proche du point de convergence. La figure 4.7 illustre le choix du paramètre L.

La seconde étape est le choix de d_{red} sachant L (fixé dans l'étape précédente).

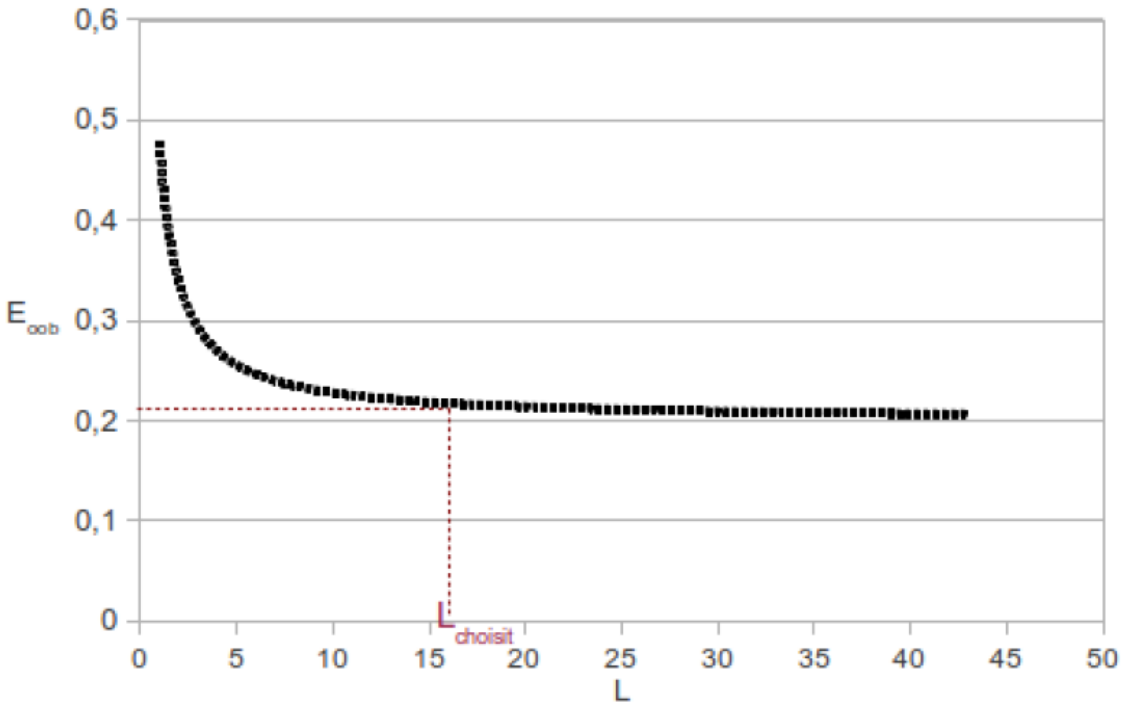


Figure 4.7 : Représentation de l'allure de l'erreur out-of-bag pour un nombre fixé de caractéristiques d_{red} en fonction de L [36].

L'évolution de l'erreur en fonction de d_{red} n'est pas strictement décroissante. Elle passe par un minimum, puis va re-augmenter. Le paramètre d_{red} correspond au minimum de la fonction erreur. D'après [74], ce minimum est tel que $d_{red} << d$.

L'ensemble classifieur de Kodovsky utilise l'analyse du discriminant linéaire de fisher (FLD). Cette méthode permet par réduction de dimensions de maximiser la compacité des classes (cover ou stego), c'est-à-dire de regrouper lors de projection les images d'une même classe. En pratique, cela équivaut à calculer la matrice de covariance de taille d_{red}^2 de chaque vecteur f_{red} et donc la

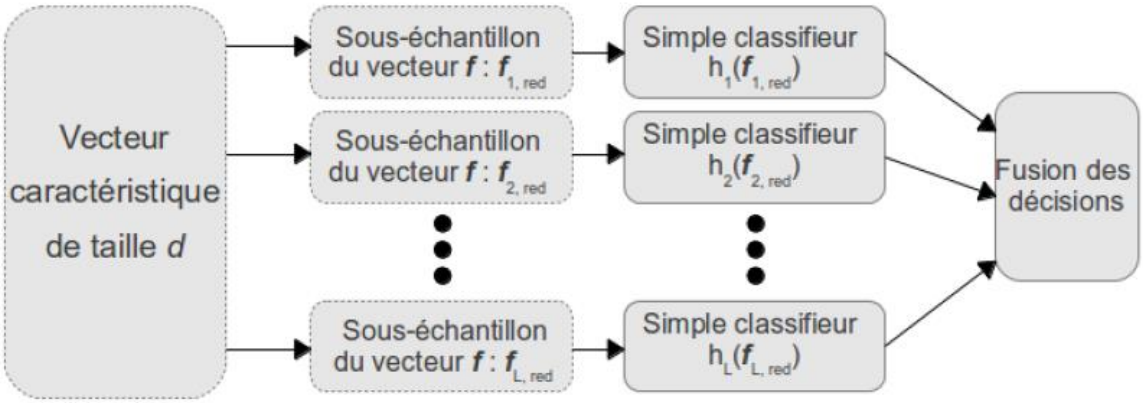


Figure 4.8 -Représentation du fonctionnement de l'ensemble classifieur [104].

complexité de ce classifieur est $O(d_{red}^2 .L.N)$. L'ensemble average perceptron est un ensemble classifieur où chaque classifieur simple est un perceptron. La complexité d'un tel classifeur est alors $O(d_{red}.L.N)$. L'ensemble classifieur est une alternative prometteuse au SVM pour la stéganalyse.

4.4 L'ensemble de classifieurs FLD

Pendant plusieurs années, et ce jusqu'à 2011, en stéganalyse pour des vecteurs caractéristiques de taille petite et moyenne, le meilleur outil d'apprentissage et de classification était les SVMs avec noyau gaussien [22].

De nos jours, avec l'augmentation de la dimension des espaces caractéristiques en stéganalyse, les machines à vecteurs supports (SVMs) ne sont plus adaptées. Pour cela, de nouveaux outils pour l'apprentissage et la classification ont été proposé comme alternative.

Parmi ces outils, nous retrouvons l'ensemble de classifieurs FLD [145], que nous présentons maintenant.

Les schémas stéganographiques modernes tel que HUGO [147] ont tendance à utiliser des espaces de caractéristiques de plus en plus grand , ce qui constitue un vrai problème pour la stéganalyse. Afin de résoudre ce problème, [106] proposent un nouvel outil d'apprentissage et de classification alternative aux outils classiques tel que les SVM, ou les réseaux de neurones. Leur classifieur est basé sur l'utilisation d'un ensemble de classifieurs de faible complexité. Ils utilisent pour l'apprentissage et la classification un ensemble F = {F1, ...,FL} de classifieurs FLD (*Fisher Linear Discriminant*) binaires.

Dans le but de réduire la complexité de calcul, l'apprentissage et la classification de chaque classifieur FLD est effectué sur un sous-espace de caractéristiques de dimension dred avec (dred ≪d). En pratique, avant la phase d'apprentissage chaque classifieur FLD choisit pseudo-aléatoirement un sous-ensemble de caractéristiques (dred caractéristiques) à partir de chaque vecteur caractéristique $f_i \in Rd$.

Lors de *la phase test*, une observation f donnée en entrée de cet ensemble de classifieurs est classée par chaque classifieur. Chaque classifieur FLD retourne alors une décision binaire indiquant le numéro de la classe attribuée à cette observation. La décision finale est obtenue en fusionnant par vote majoritaire les résultats des différents classifieurs FLD.

Lors de *l'apprentissage*, le seuil de décision de chaque classifieur FLD est ajusté, afin de minimiser l'erreur totale de détection (PE) sur les données d'apprentissage. Dans cet esprit, [106] proposent de définir la probabilité d'erreur de détection P_E comme suit :

$$P_E = \min_{P_{FP}} \frac{1}{2}\left(P_{FP} + P_{FN}\left(P_{FP}\right)\right) \qquad (4.10)$$

avec P_{FP} la probabilité de faux positif et P_{FN} la probabilité de faux négatif. Plus la probabilité d'erreur (PE) est petite, plus la classification en deux classes est meilleure.

L'utilisation des ensembles de classifieurs de faible complexité fut une grande innovation en stéganalyse. A. Ker et I. Lubenko ont réutilisé ce concept pour la stéganalyse à clairvoyance avec *cover-source mismatch*. Dans [114] [174], ils ont proposé un nouvel classifieur qui est *l'ensemble average perceptron*. Un peu plus tard, pour ce même scénario,

les auteurs de [175] ont montré qu'une adaptation de l'ensemble de classifieurs FLD de Kodovský permettrait d'obtenir de meilleurs résultats.

4.4.1 Ensemble FLD avec sélection de caractéristiques

Dans [174], les auteurs émettent l'hypothèse que pour chaque classifieur simple, de type FLD, certaines caractéristiques du sous-échantillonnage du vecteur caractéristique sont moins importantes que d'autres. Dans la méthode ensemble FLD avec sélection de caractéristiques, noté EFLDFS, les auteurs déterminent cinq métriques permettant de sélectionner ces caractéristiques. Chaque métrique donne un score à chaque caractéristique du vecteur f, noté c_i^n avec i le numéro de la métrique i $\in \{1..5\}$ et n le numéro de la caractéristique avec n $\in \{1..d_{red}\}$ avec d_{red} la taille du sous-échantillonnage du vecteur caractéristique **f**. Pour chaque score, les auteurs suppriment les caractéristiques de façon à ne pas augmenter la probabilité d'erreurs. Les tests s'effectuent donc sur un espace réduit. Sur la base de données homogène BOSSv16[175] les auteurs obtiennent un gain moyen sur le rappel de 1.7%.

4.5 Comparaisons des principales approches

Comme nous l'avons décrit dans la partie précédente, les algorithmes d'apprentissage ne peuvent pas, de part leur complexité et mécanismes différents être exécutés sur la même base d'apprentissage. Sachant que la taille de la base d'apprentissage n'est pas la même, il est difficile de les comparer. Nous allons donc donner quelques résultats de comparaison provenant de [3, 132, 6], mais la comparaison ne sera pas faite entre toutes les approches.

Les résultats que nous présentons ici proviennent de [3, 132]. L'algorithme d'insertion utilisé est nsF5 [61]. Cet algorithme insère les données dans les blocs issus d'une transformation en cosinus discret, qui a lieu lors d'une compression jpeg. Celui-ci est donc particulièrement efficace pour les images jpeg. Nous allons nous placer successivement dans deux cas utilisant des bases de données différentes.

Dans leur article [3], Ivans Lubenko et Andrew D. Ker, récupèrent à l'aide d'un réseau social 800 000 images, toutes compressées identiquement provenant de 200 utilisateurs différents. Cette base de données est très hétérogène, mais les données d'entrainement et de tests, bien que différentes proviennent des mêmes utilisateurs. Nous ne sommes donc pas dans un cas de *cover-source mismath* total [14]. Les résultats présentés dans la figure 4.5

proviennent de [3]. Sur cette figure, ils représentent l'exactitude de déférents algorithmes en fonction de la taille de la base de données d'apprentissage.

Nous constatons que dans ce scenario, l'algorithme SVM converge très rapidement (avec uniquement 20 000 données d'apprentissage) vers l'asymptote avec 93.5% de prédiction correcte. Toutefois, son utilisation est impossible avec un plus grand jeu de données. Le perceptron converge aussi mais avec beaucoup plus de données d'apprentissage (1.6 million) et oscille très fortement dans ses résultats devenant peu fiables après certaines mises à jour.

L'ensemble classifieur de Kodoský converge aussi rapidement que les méthodes SVM vers l'asymptote. Par contre, l'ensemble classifieur permet de traiter de plus grands ensembles de données, car sa complexité est plus faible, pouvant donc potentiellement être plus adapté pour traiter les problèmes de *cover-source mismatch* [14]. La méthode d'ensemble classifieur avec des perceptrons converge également et donne des résultats sensiblement meilleurs que les ensembles FLD ou SVM, cependant il est nécessaire d'effectuer l'apprentissage sur les 1.6 million d'images.

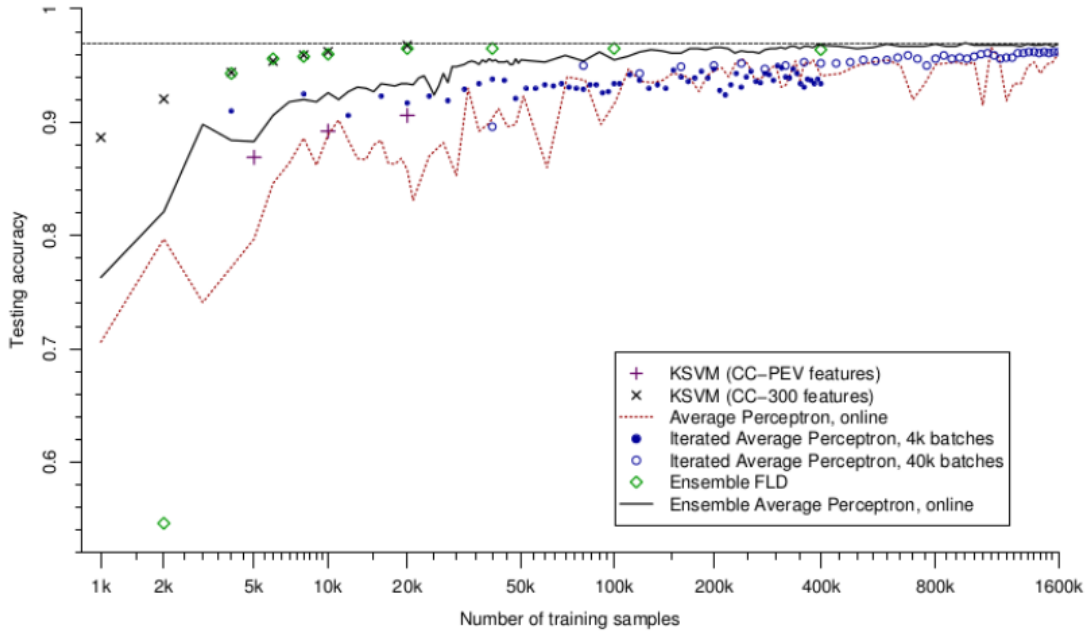


Figure 4.9 : Résultats des expériences avec une charge de 0.1 bpnc (bit par coefficient non nul [165].

D'après la figure 4.7, qui traite une base d'images hétérogènes, l'ensemble classifieur ne classe pas mieux une image stego ou cover que l'algorithme plus classique SVM. Nous allons étudier maintenant à une deuxième expérience.

Dans [14], Ivans Lubenko et Andrew D. Ker s'intéressent au cas du cover- source mismatch. Les données testées par les algorithmes sont différentes avec les données d'apprentissage :

Algorithme de classification	Exactitude des tests	Nombre de données apprentissage
KSVM	80.9%	6 000
Ensemble classifier FLD	83.6%	20 000
Ensemble Average perceptron	81.2%	1 000 000
Meilleur Ensemble Average perceptron	85.1%	400 000

Tab 4.1 : Résultats des classifieurs dans le cas du cover-source mismatch. [14].

Lors de l'apprentissage le stéganalyste ne connaît pas le type d'image utilisé par le stéganographe. Dans ce scénario, nous nous rapprochons d'un cas réel de stéganalyse de type analyse automatique de trafic.

Le tableau 4.1 représente les résultats de plusieurs algorithmes dans le cas d'une insertion plus faible (0.05bpnc avec l'algorithme nsF5). Du fait d'une insertion différente, nous ne pouvons pas comparer directement l'expérience précédente avec le tableau 4.1. Toutefois, comme nous le constatons, dans ce nouveau scenario les algorithmes de type SVM sont nettement moins efficaces en terme d'exactitude que les algorithmes d'ensemble classifieur. Nous constatons même que l'ensemble average perceptron, devient plus intéressant que l'ensemble FLD lors de l'apprentissage sur 400 000 données.

L'hypothèse impliquant que l'utilisation d'ensemble classifieurs linéaires pour traiter de grandes bases de données et permettant ainsi de mieux lutter face v au problème du cover-source mismatch [14,6] semble être prometteuse. Il est donc nécessaire de vérifier cela avec des conditions plus difficiles : images non compressées et insertion plus performante.

4.6 Classifieur OP-ELM

Le classifieur OP-ELM (Optimally-Pruned Extreme Learning Machine) [13, 166] est une variante de l'Extreme Learning Machine original de Huang [129]. Ce classifieur utilise des réseaux de neurones à couches multiples dont les coefficients sont initialisés aléatoirement.

L'OP-ELM ajoute au modèle original d'autres noyaux ainsi qu'une étape permettant de supprimer les neurones les moins utiles du réseau. Les deux théorèmes sur lesquels l'ELM de Huang est basé ne seront pas détaillés ici mais peuvent être trouvés dans [129]. L'idée principale est d'utiliser un réseau de neurones avec une seule couche cachée (Single Hidden Layer Feedforward Neural Network), pour lequel les coefficients et les biais sont initialisés aléatoirement. La Figure 4.10 présente un modèle simplifié de réseau de neurones.

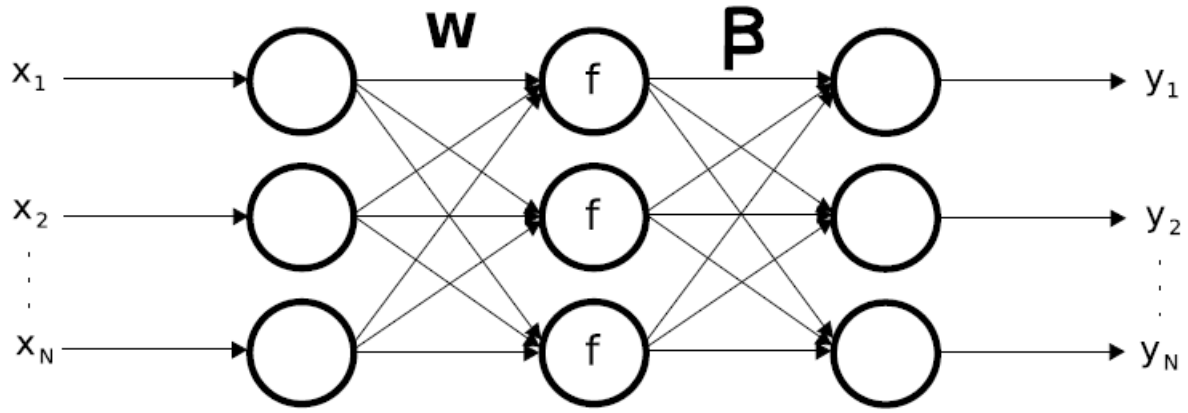


Figure 4.10 : Schéma d'un réseau de neurones possédant une seule couche cachée et ne possédant pas de boucle (récursivité). Les valeurs d'entrée X = (x1, . . . , xN) sont pondérées par les coefficients **W**. Un biais **B** peut être ajouté (non représenté ici) et le résultat passe par une fonction d'activation f, laquelle est enfin pondérée par les coefficients de sortie β pour obtenir la sortie Y = (y_1, . . . , y_N).

Les coefficients **W** et **B** (biais) sont initialisés aléatoirement par l'ELM. Les nouveautés apportées par l'OP-ELM sont une plus grande robustesse de l'ELM original face à des données ayant certaines dimensions fortement corrélées, ainsi que l'utilisation d'autres fonctions f permettant d'utiliser OP-ELM pour des cas où le modèle à approximer possède des composantes linéaires, par exemple.

L'étape de validation de ce classifieur est effectuée par un Leave-One-Out, bien plus précis qu'une k-fold, et ne nécessite pas de test [8]. Il a été montré sur un certain nombre d'applications expérimentales [13, 166], que l'OP-ELM donne des résultats très proches de ceux d'une SVM (Support Vector Machine) et se comporte d'une façon similaire, tout en ayant l'avantage de temps d'exécutions 10 à 100 fois plus faibles.

4.7 Conclusion

Dans ce chapitre nous avons présenté les plus importants outils de classification. Nous avons vu également que pour lutter contre les schémas de dissimulation récents, qui préservent des caractéristiques de haute dimension, la stéganalyse actuelle utilise à son tour un nouveau concept qui est l'ensemble de classifieurs FLD [106]. Dans le chapitre suivant (chapitre 5), nous proposons une méthodologie permettant de réduire la dimensionnalité de l'ensemble de caractéristiques, grâce à la sélection de caractéristiques et d'obtenir des résultats fiables par la détermination d'un nombre suffisant d'images pour le problème. Et la sélection de caractéristiques donnant une interprétabilité accrue aux résultats du classifieur. Cette méthodologie est testée en chapitre 6 sur six méthodes de stéganographie différentes et pour quatre taux d'insertion différents. Les résultats en classification sont ensuite interprétés grâce à aux caractéristiques sélectionnées par la méthodologie comme les plus pertinentes pour chaque méthode de stéganographie.

Chapitre 5

Caractéristiques sélectionnés de la stéganalyse

5.1 Introduction

Les techniques de stéganalyse spécifiques nécessitent de connaître les détails sur les méthodes de stéganographie ciblées, mais la stéganalyse universelle [174] nécessite moins ou même pas une telle information à priori. Une approche de stéganalyse universelle est illustré à la figure 5.1, qui prend habituellement une stratégie basée d'apprentissage et implique une étape de test : Sa fonction est de cartographier une image d'entrée dans un espace de grande dimension à une autre de petite dimension. Au cours du processus, une étape d'extraction de caractéristique est utilisée ,c'est la formation, l'objectif de cette étape est d'obtenir un classifieur formé, comme discriminante linéaire de Fisher (FLD), machine à vecteurs de support (SVM), réseau neuronal (NN), etc, cette approche peut être caractérisée par des caractéristiques numériques, et les distributions des caractéristiques des images de couverture sont probablement différentes de celles de leurs images stego[21].

Nous proposons une méthodologie (dont le principe est illustré à la Figure 5.1) permettant de déterminé un nombre suffisant d'images pour un entraînement « fiable » (en termes de variance des résultats de classification) du classifieur utilisé, puis par une sélection de caractéristiques donnant une interprétation accrue aux résultats du classifieur. Cette méthodologie est testée en chapitre 6 sur six méthodes de stéganographie différentes et pour quatre taux d'insertion différents. Les résultats de classification sont ensuite interprétés grâce aux caractéristiques sélectionnées par la méthodologie comme les plus pertinentes pour chaque algorithme de stéganographie.

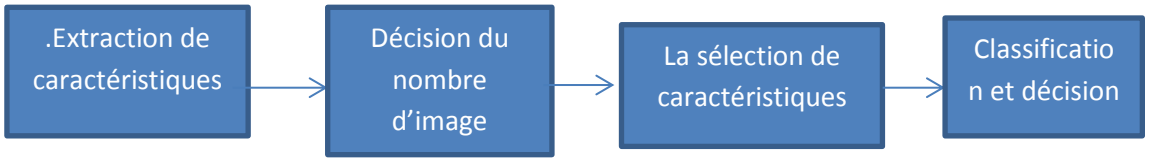


Figure 5.1 *Schéma de la méthode classique de stéganalyse pour une image*

5.2 L'extraction des caractéristiques dans la stéganographie

Les Caractéristiques sont généralement construites selon les principes heuristiques, dans un but de capturer des petites modifications en raison de l'intégration de la stéganographie. L'idée en utilisant le classifieur pour détecter la Stéganographie, qui a été proposé par Avcibas et al. [47,70]. Ces auteurs utilise des mesures de la qualité des caractéristiques de l'image et testé leur système à plusieurs algorithmes.

Par la suite, J. Wanga [177] a proposé un ensemble de fonctionnalités différent basée sur les mesures de similarité binaires entre le bit le plus bas pour classer les images de couverture et les images stego.

Ensuite, Farid [115] construit des caractéristiques d'ordre supérieur des coefficients d'ondelette de haute fréquence à plusieurs groupes et leurs erreurs de prédiction linéaire. Ainsi que Lyu et al dans [162] ont proposé une stéganalyse universelle basée sur les statistiques d'ondelettes pour une échelle de gris des images. Les quatre premiers moments statistiques de haute fréquences du sous-bandes locales de coefficients d'ondelettes et leurs erreurs de prédiction linéaire ont été utilisés pour former 72 - caractéristiques dimensionnelles (72-D) pour le vecteur de stéganalyse .

Pour détecter la stéganographie, Barbier et al [95] proposent une nouvelle méthode de bruit additif en utilisant le centre de masse (COM) de la fonction d'histogramme (FHC). Cependant, seul un petit nombre de caractéristiques a été extrait et la performance n'est pas satisfaisante. Presque dans toutes les méthodes de la stéganographie précédente le taux de détection est encore élevé, un nombre limité de caractéristiques ne pouvait pas obtenir une bonne précision de la classification.

Fridrich [6] a proposé le concept de calibrage (voir la section5.3.1), le principe de cette méthode d'analyse, consiste à estimer l'histogramme des coefficients DCT de l'image de couverture (l'image originale sans aucune modification) à partir de l'image stéganographiée, pour que le taux de détection soit positif pour certains algorithmes de stéganographie populaires. Par la suite, toutes les capacités récoltées dans le champ de DCT ne sont pas suffisantes et la précision de détection n'est pas satisfaisante pour les images stego avec certains algorithmes de stéganographie comme Jphide et steghide [4].

Shi [176] propose un nouvel ensemble de fonctionnalités, ces caractéristiques est défini comme un modèle de différences entre les valeurs absolues des coefficients DCT et un processus de Markov, y compris que la précision de détection est remarquablement mieux [176].

Plus tard dans [174,74], les auteurs proposent une technique d'extraction des caractéristiques de deux domaines de transformation (transformée en cosinus discrète et Transformé en Contourlet) séparément. Ces caractéristiques sont étudiées individuellement et manière combinatoire, des expériences ont montré que pour l'image JPEG le domaine DCT est un meilleur choix pour l'extraction de caractéristiques, et que l'extraction de caractéristiques dans plus d'une zone de traitement améliore le rendement [144].

Alors, la stéganalyse universelle à pour but de construire un détecteur pour découvrir les images stego produites par l'algorithme de stéganographie inconnu, au lieu de viser une méthode particulière de stéganographie [31].

5.3. L'approche proposée

L'ajout d'un message à une image de couverture n'affecte pas l'aspect visuel de l'image mais peut affecter certaines statistiques. Les caractéristiques requises pour la tâche de la stéganalyse devraient être capable d'attraper ces statistiques qui sont créés pendant le processus de masquage de données [49].

L'idée générale de notre approche est d'identifier ce qui caractérise une image de couverture d'une autre image stéganographiée, pour pouvoir discriminer les deux classes *cover* et *stégo*. En pratique, cela se traduit par l'extraction de caractéristiques pertinentes séparant les deux classes (*cover* et *stégo*) en premier lieu, en suite par l'emploi d'un classifieur particulier pour les différentes phases d'apprentissage et de classification [19].

– *Une phase d'apprentissage* : pour laquelle nous disposons au préalable d'une large base de données (dans notre cas une base d'images), et tel que la classe de chacun de ses éléments est connue d'avance (classe *cover*, ou classe *stego*). Les images utilisées doivent être de la même dimension afin de respecter la loi de la racine carré. Lors de cette phase, nous procédons d'abord à l'extraction des caractéristiques de chacun des média composant la base d'images. Ensuite nous choisissons un classifieur donné et règle ses paramètres (Par exemple, le taux de fausses alarmes est fixé au minimum), pour discriminer le plus précisément possible les deux classes d'images, à partir des caractéristiques extraites.

À la fin de cette première phase, le détecteur est opérationnel, et peut alors être utilisé pour la classification.

– *Une phase de test* : qui consiste à tester la performance du détecteur avant son utilisation en situation réelle. Lors de cette phase, des nouvelles images sont fournies au détecteur, qui doit décider de la classe à laquelle chacune d'elles appartient.

De cette description, il ressort deux choix importants lors de la conception, le premier choix crucial étant les caractéristiques utilisées, qui doivent être pertinentes pour la discrimination des classes. Le deuxième choix est celui du classifieur qui doit être efficace lors de la classification.

Dans ce manuscrit, nous nous focalisons principalement sur l'extraction de caractéristiques dans les images JPEG. Nous présentons dans ce qui suit une méthodologie pour la sélection des caractéristiques et permettant une interprétation accrue aux résultats du classifieur. Cette méthodologie est testée sur six algorithmes de stéganographie différentes et pour quatre taux d'insertion différents, les résultats de classification sont ensuite interprétés grâce aux caractéristiques sélectionnées comme les plus pertinentes pour chaque algorithme de stéganographie [20]. Le modèle de stéganalyse universelle en image JPEG est illustrait à la figure 5.2.

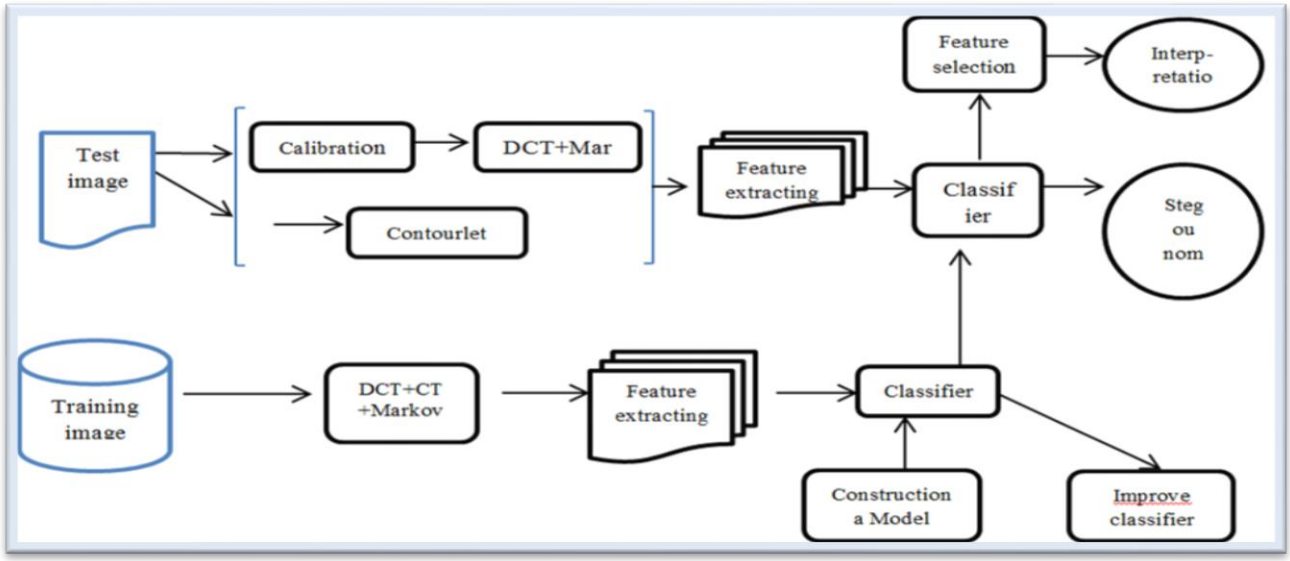


Figure 5.2. Le modèle de stéganalyse universelle d'image JPEG.

Dans ce travail, nous avons étudié le concept de stéganographie des données cachées dans des images numériques, et nous avons rétablir les statistiques d'image qui sont modifiés pendant le calibrage et sont généralement exploités par les attaques stéganalytique [91]. Tout d'abord nous devons extraire les caractéristiques diverses d'une image JPEG, d'après le calibrage. Globalement, on observe une augmentation particulièrement forte du nombre de caractéristiques utilisées pour la stéganalyse. Dans la deuxième section, nous citons les trois domaines d'extraction des caractéristiques (caractéristiques DCT, Markov, Contourlet). Ensuite, la détection est souvent présentée comme un problème de classification, nous avons utilisé une approche pour réduire la complexité du classifieur et réduire la dimensionnalité du problème par la sélection de variables.

5.3.1 Calibration

Le calibrage a été introduit en 2002 comme un nouveau concept pour attaquer l'algorithme F5. Depuis lors, il est devenu une partie essentielle d'un grand nombre de fonctionnalités basées sur la stéganalyse passive et ciblées en format JPEG ainsi que domaine spatial. Le calibrage a été également montré pour améliorer la précision de détection de la stéganalyse passive et consiste à estimer les caractéristiques d'image de couverture de l'image stego. Ainsi, l'effet net de calibrage est de diminuer les variations d'image et d'augmenter la sensibilité des fonctionnalités à intégrer [5,6].Il est utilisé aussi pour estimer les propriétés macroscopiques de l'image de couverture et stego.

Le processus de calibrage démarre avec une image JPEG, L'image stego J1 est décompressée dans le domaine spatial en utilisant DCT inverse (IDCT), cultivée par quatre pixels dans les deux sens, et compressée à nouveau avec la même matrice de quantification que l'image J1 stego [92] . La figure 5.3 montre une explication de ce processus.

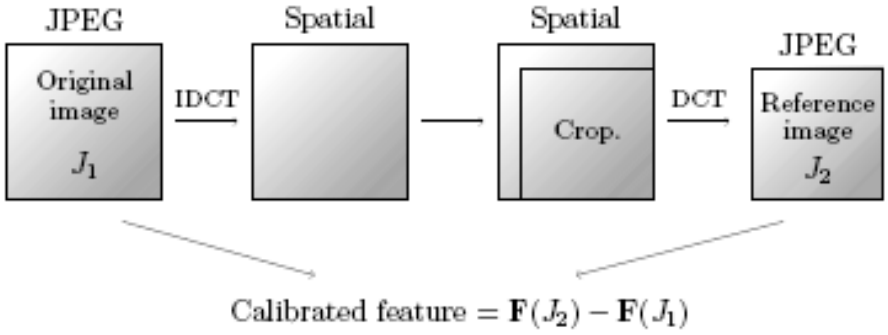


Figure 5.3 : le processus de calibration.

La fonction calibrée est obtenue comme la différence entre les caractéristiques calculées pour J1 et J2. Ce travail constitue également une évaluation plus détaillée de la performance des fonctions. Nous proposent un ensemble de fonctionnalités **F** nouvelles ont fusionné, dont la précision de détection est remarquablement meilleure.

5.3.2 Extraction des caractéristiques DCT

Les caractéristiques DCT ont été construites par l'utilisation du 23 **F** fonctionnelles qui produisent un scalaire, vecteur, une matrice. Chaque **F** fonctionnelle est évaluée pour l'image stego **J1** et sa version calibrée **J2**. La fonction *f* calibrée est obtenue comme la différence **F** (J1) - **F** (J2).

Supposons que le fichier traité est une image JPEG avec une taille M × N. DCT (i, j) désigne le coefficient DCT à emplacement (i, j) dans un bloc 8 × 8 DCT, où $1 \leq i \leq 8$ et $1 \leq j \leq 8$ Dans chaque bloc, dct (1,1) est appelé le DC coefficient, qui contient une fraction importante de l'énergie sur l'image. Donc, il ne considère que les 63 restants coefficients AC dans chaque bloc DCT [175], Le tableau 5.1 démontre la liste des types de caractéristiques individuelles et des symboles d''aide qui font référence dans le présent document.

$$g_{ij}^{d} = \sum_{k=1}^{B} \delta\left(d, d_k\left(i, j\right)\right) \qquad (5.1)$$

variation V :

$$v = \frac{\sum_{i,j=1}^{8}\sum_{k=1}^{|I_r|-1}\left|d_{I_r(k)}(i,j)-d_{I_r(k+1)}(i,j)\right| + \sum_{i,j=1}^{8}\sum_{k=1}^{|I_c|-1}\left|d_{I_c(k)}(i,j)-d_{I_c(k+1)}(i,j)\right|}{|I_r|+|I_c|} \quad (5.2)$$

Blockiness :

$$B_\alpha = \frac{\sum_{i=1}^{\lfloor (M-1)/8 \rfloor}\sum_{j=1}^{N}\left|x_{8i,j}-x_{8i+1,j}\right|^\alpha + \sum_{j=1}^{\lfloor (N-1)/8 \rfloor}\sum_{i=1}^{M}\left|x_{i,8j}-x_{i,8j+1}\right|^\alpha}{N\lfloor (M-1)/8 \rfloor + M\lfloor (N-1)/8 \rfloor} \quad (5.3)$$

la matrice de co-occurrence :

$$C_{st} = \frac{\sum_{k=1}^{|I_r-1|}\sum_{i,j=1}^{8}\delta\left(s,d_{I_r(k)}(i,j)\right)\delta\left(t,d_{I_r(k+1)}(i,j)\right) + \sum_{k=1}^{|I_c|-1}\sum_{i,j=1}^{8}\delta\left(s,d_{I_c(k)}(i,j)\right)\delta\left(t,d_{I_c(k+1)}(i,j)\right)}{|I_r|+|I_c|} \quad (5.4)$$

Functional/Feature name	Functional F
Global histogram	$H / \|H\|_{L_1}$
Individual histograms for 5DCT modes	$\frac{h^{21}}{\|h^{21}\|_{L_1}}, \frac{h^{31}}{\|h^{31}\|_{L_1}}, \frac{h^{12}}{\|h^{12}\|_{L_1}}, \frac{h^{22}}{\|h^{22}\|_{L_1}}, \frac{h^{13}}{\|h^{13}\|_{L_1}}$
Dual histograms for 11 DCT values (-5,…,5)	$\frac{g^{-5}}{\|g^{-5}\|_{L_1}}, \frac{g^{-4}}{\|g^{-4}\|_{L_1}}, \ldots, \frac{g^{4}}{\|g^{4}\|_{L_1}}, \frac{g^{5}}{\|g^{5}\|_{L}}$
Variation	v
L_1 and L_2 blokiness	B_1, B_2
co-occurences	N_{00}, N_{01}, N_{11}(features, not functionals)

Tab 5.1. L'extraction des caractéristiques DCT avec 193 caractéristiques.

Afin de pallier la perte d'information due à l'utilisation de la norme L1 et de garder la dimensionnalité des caractéristiques «raisonnable», nous avons remplacé la norme L1 par les différences suivantes.

$$H_l(J_1) - H_l(J_2), l \in \{-5, ..., 5\},$$

$$g_{ij}^d(J_1) - g_{ij}^d(J_2), (i,j) \in \{(2,1),(3,1),(4,1),(1,2),(2,2),(3,2),(1,3),(2,3),(1,4)\}.$$

$$C_{st}(J_1) - C_{st}(J_2), (s,t) \in [-2,+2] \times [-2,+2].$$

Après on remplace la norme L1 par les différences proposées, la dimension de l'ensemble des fonctionnalités (encore dénommée la fonction DCT) est augmenté jusqu'à 193 caractéristiques [74].

5.3.3 Les caractéristiques de Markov

La fonction de Markov proposé en [176] est définie comme un modèle de différences entre les valeurs absolues des coefficients DCT voisins et un processus de Markov. Le calcul des fonctions démarre en formant la matrice F (u, v) des valeurs absolues des coefficients DCT de l'image. Les coefficients DCT de F (u, v) sont disposés de la même manière que pixels de l'image en remplaçant chaque bloc 8 × 8 pixels avec le bloc correspondant de coefficients DCT [75].

Ensuite, quatre tableaux de différence sont calculés selon quatre directions: horizontale, verticale, diagonale, et mineure en diagonale (en outre noté $F_h(u,v), F_v(u,v), F_d(u,v)$ et $F_m(u,v)$ respectivement)

$$F_h(u,v) = F(u,v) - F(u+1,v)$$

$$F_v(u,v) = F(u,v) - F(u,v+1)$$

$$F_d(u,v) = F(u,v) - F(u+1,v+1)$$

$$F_m(u,v) = F(u+1,v) - F(u,v+1)$$

A partir de ces déférents tableaux, les quatre matrices de probabilité de transition (TPM) M_h, M_v, M_d, M_m sont construites [7]. Si une valeur dans la matrice de différence est en dehors de l'intervalle [-T, T], en le changent par -T ou T tous dépond qu'elle soit positive ou négative, la figure 5.4 illustre le principe étape de markov.

$$M_h(i,j)=\frac{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v}\delta\left(F_h(u,v)=i,F_h(u+1,v)=j\right)}{\sum_{u=1}^{S_u-1}\sum_{v=1}^{S_v}\delta\left(F_h(u,v)=i\right)} \quad (5.5)$$

$$M_v(i,j)=\frac{\sum_{u=1}^{S_u}\sum_{v=1}^{S_v-2}\delta\left(F_v(u,v)=i,F_v(u,v+1)=j\right)}{\sum_{u=1}^{S_u}\sum_{v=1}^{S_v-1}\delta\left(F_v(u,v)=i\right)} \quad (5.6)$$

$$M_d(i,j)=\frac{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v-2}\delta\left(F_d(u,v)=i,F_d(u+1,v+1)=j\right)}{\sum_{u=1}^{S_u-1}\sum_{v=1}^{S_v-1}\delta\left(F_d(u,v)=i\right)} \quad (5.7)$$

$$M_m=\frac{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v-2}\delta\left(F_m(u+1,v)=i,F_m(u,v+1)=j\right)}{\sum_{u=1}^{S_u-1}\sum_{v=1}^{S_v-1}\delta\left(F_m(u,v)=i\right)} \quad (5.8)$$

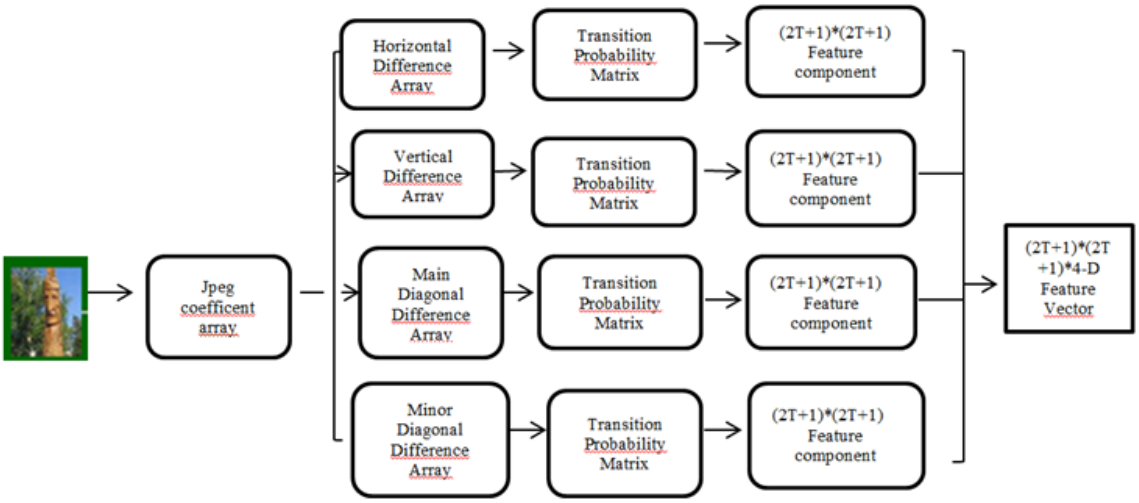


Figure 5.4 processus de markov .

5.3.4 Les caractéristiques de contourlet

La transformation en contourlet est une nouvelle extension à deux dimensions de l'ondelette utilisant un multi directionnelle et des bancs de filtres [14]. Pour l'extraction de caractéristiques dans le domaine de Contourlet, nous avons décomposé l'image en trois niveaux pyramidales et 2n directions où n = 0, 2, 4. La figure ci-dessous montre les niveaux de cette décomposition. Pour la phase de décomposition pyramidale laplacienne, la pyramide fait la décomposition en sous-bandes d'images, puis les bancs de filtres directionnels analysent chaque image. Il se compose de deux étapes majeures: la décomposition en sous-bandes et la transformation directionnelle.

À la première étape, nous avons utilisé la pyramide de Laplace (LP), et pour la deuxième étape, nous avons utilisé des bancs de filtres directionnels (DFB) [12]. La banque pyramidale

directionnelle de filtre (PDFB), proposé par Minh Do et Vetterli, qui surmonte l'approche basée sur les blocs de transformée en curvelet par un banc de filtres directionnels, appliqué sur toute l'échelle aussi connue comme transformer en contourlet (CT) [67]. La figure 5.4 montre une structure à double banque de filtre comprenant la pyramide de Laplace.

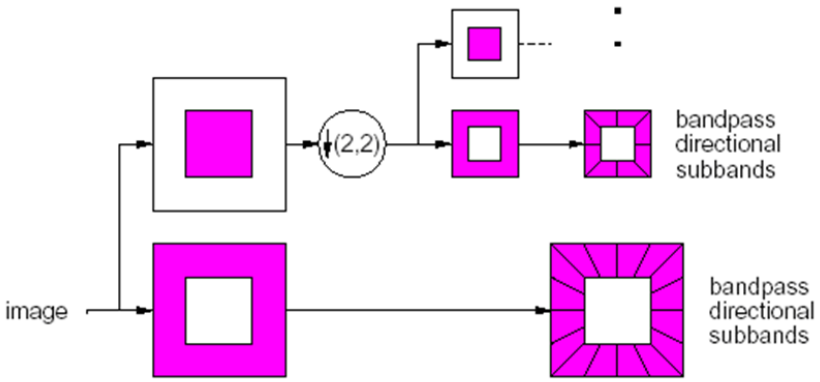


Figure 5.5 : transformé en contourlet .

Dans chaque échelle, on appliquant la décomposition pyramidale directionnelle comme banque de filtres sur toute le nombre de directions (16 nombre), et on ignore la sous-bande passe-bas [62]. Et le calcules des trois premiers moments FC normalisés pour chacune des 23 sous-bandes, ce qui donne un **69-D** vecteur de caractéristiques**.**

Dans une décomposition de 3 niveaux, le grand nombre de coefficients générés de cette décomposition (8 sous-bandes sont générés) nécessité de ne pas être impliqué dans l'étape de classification pour réduire le temps de calcul. Donc nous avons travaillés sur la réduction de dimension des fonctions en calculant les caractéristiques énergétiques suivantes de chaque sous-bande (eq 5.10 à 5.14), la Figure 5.5 donne la vue des trois niveaux de décomposition pour garder le simple chiffre.

E1– Mean

E2 –Standard deviation

E3 –Absolute Mean Energy

E4 –Energy

E5 –Skewness

E6 – Kurtosis

$$E1(s,k) = \mu(s,k) = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N} W_{s,k}(i,j) \tag{5.9}$$

$$E2(s,k) = \sigma(s,k) = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|W_{s,k}(i,j) - \mu_{s,k}\right|\right]^{\frac{1}{2}} \tag{5.10}$$

$$E3(s,k) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|W_{s,k}(i,j)\right| \tag{5.11}$$

$$E4(s,k) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|W_{s,k}(i,j)\right|^{2} \tag{5.12}$$

$$E5(s,k) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left(W_{s,k}(i,j) - \mu(s,k)\right)^{3}}{\sigma(s,k)^{3}} \tag{5.13}$$

$$E6(s,k) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left(W_{s,k}(i,j) - \mu(s,k)\right)^{4}}{\sigma(s,k)^{3}} \tag{5.14}$$

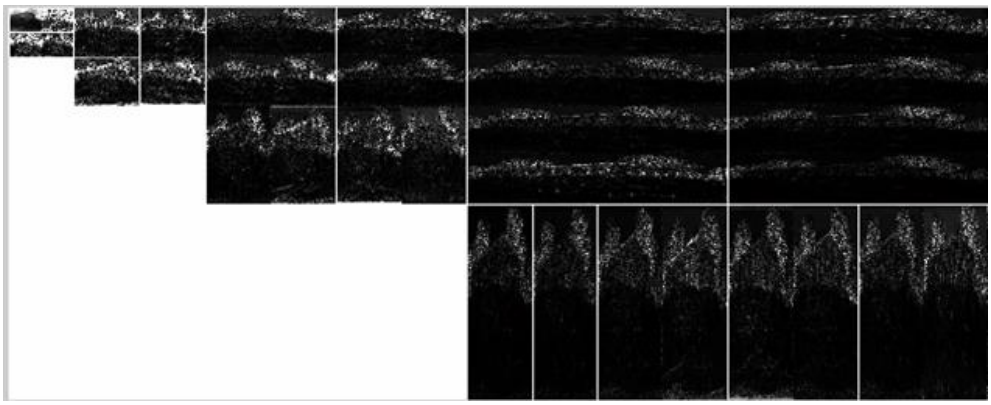


Figure 5.6 : La décomposition Sous-bande à trois niveaux de résolution.

5.3.5. Les caractéristiques fusionnées

Une combinaison directe des trois ensembles de caractéristiques produira un 517-D de vecteur de caractéristiques. Pour réduire la dimensionnalité de résultat, nous avons utilisé la moyenne $\bar{M} = \left(M_h^{(c)} + M_v^{(c)} + M_v^{(c)} + M_d^{(c)} + M_m^{(c)} \right) / 4$ des quatre matrices calibrées, ce vecteur de caractéristiques à 81D. Puis nous observons que les caractéristiques de performance moyenne $\bar{M}$ produit est très similaire à celle de leur version intégrale $M_h^{(c)}, M_v^{(c)}, M_d^{(c)}, M_m^{(c)}$, après la fusion de 193 caractéristiques de DCT étendue avec les 81 caractéristiques en moyenne calibrée et les 174 caractéristiques de Markov, et les 69 caractéristiques de Contourlet , la dimension de résultat des caractéristiques fusionné sera 517 D. Ce jeu de fonctionnalité a été choisi parce qu'il est très populaire et elle fournit des résultats fiables pour la stéganalyses[67].

5.4 La Classification

Un classifieur entraîné au préalable, permet d'obtenir une probabilité concernant la possibilité que l'image été modifiée par la stéganographie ou non. Un des problèmes étudiés dans cette thèse liés directement à ce concept de stéganalyseur universel et au nombre de caractéristiques extraites de l'image. En effet, le nombre de caractéristiques utilisé définit directement la dimension de l'espace dans lequel le classifieur utilisé doit établir une frontière entre images originales et stéganographie .

Les Méthodes basé sur le noyau, tels que KPCA [11] et KDA [155], s'étendent à des méthodes linéaires existant pour les services non linéaires. En particulier, Machine à vecteurs de support (SVM) [36] avec le noyau a été appliquée avec succès dans des problèmes polyvalents. Ces méthodes basées sur le noyau, cependant, nécessitent un assez grand coût de calcul puisqu'il faut calculer les fonctions du noyau pour tous les vecteurs, appelés vecteurs de soutien SV, c'est à due soutenir le classifieur (ou la projection).

Le nombre de SV augmente linéairement dans SVM pour un grand problème d'échelle et la classification devient beaucoup plus de temps, notamment pour des problèmes de grande envergure. En pratique, il est souhaitable de réduire le coût de calcul nécessaire à la classification par des méthodes basées sur le noyau [15].

Certains travaux antérieurs ont montré qu'un classifieur construit par un ensemble réduit de SV peut encore conserver la haute performance. Burges et al. [20] ont approché sur le classifieur SVM en utilisant un petit nombre de vecteurs en termes de distance euclidienne, mais ils ont coûteux en calcul pour trouver un tel ensemble réduit. Les travaux de [12, 166]

sont liés étroitement à notre méthode qui portent sur la dépendance linéaire de SV. On peut effectuer un tel procédé permettant de réduire le coût de calcul nécessaire dans le classifieur en tant que post-traitement, pas de prétraitement. Dans [108, 113], les candidats de SV sont supposés être prédéfinis, et c'est une tâche difficile et exhaustive pour sélectionner les candidats qui augmentent la performance.

Dans la sélection de classificateur, Farid [115] utilise SVM pour distinguer les images de couverture et images stego, Fridrich [76] utilise également les SVM à partir d'images JPEG stéganalysé. Comment choisir une fonction de noyau approprié pour une classification de motif donné est une question ouverte et difficile. Si Wu et al. Ont proposé l'idée qui améliore le SVM en modifiant la fonction du noyau, dans [156] ils ont utilisé une méthode de transformation qui a pour objet de modifier la fonction du noyau.

Dans les algorithmes de stéganalyse actuels basés-SVM : la précision a besoin de beaucoup de calcul pour sélectionner un paramètre approprié [157,29], afin de parvenir à un équilibre entre le temps et la performance de classification. Cette thèse propose un nouvel algorithme universel pour la stéganalyse d'image JPEG, dont on utilise les SVS pour réduire le coût de calcul de la classification. Cette méthode est basée sur l'algèbre linéaire d'un noyau de la matrice Gram de SV et prend les SVS redondants à faible coût de calcul tout en gardant la haute performance du classifieur. Pour la taille de SVM, nous utilisons le critère d'évaluation du classificateur qui est construit par l'ensemble réduit de SV [15].

5.4.1Redundant Support Vector

Dans les méthodes basées sur le noyau, nous considérons l'espace des caractéristiques via la fonction non linéaire φ de l'espace d'entrée. Le classifieur correspond à un hyperplan, le vecteur normal est représenté par une combinaison linéaire des SV_s:

$$y = w^T \phi(x) + b ,$$

$$w = \sum_{i=1}^{n_s} \alpha_i \phi(s_i) \qquad (5.15)$$

Où y est la valeur de sortie du classifieur, x est le vecteur d'entrée, w et b sont le vecteur normal et le biais de l'hyperplan, respectivement, n_s est le nombre de vecteurs de support (S_i), et α_i est le coefficient linéaire de vecteurs de soutien φ (S_i). Supposons qu'un vecteur de soutien est linéairement dépendant des autres:

$$\phi(s_i) = \sum_{j \neq i} \beta_j \phi(s_j) \tag{5.16}$$

L'approximation d'erreur quadratique de l'équation (5.16) est

$$\varepsilon = \left\| \phi(s_i) - \sum_{j \neq i} \beta_j \phi(s_j) \right\|^2 = k_{ii} - 2\beta^T k_{(i)} + \beta^T K_{(i)} \beta \tag{5.17}$$

Afin de résoudre efficacement, nous employons l'algèbre linéaire élémentaire comme suit. Tout d'abord, nous considérons la matrice inverse de la matrice régularisé de Gram K + λI:

$$H = (K + \lambda I)^{-1} = \begin{pmatrix} K_i + \lambda I & k_{(i)} \\ k_{(i)}^T & k_{ii} + \lambda \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} \breve{K}_{(i)}^{-1} + \dfrac{\breve{K}_{(i)}^{-1} k_{(i)} k_{(i)}^T \breve{K}_{(i)}^{-1}}{\breve{k}_{ii} - k_{(i)}^T \breve{K}_{(i)}^{-1} k_{(i)}} & \dfrac{-\breve{K}_{(i)}^{-1} k_{(i)}}{\breve{k}_{ii} - k_{(i)}^T \breve{K}_{(i)}^{-1} k_{(i)}} \\ \dfrac{-k_{(i)}^T \breve{K}_{(i)}^{-1}}{\breve{k}_{ii} - k_{(i)}^T \breve{K}_{(i)}^{-1} k_{(i)}} & \dfrac{1}{\breve{k}_{ii} - k_{(i)}^T \breve{K}_{(i)}^{-1} k_{(i)}} \end{pmatrix} \tag{5.18}$$

Où $\breve{K}_{(i)} = K_{(i)} + \lambda I$, $\breve{k}_{ii} = k_{ii} + \lambda$ et nous nous intéressons à l'échantillon de rang *i*-th en permutant l'ordre de **K**, la solution du problème des moindres carrés peut être simplement décrit comme

$$\boldsymbol{\beta} = -\frac{\boldsymbol{h}_{(i)}}{\boldsymbol{h}_{ii}}, \quad \breve{\in} = \in + \lambda \|\beta\|^2 = \frac{1}{h_{ii}} - \lambda = \breve{\in} \tag{5.19}$$

Où $\breve{\in} \approx \in$, en raison λ << 1. Ainsi, une fois que nous calculons la matrice inverse **H** de la matrice de Gram **K**, le problème des moindres carrés pour chaque SV peut être résolu à un coût de calcul assez faible. Cela nous permet d'éviter le traitement de beaucoup de temps. Nous pouvons réduire efficacement le coût en se concentrant sur la propriété de **H** à nouveau:

$$H_{(i)} = \breve{K}^{-1}_{(i)} + \frac{\breve{K}^{-1}_{(i)} k_{(i)} k^{T}_{(i)} \breve{K}^{-1}_{(i)}}{\breve{k}_{ii} - k^{T}_{(i)} \breve{K}^{-1}_{(i)} k_{(i)}} = \breve{K}^{-1}_{(i)} + \frac{h_{(i)} h^{T}_{(i)}}{h_{ii}}$$

$$\therefore \left(K_{(i)} + \lambda I \right)^{-1} = \breve{K}^{-1}_{(i)} = H_{(i)} - \frac{h_{(i)} h^{T}_{(i)}}{h_{ii}} \qquad (5.20)$$

La matrice inverse de la matrice de Gram réduite ($\boldsymbol{K}_{(i)} + \lambda \boldsymbol{I})^{-1}$, dans laquelle l'item SV est éliminé, peut être facilement calculée (mise à jour) à partir de celle de la matrice inverse originale **H**.

Au total, le temps consommé est un seul calcul de la matrice inverse de celle de Gram comme dans l'équation (5.20). L'identification des SV_s redondants et le calcul des coefficients linéaires dans l'équation. (5.16) sont ensuite effectuées à petit coût de calcul supplémentaire. L'algorithme proposé pour réduire efficacement SV est décrit dans l'algorithm Efficient Support Vector Reduction :

Algorithm Efficient Support Vector Reduction

Require : Kernel gram matrix of SVs K
$\in \Re^{n_s \times n_s} \left(k_{ij} = k\left(s_i, s_j\right), 1 \le i, j \le n_s \right)$
Require : Linear coefficients for SVs $\alpha \in \Re^{n_s}$
1: $H \leftarrow \left(K + \lambda \mathrm{I}\right)^{-1}, \hat{\alpha} \leftarrow \alpha$
2: **repeat**
2.1 : $i^* \leftarrow \arg\max_i h_{ii}$ /* redundant SV */
2.2 : $\hat{\alpha} \leftarrow \hat{\alpha}_{(i^*)} - \hat{\alpha}_{i^*} \frac{h_{(i^*)}}{h_{i^* i^*}}$
2.3 : $\mathrm{H} \leftarrow \mathrm{H}_{(i^*)} - \frac{h_{(i^*)} h^{\mathrm{T}}_{(i^*)}}{h_{i^* i^*}}$
3 : **Until** Stopping criterion is satisfied

Critère d'arrêt

Un critère d'arrêt pour la réduction séquentielle est nécessaire. Nous nous concentrons sur l'augmentation des coûts (perte) des valeurs pour l'ensemble de données de formation. Dans

SVM, Hinge losses sont employées:

$$HingeLoss = \frac{1}{N}\sum_{i=1}^{N}\max\left(0,1-g_i y_i\right) \qquad (5.21)$$

où N est le nombre d'échantillons d'apprentissage, $g_i \in \{+1, -1\}$ est la véritable étiquette et y_i est la valeur de sortie du classifieur pour l'échantillon de i- ième rang. On définit le critère que l'augmentation de Hinge loss du classificateur d'origine. La réduction de SV est arrêté lorsque la valeur du critère dépasse un certain seuil τ. Notez qu'un critère pourrait être pratiquement déterminée que l'erreur de reconnaissance calculée en utilisant un ensemble de données de validation, comme la procédure de validation croisée.

5.4.2 Évaluation des performances

Tous les stéganalyseur ont été mis en œuvre comme les classificateurs binaires réalisés en utilisant une machine à vecteurs de support souple avec un noyau gaussien. Les paramètres ont été optimisés par une méthode de gradient [129] sur l'ensemble de la formation. La base d'image a été divisée au hasard en deux parties; l'une a été utilisée pour formation et l'autre pour le test. La performance de steganalyzer est évaluée en utilisant la probabilité minimale de classification erronée par P_E $\qquad P_E = \min\frac{1}{2}\left(P_{FA} + P_{MD}\right) \qquad (5.22)$

où PFA est la probabilité de fausses alarmes, PMD est la probabilité de détections manquées.

5.5 Conclusion

Nous avons proposé une méthode efficace pour réduire le coût de calcul de la classification basée sur le noyau, une fois que nous calculons les vecteurs noyau de la matrice inverse de Gram , les vecteurs de soutien SV redondant sont séquentiellement identifiés et éliminés à très faible coût de calcul, tout en gardant la haute performance du classificateur. Pour arrêter la réduction séquentielle, nous appliquons le critère fondé sur l'augmentation du Hinge loss calculés pour les échantillons de formation. Dans la plupart des comparaisons avec SVM classique, notre approche a une meilleure efficacité qui sera détaillé dans le chapitre qui suit.

Chapitre 6

Expérimentations et évaluation

6.1 Introduction

L'idée principale derrière cette augmentation était celle de « stéganalyseur universel ». En effet, les premiers modèles de stéganalyse permettaient d'identifier une seule méthode de stéganalyse à la fois, il fallait donc d'utiliser chacun de ces modèles séparément sur une image à analyser, afin de vérifier la possible présence d'une information. L'aspect universel des nouvelles approches souhaite détecter l'ensemble des méthodes de stéganographie, et de ne pouvoir pas dire quelle méthode de stéganographie a été utilisée,mais bien d'identifier l'image comme suspecte ou non(même si la recherche de la méthode de stéganographie utilisée est une suite logique à ce concept de stéganalyseur universel).

L'extraction des caractéristiques permettent en effet d'améliorer les performances globales de stéganalyse, puis l'enembles extraient sont utilisés pour entraîner un classificateur. Ce dernier est entraîné au préalable,et permet d'obtenir une probabilité concernant la possibilité que l'image ait été modifiée par stéganographie ou non. Un des problèmes étudié dans cette thése est lié directement à ce concept de stéganalyseur universel et au nombre de caractéristiques extraites de l'image.

En effet, le nombre de caractéristiques utilisé définit directement la dimension de l'espace dans lequel le classifieur utilisé doit établir une frontière entre images originales et stéganographies .

La stéganalyse proposée a été mise en œuvre en utilisant MATLAB 7.6.0 avec des scripts Matlab voir (Annexe B), l'UCID [20, 12]la base d'image voir (Annexe B). contenant 1382 images naturelles de la résolution 512 * 384, cette base de données a été divisée en deux parties: un ensemble de formation ayant 854 images, et le reste 484 images de tests, nous avons utilisé les algorithmes Jsteg, F5 [61], JPHide, outguess [133], MB1, MB2 pour générer un message de 6 groupes d'images stego, incorporé aléatoirement compris entre 20% et 90% de capacité d'intégration.

Pour évaluer la méthode proposée dans cette thése, nous avons fait quelques expériences en comparant entre notre approche (les caratéristiques (DCT+markov+CT)) et les caractéristiques basées ondelette (WBS) et les caractéristiques basé contourlette (CBS). Nous remarquons que notre méthode proposée donne un meilleur résultat par rapport à d'autre méthode comme illustre la fiqure 6.1.

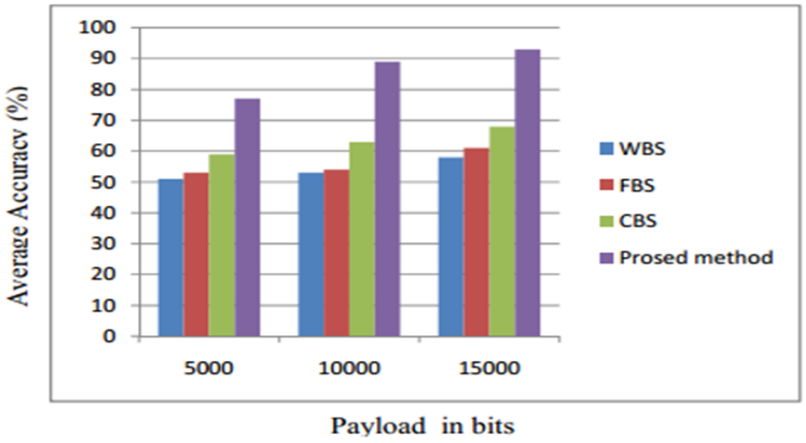


Figure 6.1 : Comparaison entre la méhode proposée et WBS,FBS, CBS.

Afin d'assurer l'exactitude des expériences , Le tableau ci_dessous montre la comparaison entre le résultat des expériences de stéganalyse base SVM [36] et notre méthode, la précision de détection des images de couverture et images stego ont été répertoriées dans le tableaux 6.1.

Stego Algorithm	Classic SVM-Based			Our Method		
	Stego	cover	nSV	Stego	Cover	nSV
F5	9.68%	100%	48	99.6%	99.8%	9
JPHide	93.8%	97.8%	137	97.8%	100%	91
Jsteg	98.4%	100%	36	100%	99.6%	11
OutGuess	99.4%	100%	10	99.4%	100%	5
MB1	80.4%	87.4%	257	93.2%	84.6%	39
MB2	93.6%	40.6%	342	90.2%	80.4 %	116

Tab 6.1. Précision de détection et le nombre de SV.

Pour MB1 et MB2, la précision de détection du stéganalyse basé SVM n'est pas souhaitable, avec grand nombre du SV même si un meilleur résultat peut être atteint après avoir sélectionné un paramètre approprié, il ne peut pas répondre aux questions des applications pratiques. La sélection du caractéristiques nécessitant un temps considérable et la performance de classification connaitra une grande amélioration, En remarque, la diminution du nombre de SV pour notre classifieur peut accélérer la classification.

Cependant, comment la fonction ou la stratégie d'apprentissage affecte la distribution d'erreur ou de son paramètre? est encore inconnue .Comme nous pouvons le voir dans la figure 6.2 et la figure 6.3, la distribution d'erreur de méthode sont symétriques à zéro, et pour notre procédé, ils sont plus denses que celle de toute autre méthode. Il est remarquable que les distributions d'erreur de F5 ont une grande variance. Ces propriétés sont en accord avec les résultats expérimentaux.

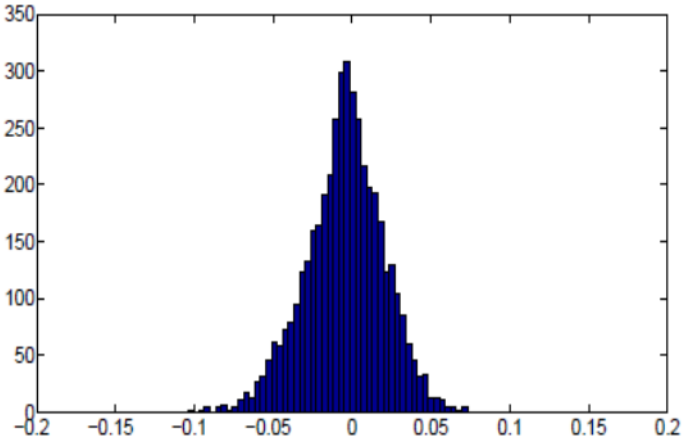


Figure 6.2. La distribution d'erreur de caratéristiques fusioné + stimulant sur F5.

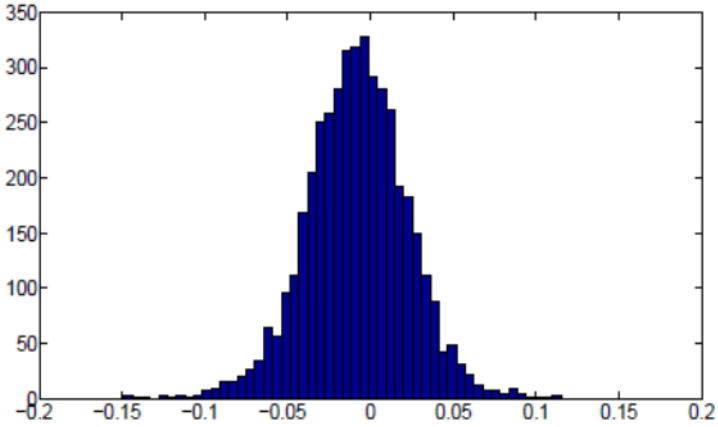


Figure 6.3 .La distributuion d'erreur des caratéristiques + SVR sur F5.

Nous avons calculé des statistiques qui présentent une déviation lorsque les images sont stéganographiées par des algorithmes tels-que steghide, Mbsteg et JPHide.

D'autre part, nous avons mesurées indépendamment les coordonnées du vecteur statistique avec des algorithmes de stéganographie, elles ne sont pas toutes impactées de la même manière par chaque algorithme. Les lois de probabilité dépendent de l'algorithme qui a produit ces stégo média. Par contre, les propriétés essentielles doivent être conservées par changement de base d'images. Les figures 6.4, 6.5 et 6.6 mettent en évidence les lois de probabilité suivies par les coordonnées de V.

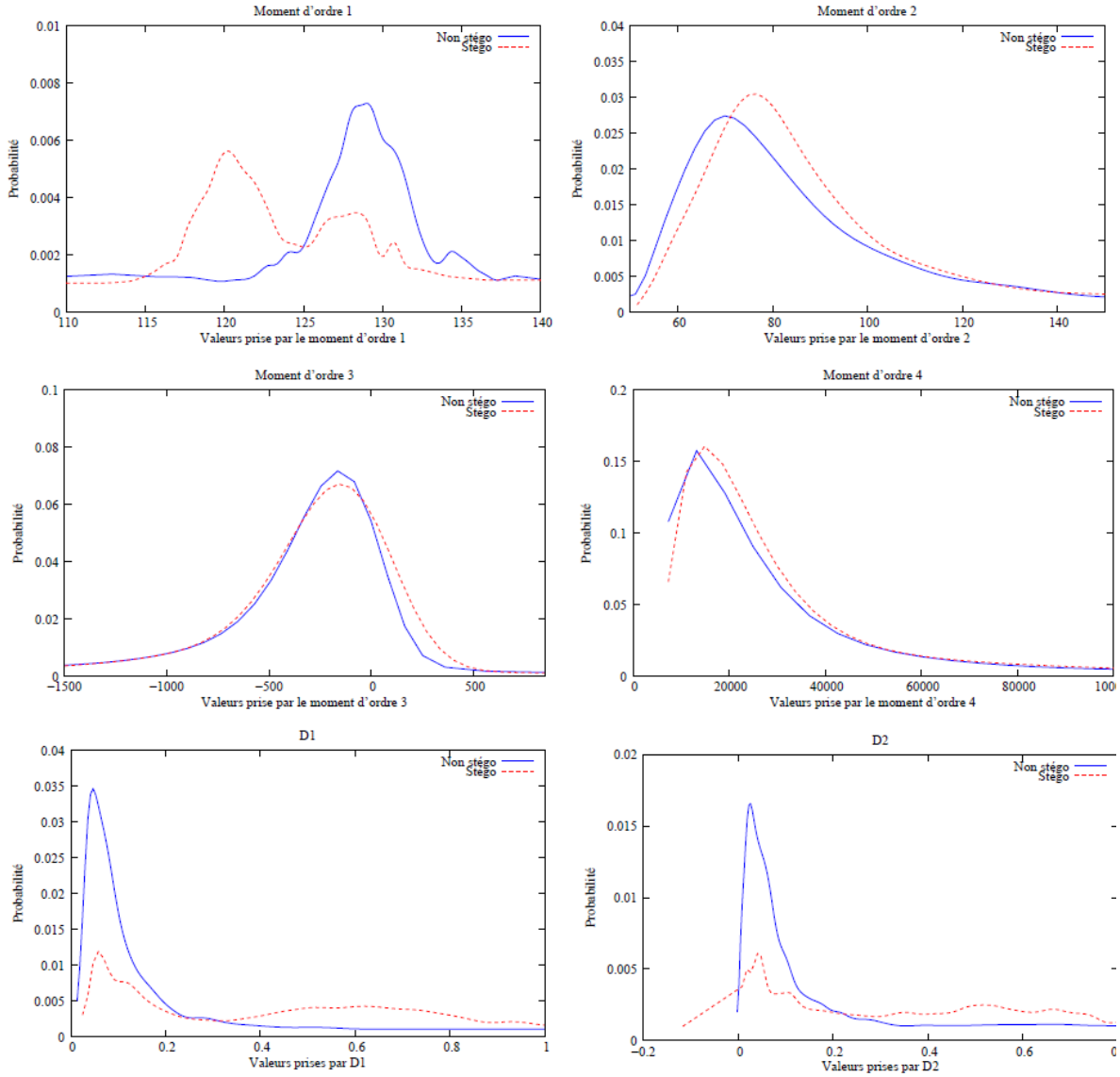


Figure 6.4 Attribution des coordonnées de V pour *Steghide*

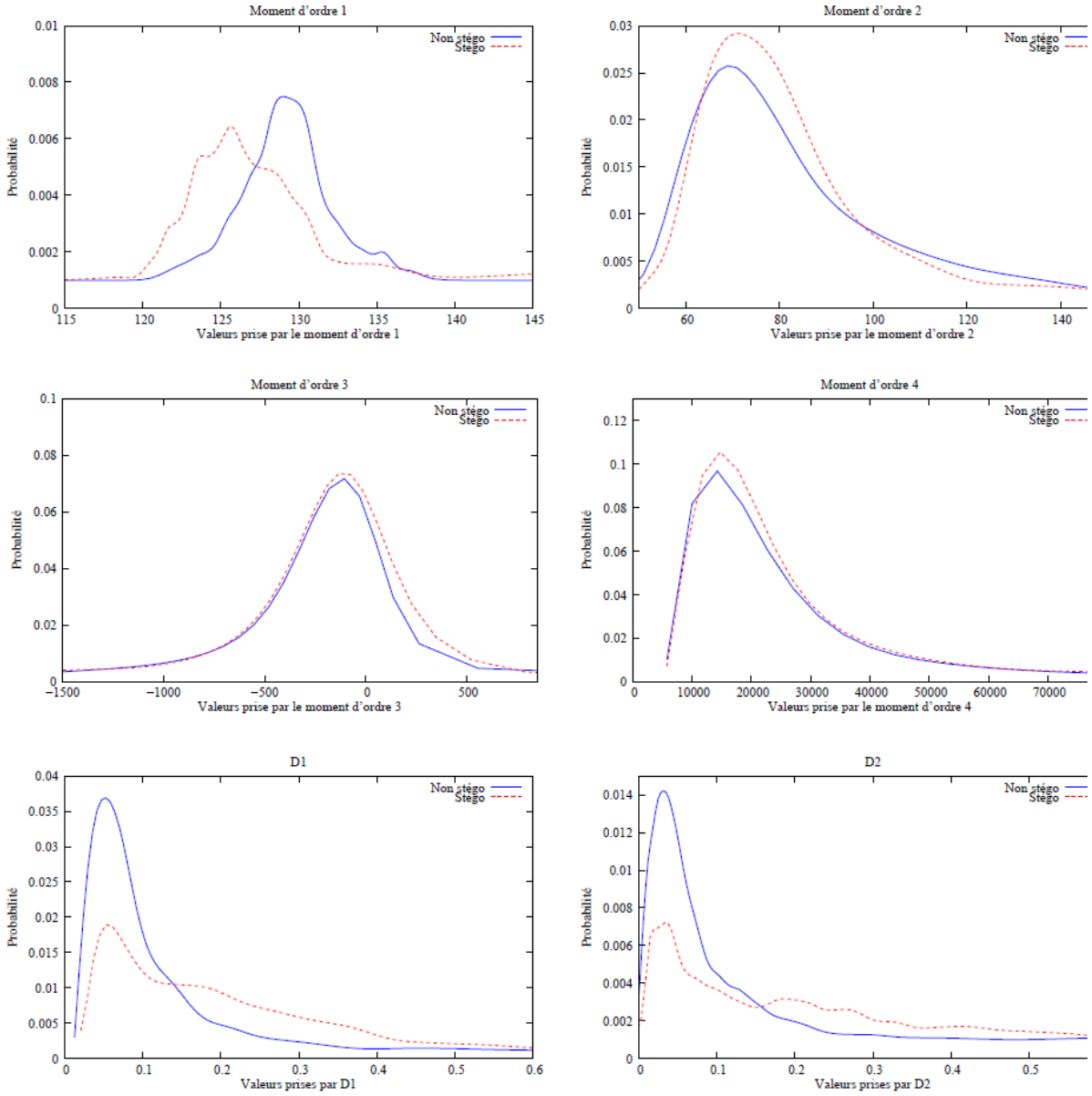


Figure 6.5 : Attribution des coordonnées de V pour *Mbsteg*.

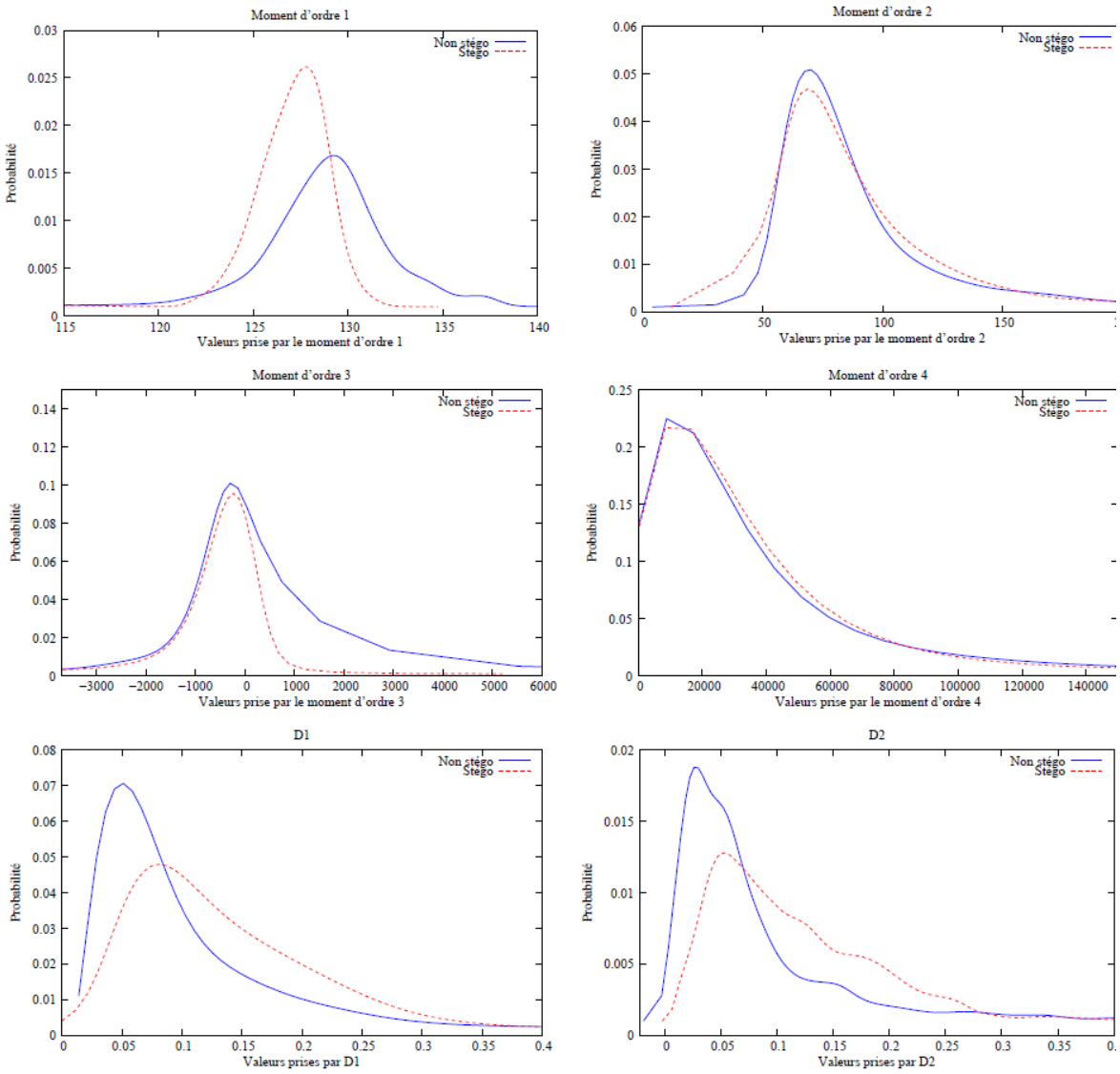


Figure 6.6 : Attribution des coordonnées de V pour *JPHide* .

Pour tester les performances de notre technique, nous avons comparé avec 800 images choisies au hasard, y compris 645 stego images de couverture, pour un taux d'insertion 10^{-6} à 10^{-1}. Ces résultats sont résumés dans la figure 6.7.

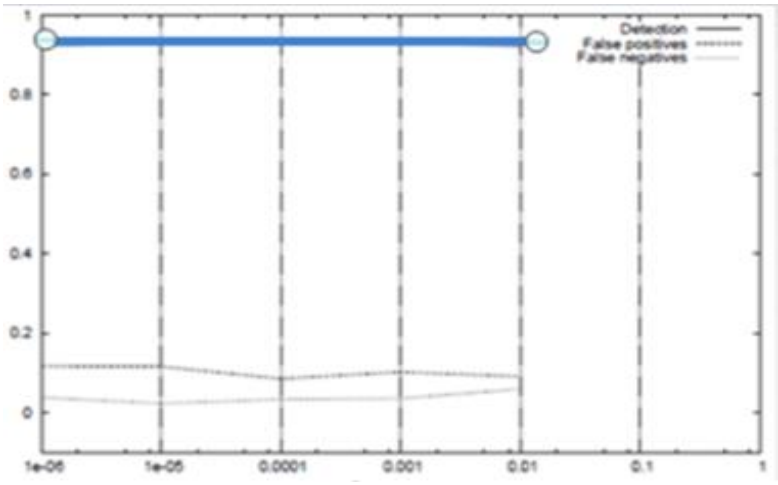
Nous avons observé tout d'abord, que la méthode semble être très efficace, en particulier lorsque le taux d'insertion est faible. D'autre part, le taux de détection apparaît comme constant et indépendant .Plus précisément, nous avons observé ce qui suit :

-le taux de détection pour Steghide est 95%, le taux d'erreur de faux positifs est 20% et le taux d'erreur de faux négatifs est 5%.

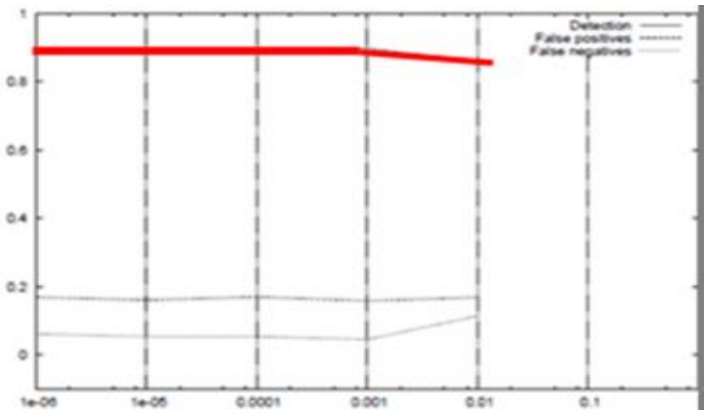
_le taux de détection de Mbsteg est 89,5%, le taux d'erreur de faux positifs est 18% et le taux d'erreur de faux négatifs est 9,6%.

_le taux de détection pour JPHide est 95,7%, le taux d'erreur de faux positifs est 4% et le taux d'erreur de faux négatifs est 1%.

Pourcentage de détection pour Steghide



Pourcentage de détection pour Mbsteg



Pourcentage de détection pour JPHide

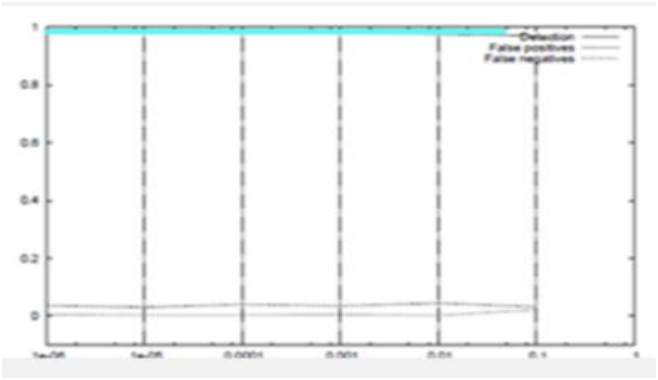


Figure 6.7 Les pourcentages de détection pour Steghide, Mbsteg et JPHide.

6.2 La Sélection des caractéristiques

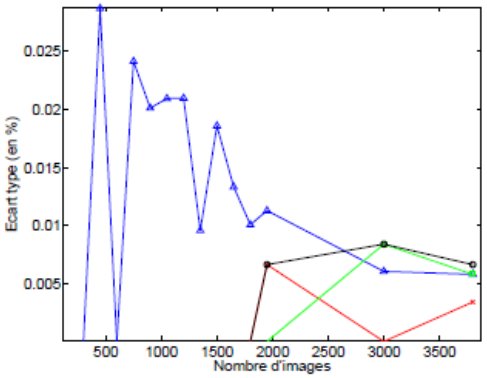
La Figure 6.8 illustre les résultats sur le changement de la valeur de l'écart type en fonction du nombre d'images utilisé. Les tracés ne vont pas au-delà de 2800 points pour deux raisons : les temps de calculs pour des ensembles plus importants, et d'autre part, l'écart type des résultats est déjà relativement faible. Et la valeur d'écart type dépend du nombre d'images utilisé, et on voit les tracés que lorsque on utilise les 2800 images, il est toujours inférieur à 1% de la meilleure valeur de classification. Les résultats est considérés fiables pour un nombre précis d'ensemble d'images.

La Figure 6.8 permet de voir que l'écart type des résultats diminue avec l'augmentation du nombre d'images, pour les algorithmes JPHS [103], MBSteg[126], OutGuess et StegHide [164] : la valeur décroit et reste au final entre 0 et 2% pour une meilleure performance de classification, avec un nombre d'images de 2800. On peut remarquer qu'il ya une relation entre le taux d'insertion et la valeur d'écart type, puisque ce dernier augmente lorsque le taux d'insertion diminue.

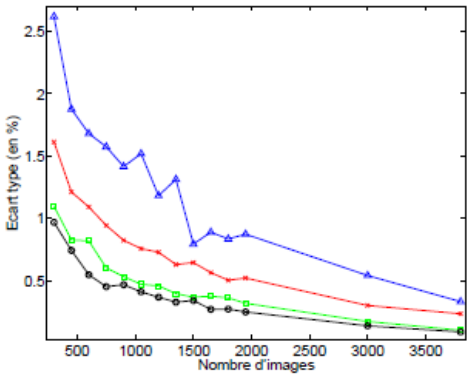
Avec un nombre très faible d'images, l'écart type des résultats diminue rapidement avec le nombre d'images, elle reste assez importante en dessous de 2000 images.

En effet, pour les quatre algorithmes mentionnés précédemment, il y a un écart important entre les taux 5 %, 10% et 20%. Ceci est un effet des performances des algorithmes de stéganographie : un faible taux d'insertion est plus difficile à détecter l'algorithme. La Figure 6.9 illustre la valeur d'écart type pour les cas de JPHS, StegHide et OutGuess.

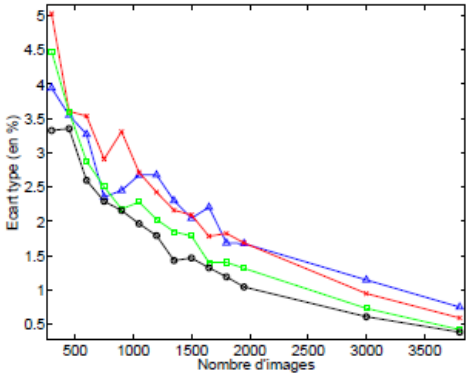
Pour la figure 6.8, Les algorithmes MM3 et F5 peut s'expliquer par les très bons résultats du classifieur, proches de 100% de bonne classification, ce qui rend l'écart type des résultats très faible : entre 2% et 0 pour MM3 et entre 1% et 0 pour F5.



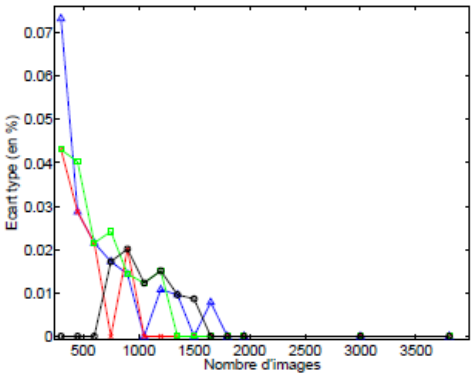
(a) F5



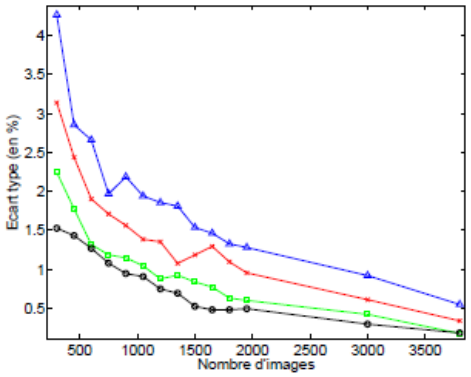
(b) JPHS



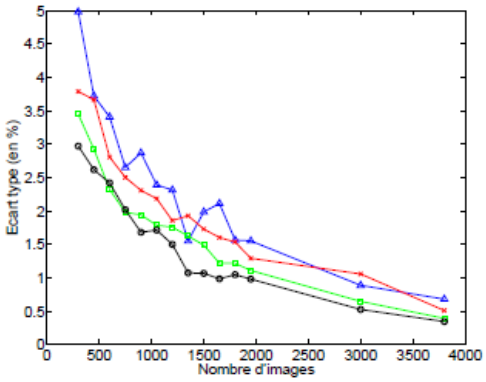
(c) MBSTEG



(d) MM3



(e) OutGuess



(f) StegHide

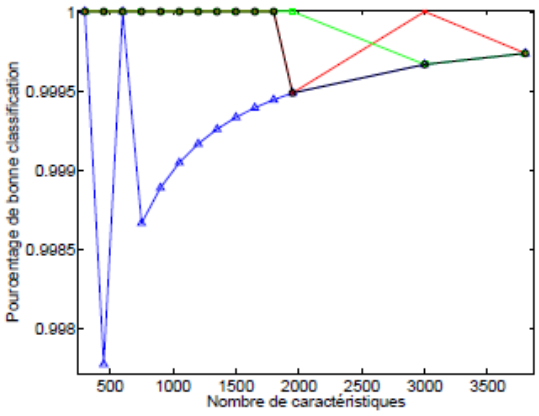
Figure 6.8 : Pourcentages de bonne classification pour les six algorithmes de stéganographie en fonction du nombre d'image (pour les quatre taux d'insertion utilisés) : cercles noirs (●) pour 20 %, carrés verts (■) pour 15 %, croix rouges (×) pour 10 % et triangles bleus (▲) pour 5 %.

À nouveau, le cas de MM3[127] est séparé des autres : quel que soit le taux d'insertion utilisé, la stéganalyse donne de très bons résultats. Avec seulement 10 caractéristiques pour ces deux cas, le pourcentage de bonne classification est 100 %, même pour un faible taux d'insertion .Les cinq autres algorithmes donnent des résultats différents :

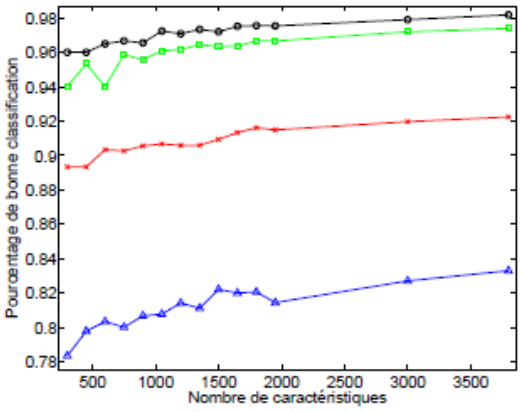
- JPHS obtenu (20 caractéristiques) pour tous les taux d'insertion ;
- OutGuess : 25 caractéristiques ;
- F5 : 20 caractéristiques ;
- Les performances pour l'algorithme StegHide est considérés comme maximales pour 60 caractéristiques à 20 % de taux d'insertion , même si pour des taux d'insertion de 5 et 10%.

Enfin, l'algorithme MBSteg: les performances de classification sont en effet à leur maximum dès 25 caractéristiques à 5 % ; pour les autres taux d'insertion représentent environ 60 caractéristiques pour arriver à des performances stables.

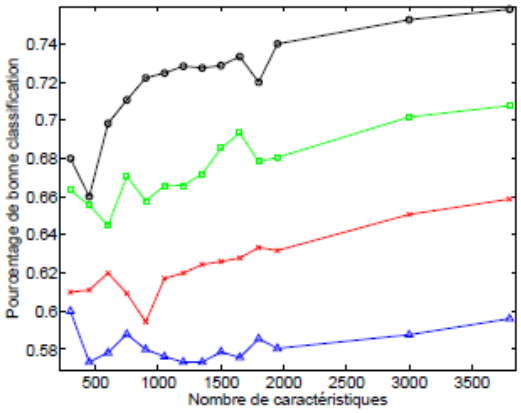
La Figure 6.9 trace le pourcentage de bonne classification en fonction du nombre de caractéristiques.



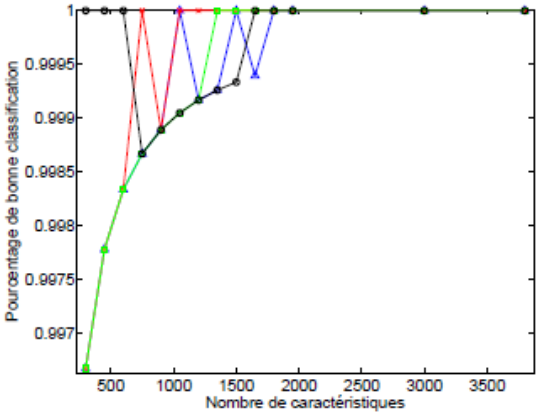
(a) F5



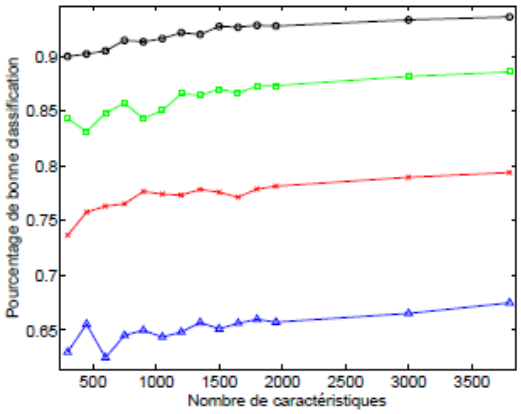
(b) JPHS



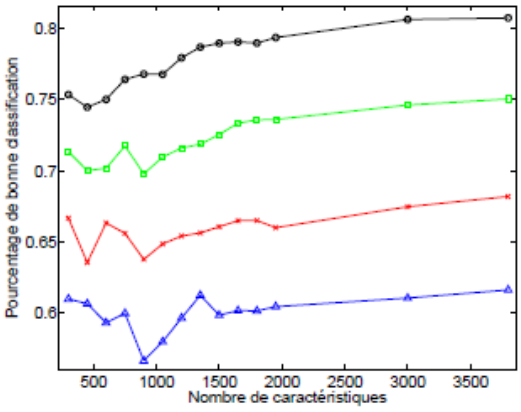
(c) MBSTEG



(d) MM3



(e) OutGuess



(f) StegHide

Figure 6.9. Pourcentages de bonne classification pour les six algorithmes de stéganographie en fonction du nombre de caractéristiques (pour les quatre taux d'insertion utilisés) : cercles noirs (O) pour 20 %, carrés verts ( ) pour 15 %, croix rouges (×) pour 10 % et triangles bleus (△) pour 5 %.

6.2.1 Performances de bonne classification avec le nombre réduits

Les ensembles évoqués précédemment, propres à chaque algorithme et taux d'insertion, sont comparés en termes de performances de bonne classification, en Table 6.2.

	5%	*#*	*10%*	*#*	*15%*	*#*	*20%*	*#*
F5	*100.0 (99.98)*	*10*	*100.0 (100)*	*10*	*100(100)*	*10*	*100(100)*	*10*
JPHS	*85.65(83.25)*	*20*	*92.15(93.20)*	*20*	*97.54(96.60)*	*20*	*98.25(97.50)*	*20*
MBSteg	*65.55(59.70)*	*25*	*66.55(66.20)*	*60*	*69.40(70.20)*	*60*	*74.85(76.25)*	*60*
MM3	*100.0 (99.95)*	*10*	*100 (100)*	*10*	*100.0(100.0)*	*10*	*100(100)*	*10*
OutGuess	*69.09 (67.30)*	*25*	*79.95(78.95)*	*25*	*88.36(86.90)*	*25*	*94.40(92.45)*	*25*
Steghide	*65.20 (62.10)*	*25*	*69.10(67.45)*	*25*	*66.45(74.00)*	*40*	*90.50(78.85)*	*40*

Tab 6.2 : Performances de classifieur pour l'ensemble complet de caractéristiques et les performances en utilisant l'ensemble réduit () ;

la sélection de variables permettant que les pourcentages obtenus avec les ensembles réduits soient potentiellement inférieurs aux valeurs maximales, au sein de l'intervalle de confiance pour 2800 images (écart type de 1%), les performances restent inferieur.

Même si des performances supérieures à celles présentées pour l'ensemble réduit choisi sont atteignables, seule la conservation de performances proches et équivalentes à la meilleure valeur, obtenue avec un nombre le plus réduit possible de caractéristiques.

Il apparait donc que les ensembles réduits donnent globalement les mêmes résultats que l'ensemble complet. Les cas de OutGuess et StegHide résultats moins bons pour les ensembles réduits que pour l'ensemble complet, même s'il faut noter que ce sont les deux algorithmes ayant le plus grand écart type dans les résultats.

6.2.2 L'ensemble final

L'idée est d'obtenir un ensemble commun, le plus petit possible, tout en conservant les mêmes performances pour tous les taux d'insertion. L'intersection est utilisée pour avoir le but de la sélection de caractéristiques: réduire autant que possible l'ensemble des caractéristiques utilisées, tout en conservant de bonnes performances.

Le nombre réduits présentés précédemment ont été utilisés pour constituer les intersections. Ainsi, les 10 caractéristiques obtenues pour F5 (pour chaque taux d'insertion), les 16 premières pour JPHS, et ainsi de suite, comme mentionné en Table 6.3.

Les résultats pour chaque algorithme et taux d'insertion sont présentés en Table 6.3, avec la taille de l'ensemble intersection.

	5%	10%	15%	20%	#
F5	100(99.95)	100(99.98)	100(100)	100(100)	8
JPHS	80.65(82.25)	90.15(90.57)	95.54(95.33)	96.25(95.62)	11
MBSteg	50.55(58.70)	66.55(60.70)	69.40(62.51)	74.85(64.60)	23
MM3	100(99.98)	100(99.95)	100(100)	100(100)	9
OutGuess	68.09(67.33)	80.95(78.35)	90.36(86.04)	94.40(92.50)	21
Steghide	65.05(60.40)	70.10(67.05)	76.45(72.09)	80.53(75.65)	15

Table 6.3 : Performances de classifieur pour l'ensemble des caractéristiques et l'ensemble intersection () ; la taille de l'ensemble intersection est également précisée.

On peut remarquer:

- Pour F5 et MM3, comme déjà observé, la stéganalyse est facile a détecté, même pour un taux d'insertion aussi faible que 5%. Les résultats utilises le nombre réduits ou intersection sont toujours aussi bons par rapport à l'ensemble complet.
- MBSteg présente un nombre relativement faible de caractéristiques est suffisant pour de faibles taux d'insertion, et pour les taux d'insertion plus importants nécessite environ 60 caractéristiques. L'ensemble intersection ne comporte que 23

caractéristiques, et ne contient pas les caractéristiques qui provoquent l'augmentation de performances.

- Les cas de JPHS, OutGuess et StegHide possèdent un comportement similaire, la performance de StegHide est moins bonne à celles obtenues avec l'ensemble complet.

Au vu des ensembles pour les différents taux d'insertion, ceux obtenus pour un taux de 5% sont très différents des autres. Les tracés de la Figure 6.8 permettent déjà de se douter de ce fait : l'évolution pour les taux 10, 15 et 20% sont relativement identiques, comparés avec le cas des 5%, au sein d'un même algorithme.

Il apparaît donc que ce type d'ensembles communs pour un algorithme donné ne doive être construit qu'à partir d'ensembles obtenus pour des taux d'insertion significatifs. Les faibles taux d'insertion tendent en effet à rendre le procédé de sélection de caractéristiques moins fiable et les interprétations difficiles sinon impossibles.

6.3 Analyse de caractéristiques sélectionnées

La sélection peut alors être considérée comme plus fiable et met le mieux en valeur les possibles faiblesses des algorithmes étudiés. L'analyse suivante est une interprétation possible des ensembles de caractéristiques obtenues. Les conclusions et les détails obtenus concernant les algorithmes de stéganographie au travers des caractéristiques pour identifier une partie des faiblesses dans le but de pouvoir détecter l'algorithme et rende la stéganalyse plus facile.
On peut remarquer qu'en raison de la méthode utilisée pour la sélection de caractéristiques, qui permettrait de tirer des conclusions concernant la méthode de stéganographie considérée, mais une bonne sélection néanmoins permettant une analyse pertinente.

Tout d'abord, les notations de l'ensemble de caractéristiques utilisé [11] sont données, pour l'ensemble initial de 23 caractéristiques, dans la Table 6.4 :

Fonctionelle/ Caractéristique	**Fonctionelle F**
Histogramme Global	H / $\|H\|$
Histogramme Individuel pour 5 Modes DCT	$h^{21} / \|h^{21}\|, h^{12} / \|h^{12}\|, h^{13} / \|h^{13}\|,$ $h^{22} / \|h^{22}\|, h^{31} / \|h^{31}\|$
Histogramme dual pour 11 valeurs DCT	$g^{-5} / \|g^{-5}\|, g^{-4} / \|g^{-4}\|, ..., g^{4} / \|g^{4}\|,$ $g^{5} / \|g^{5}\|$
Variation	V
Facteurs de bloc L_1 et L_2	B_1, B_2
Co-occurrence	N_{00}, N_{01}, N_{11}

Tab 6.4 : les 23 caractéristiques.

Cet ensemble de 23 caractéristiques a été étendu à l'ensemble utilisé de 517 caractéristiques, en supprimant la norme $L1$ et en conservant l'ensemble des valeurs des matrices et vecteurs. Les notations définitives des caractéristiques sont les suivantes:

- Histogramme global de 11 dimensions $\mathbf{H}(i), i = [[1,11]]$
- 5 Histogrammes de coefficients DCT basse fréquence (11 dimensions chacun) $\mathbf{h}^{21}(i) \ldots \mathbf{h}^{31}(i), i = [[1,11]]$
- 11 histogrammes duaux (9 dimensions chacun)

$$\mathbf{g}^{-5}(i) \ldots \mathbf{g}^{5}(i), i = [[1,9]]$$

- Variation (1 dimension) $\mathbf{V}$
- 2 facteurs de bloc de dimension 1 $\mathbf{B}_1, \mathbf{B}_2$
- Matrice de co-occurrence de dimension 5×5

$$\mathbf{C}_{i,j}, i = [[-2,2]], j = [[-2,2]]$$

Les tables d'intersections des ensembles réduits de caractéristiques sont présentées ci-après, avec une analyse pour chaque algorithme.

6.3.1. F5 et MM3

$h^{22}(3)$	$h^{21}(6)$	$h^{21}(7)$	$h^{12}(5)$
$h^{12}(9)$	$h^{22}(9)$	$C_{-2,-2}$	$C_{-1,+1}$

Tab 6.5 Ensemble commun de caractéristiques (8) pour F5 (pour 10,15 et 20% de taux d'insertion).

$h^{22}(6)$	$h^{21}(7)$	$h^{12}(5)$	$h^{12}(7)$	$g^{-5}(1)$
$C_{-2,-1}$	$C_{-1,+1}$	$C_{+0,+2}$	$C_{+1,+1}$	

Tab 6.6 Ensemble commun de caractéristiques (9) pour MM3 (Intersection des 14 premiers pour 10,15 et 20% de taux d'insertion).

Les cas F5 et MM3 sont à nouveau proches, ont basé notamment sur le codage matriciel, et possèdent une liste de caractéristiques sélectionnées assez proche. Les points faible à la stéganalyse de MM3 : que les histogrammes de coefficients DCT ne soient pas préservés. Les coefficients extrêmes (−2,+2) sont sélectionnés pour F5, tandis que des plus faibles (−1,+1) le sont pour MM3.

Conclusion

- Les coefficients DCT extrêmes pris en compte pour F5, et des valeur absolues plus faibles pour MM3.
- Un nombre faible de caractéristiques sélectionné (8) pour F5.

6.3.2 JPHS

Pour le cas de JPHS (Table 6.7), un nombre faible de caractéristiques a également été sélectionné (11). JPHS ne préserve pas la cohérence fréquentielle de la matrice de co-occurrence et les coefficients basses fréquences.

$h^{12}(7)$	$h^{12}(6)$	$h^{13}(6)$	$h^{22}(6)$	$h^{31}(7)$	$h^{12}(7)$
$C_{-2,-1}$	$C_{-1,+1}$	$C_{-1,+2}$	$C_{+0,+1}$	$C_{+0,+2}$	

Tab 6.7 Ensemble commun de caractéristiques (11) pour JPHS (Intersection des 16 premiers pour 10,15 et 20% de taux d'insertion).

Conclusion

- L'histogramme des coefficients DCT sont prise en compte.
- La cohérence fréquentielle n'est pas conservée
- Le nombre de caractéristiques retenues est faible.

6.3.3. OutGuess

Principalement des valeurs extrêmes d'histogrammes sont utilisées (–2, –1) pour OutGuess (Table 6.8). La valeur 0 d'histogramme a été pris en compte. Les valeurs de co-occurrence entre –2 et +2 sont également d'importance.

$h^{22}(5)$	$h^{22}(5)$	$h^{21}(7)$	$h^{12}(7)$	$h^{12}(5)$	$h^{13}(7)$	$h^{13}(4)$	$h^{13}(6)$	$h^{21}(4)$	$H(6)$
$h^{31}(4)$	$h^{31}(5)$	$h^{31}(6)$	$g^{-2}(1)$	$g^{-2}(2)$	$C_{+2,-1}$	$C_{-2,+1}$	$g^{2}(1)$	$C_{-1,+1}$	$C_{+0,-2}$

Tab 6.8 Ensemble commun de caractéristiques (20) pour OutGuess (Intersection des 25 premiers pour 10,15 et 20% de taux d'insertion).

Conclusion

- Les coefficients –2 et –1 sont clairement les points faibles d'OutGuess. Le nombre de caractéristiques utilisé (20) pour obtenir un résultat suffisant en classification, ce qui tend à montrer qu'OutGuess reste un algorithme relativement fiable (difficile à détecter par ce processus de stéganalyse avec un nombre faible de caractéristiques).

6.3.4. Steghide

$h^{21}(6)$	$h^{12}(7)$	$h^{12}(8)$	$h^{13}(4)$	$h^{13}(7)$	$h^{13}(6)$	$h^{21}(4)$	B_2
$h^{22}(6)$	$h^{31}(4)$	$h^{31}(6)$	$C_{+2,-1}$	$h^{31}(8)$	$g^{-2}(1)$	$h^{31}(4)$	

Tab 6.9 Ensemble commun de caractéristiques (15) pour StegHide (Intersection des 25 premiers pour 10% et 40 premiers pour 15 et 20% de taux d'insertion).

L'algorithme StegHide utilise les histogrammes avec des coefficients haute fréquence (31, 13, 22) et pour des valeurs faibles. La matrice de cooccurrence est utilisée pour grands écarts .

Conclusion

- L'utilisation de hautes fréquences avec de faibles valeurs.

Au vu du nombre de caractéristiques obtenues, StegHide apparait comme un algorithme difficile à détecter.

6.3.5 MBSteg

$H(4)$	$H(6)$	$H(8)$	$h^{22}(5)$	$h^{21}(6)$	$h^{21}(5)$	$h^{12}(4)$	$h^{12}(5)$
$h^{12}(6)$	$h^{12}(1)$	$h^{13}(3)$	$h^{13}(4)$	$h^{13}(5)$	$h^{13}(6)$	$h^{13}(7)$	$h^{13}(8)$
$h^{21}(3)$	$h^{21}(4)$	$h^{22}(6)$	$h^{22}(9)$	$h^{31}(3)$	$h^{31}(4)$	$h^{31}(6)$	$h^{31}(8)$
$h^{31}(10)$	$g^{-5}(1)$	$g^{-5}(4)$	$g^{-5}(1)$	$g^{-3}(1)$	$g^{-1}(1)$	$g^{-1}(2)$	$g^{-1}(4)$

$g^2(1)$	$g^4(3)$	$C_{-2,-2}$	$C_{-2,-1}$	$C_{-2,+0}$	$C_{-2,+1}$	$C_{-2,+2}$	$C_{+0,+0}$
$C_{+0,+2}$	$C_{+1,-2}$	$C_{+1,-1}$	$C_{+2,+0}$	$C_{+2,+2}$	B_2		

Tab 6.10 Ensemble commun de caractéristiques (46) pour MBSteg (Intersection des 70 premiers pour 10% ,15 et 20% de taux d'insertion).

MBSteg possède une liste de caractéristiques exceptionnelles et un nombre important de caractéristiques pour obtenir des bonnes performances. Ceci signifie que l'algorithme est particulièrement difficile à détecter. Les caractéristiques importantes sont par exemple celles d'histogramme global pour les valeurs 0, –2 et 2, MBSteg préserve les coefficients des histogrammes,

6.4 Conclusion

Les résultats expérimentaux montrent que notre méthode de stéganalyse est capable de détecter l'utilisation de outguess, F5 et JPHide et JPHS, MBSteg même si le taux d'insertion est très faible (10^{-6}).Les expériences indiquent qu'en général, la précision du test du SVR est un peu moins bonne que celle de la SVM standard. Bien que SVR garde des contraintes similaires à la forme primitive de SVM. Pour le temps de formation qui est la principale motivation, SVR sera plus rapide que SVM régulière sur les grands problèmes ou certains cas difficiles avec de nombreux vecteurs de soutien.

Grâce à la seconde étape de la méthodologie, le nombre de caractéristiques nécessaire à une bonne classification, peut être diminué. Cette étape possède trois avantages : (1) les performances est augmenté pour l'ensemble réduit; (2) les caractéristiques sélectionnées sont pertinentes pour le problème, et rendent une analyse possible ; (3) les faiblesses de l'algorithme de stéganographie considérée apparaissent plus clairement par cette sélection, et peuvent mener à des améliorations en sécurités.

- Le nombre de caractéristiques sélectionné conditionne les performances de l'algorithme, ce qui signifie qu'un nombre de caractéristiques sélectionné important tend à indiquer que l'algorithme est meilleur, en termes de sécurité. Une possible extension de cette idée est de

vouloir modifier autant de caractéristiques indépendantes que possible, afin de rendre la stéganalyse plus facile.

CONCLUSION

Conclusion et perspectives

Le travail présenté dans cette thèse s'inscrit dans le cadre de la stéganographie et plus précisément la détection des informations cachées dans les images JPEG.

Dans notre travail, nous avons proposé une méthode de stéganalyse passive pour la détection des messages dissimulés dans une image JPEQ, 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 différents domaine de transformation afin d'extraire un ensemble de caractéristiques pertinentes qui permettent de caractériser une image.

Dans ce contexte, nous proposons une nouvelle méthode de stéganalyse universel pour les images JPEG basées sur les fonctionnalités de transformation hybride (cosinus discret et transformer en contourlet). Ces caractéristiques sont étudiés individuellement et de façon combinatoire. On a intégré le calibrage dans le processus de calcul pour les améliorer et d'augmenter la sensibilité des fonctionnalités à intégrer. Une combinaison des deux domaines de transformation produira un vecteur de 517-dimension.

Ensuite, la détection est souvent présentée comme un problème de classification, nous avons utilisé une approche pour réduire la complexité du classifieur par des méthodes basées sur le noyau, tout en conservant la haute performance. En se servant de l'algèbre linéaire d'un noyau Gram des vecteurs de support (SVS) à faible coût de calcul.

Ces méthodes basées sur le noyau exigent un grand coût de calcul dans ce contexte, nous proposons une autre méthode pour tailler efficacement les SV utilisés dans ces dernières pour réduire le coût de calcul de la classification. Une fois que nous calculons la matrice inverse du noyau Gram de SV, les vecteurs de support redondants sont successivement identifiés et éliminés, tout en gardant la haute performance du classifieur. Dans la plupart des comparaisons avec SVM, notre approche donne une meilleure efficacité.

La deuxième étape de la méthodologie tente de réduire le nombre de caractéristiques nécessaires pour retirer une interprétation grâce à leur nombre réduit, par une sélection des caractéristiques les plus efficaces et utiles pour le problème de classification. Un algorithme SVR permet à cette sélection d'analyser les caractéristiques les plus sensibles pour un

algorithme donné, et met en évidence les possibles faiblesses du fonctionnement de l'algorithme de stéganographie.

Grâce à la seconde étape de la méthodologie, la diminution de nombre de caractéristiques est nécessaire à une bonne classification, Cette étape possède trois avantages : (1) les performances restent identiques si l'ensemble réduit a été correctement construit ; (2) les caractéristiques sélectionnées sont pertinentes pour le problème, et rendent une analyse possible ; (3) les faiblesses de l'algorithme de stéganographie considéré et peuvent mener à des améliorations éventuelles de l'algorithme, en vue d'une plus grande sécurité.

L'analyse des ensembles d'intersection tend à montrer que les algorithmes étudiés sont sensibles à des caractéristiques relativement similaires. Néanmoins, lorsque le taux d'insertion est de seulement 5%, ou pour les algorithmes les plus sûrs, certaines caractéristiques particulières apparaissent.

Le nombre de caractéristiques sélectionnés conditionne les performances de l'algorithme, ce qui signifie qu'un nombre de caractéristiques sélectionné important tend à indiquer que l'algorithme est meilleur, en termes de sécurité. Une possible extension de cette idée est de vouloir modifier autant de caractéristiques indépendantes que possible, afin de forcer le stéganalyseur à utiliser un nombre important de caractéristiques et rendre la stéganalyse plus difficile.

Le format JPEG ne paraît pas être un format adapté pour la stéganographie, notamment parce qu'il possède une structure très forte et agit dans trois domaines corrélés entre eux. Les taux de détection élevés et indépendants, en pratique, le taux stéganographique, nous poussent à essayer d'adapter l'étape de compression sans perte du format JPEG afin de concevoir des détecteurs adaptés à d'autres types de média.

Dans des recherches futures, nous allons essayer d'améliorer l'efficacité de notre classifieur, puis d'extraire des nouvelles fonctionnalités dans le domaine de fréquence comprimé et enfin le combinant avec des détecteurs dans d'autres domaines. Afin de rendre la méthode de stéganalyse proposée de type active, nous proposons de compléter le travail pour estimer la taille du message insérer. Cela peut être effectué par une étude de la répartition des motifs présents dans l'image.

Bibliographie

[1] A.D. Ker : Improved detection of LSB steganography in grayscale images. In J. Fridrich, éditeur : Proc. Information Hiding, 6th International Workshop, volume 3200 de Lecture Notes in Computer Science, pages 97–115, Toronto, Canada, mai 2004. Springer. ISBN : 3-540-24207-4.

[2] A. Kerckhoffs : La cryptographie militaire. Journal des Sciences Militaires, février 1883.

[3] A.Latham: Steganography: Jphide and Jpseek**,** http://linux01.gwdg.de/alatham /stego.html, 1999.

[4]A. Latham : Steganography : JPHIDE and JPSEEK, 1999. http ://linux01.gwdg.de/~alatham/stego.html.

[5]Analyse de programme stéganographique. http ://www.guillermito2.net/stegano/index.html, date 2015.

[6] Andrew D. Ker, (2005): Resampling and the Detection of LSB Matching in Bitmaps. Oxford University Computing Laboratory, England.

[7] Audio Steg. http ://www.snotmonkey.com/work/school/405/overview.html

[8] A. Westfeld : F5-a steganographic algorithm. In I.S. Moskowitz, éditeur : Proc. Information Hiding, 4th International Workshop, IHW 2001, volume 2137 de Lecture Notes in Computer Science, pages 289–302, Pittsburgh, PA, USA, avril 2001. Springer. ISBN : 3-540-42733-3.

[9] A. Westfeld : The steganographic algorithm F5, 1999. http ://www.rn.inf.tudresden. de/~westfeld/f5.html.

[10] A. Westfeld et A. Pfitzmann : Attacks on steganographic systems. In A. Pfitzmann, éditeur : Proc. Information Hiding, Third International Workshop, IH'99, volume 1768 de Lecture Notes in Computer Science, pages 61–76, Dresden, Germany, septembre 1999. Springer. ISBN : 3-540-67182-X.

[11] B. Sch¨olkopf and A. Smola, Eds : Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond , MIT Press, 2002.

[12]B. Sch¨olkopf, S. Mika, C.J.C. Burges, P. Knirsch, K-R. Muller, G. Raetsch, and A.J.Smola, : Input space vs feature space in kernel-based methods, IEEE Transactions on Neural Networks, vol. 10, pp. 1000–1017, 1999.

[13] B. Smith : An approach to graphs of linear forms (Unpublished work style), unpublished.

[14] B. Pfitzmann : Information hiding terminology. In Proceedings of the Workshop on Information Hiding, numéro 1174, pages 347–350, Cambridge, England, mai 1996. Springer Verlag.

[15] Bender, W., D. Gruhl, and N. Morimoto, *Techniques for data hiding*, IBM Systems Journal, vol. 35, no. 3/4, 1996, pp. 131-336.

[16] B.H. Astrowsky : steganography hidden images, a new challenge in the fight against child porn. UPDATE, 13(2), 2000.

[17] B.C. Nguyen, S.M. Yoon et H.-K. Lee : Multi bit plane image steganography. In Y. Q. Shi et B. Jeon, éditeurs : Proc. Digital Watermarking, 5th International Workshop, IWDW 2006, volume 4283 de Lecture Notes in Computer Science, pages 61–70, Jeju Island, Korea, novembre 2006. Springer.

[18] Bouguerne Imen, Tlili Yamina : Steganographic detection in image using the reduction of support vectors . In int. J. of Electronic Security and Digital Forensics 2015 - Vol. 7, No.1 pp. 20 – 29

[19] Bouguerne Imen, Tlili Yamina : A Steganalytic Based on DCT and Markov and Spatial Domain for JPEG Images. International Journal of computing vol 4 issue 4, April 2012

[20] Bouguerne imen, Merouani Hayet Farida: A Multi Resolution Decomposition for a Passive Steganalysis Based on a Multi Agent System . The International Conference on Image Processing, Computer Vision, and Pattern Recognition(IPCV'11).

[21] Bouguerne imen, Tlili Yamina : La stéganalyse adaptée aux images JPEG. 2 ème journées sur les problèmes inverses, 28_30 octobre 2013, Annaba.

[22] B. Roue, P. Bas, J. C. (2005). Influence des Vecteurs Caractéristiques en Stéganalyse par Séparateurs à Vastes Marges. In 20° Colloques sur le Traitement du Signal et des Images, pages 317–320. Cité page 68.

[23] Benoit Roue, Patrick Bas, Jean-Marc Chassery : Improving LSB Steganalysis Using Marginal and Joint Probabilistic Distributions. MM&Sec : 75-80, 2004.

[24] Cortes, C. et Vapnik, V. (1995). Support-Vector Networks. *Mach. Learn*, 20(3):273–297.

[25] C.J.C. Burges and B. Sch¨olkopf : Improving the accuracy and speed of support vector learning machines, Advances in Neural Information Processing Systems,vol. 9, pp. 375–381, 1997.

[26] C. Fontaine : Contribution à la recherche de fonctions booléennes hautement non linéaires, et au marquage d'images en vue de la protection des droits d'auteur . Thèse de doctorat, Université Paris VI, novembre 1998, Suisse.

[27] Chandramouli, R. & Memon, N.D. (2003). Steganography Capacity: A Steganalysis Perspective, *Proceeding of SPIE Security and Watermarking of Multimedia Contents*, vol. 5020, pp. 173-177.

[28] Compression JPEG. http ://fr.wikipedia.org/wiki/Compression_JPEG.

[29] Cogranne, R. (2011) : Détection statistique d'informations cachées dans une image naturelle à partir d'un modèle physique. Thèse de doctorat, Université de Technologie de Troyes (UTT). Cité page 71.

[30] C. W. Brown et B. J. Shepherd : Graphics File Formats, reference and guide. Manning, 1995.

[31] C.E. Shannon et W. Weaver : The Mathematical Theory of Communication. University of Illinois Press, Urbana, 1949.

[32] C. Cachin : An information-theoretic model for steganography. Information and Computation, 192(1):41–56, juillet 2004.

[33] C. Cachin : Digital steganography. In H.C.A. van Tilborg, ´editeur : Encyclopedia of Cryptography and Security. Springer, 2005. ISBN : 978-0-387-23473-1.

[34] Chen Ming, Zhang Ru, Niu Xinxin et Yang Yixian, (2006). Analysis of Current Steganography Tools : Classifications & Features. Information Security Center, Beijing University of Posts & Telecomm.

[35] C. Fontaine et F. Galand : How can Reed-Solomon codes improve steganographic schemes. In Proc. International Workshop on Information Hiding, IH'07, Lecture Notes in Computer Science, Saint-Malo, France, juin 2007. Springer.

[36] Djeffal, A. (2012) : Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données. Thèse de doctorat, Université Mohamed Khider - Biskra.

[37] D. Brown : Da Vinci Code. Jean-Claude Lattès, 2004. ISBN : 2709624931.

[38] D. Brown : Forteresse Digitale. Jean-Claude Lattès, février 2007. ISBN : 2709626306.

[39] D. Kahn : The Codebreakers. MacMillan, New York, 1967.

[40] D. Sieberg : Bin Laden exploits technology to suit his needs. CNN, septembre 2001.

[41] Des messages cachés sur l'internet pour préparer les attentats. Agence Française de Presse, 12 octobre 2001.

[42] D. Sch¨onefeld et A. Winkler : Embedding with syndrome coding based on BCH codes. In Proc. ACM Multimedia and Security Workshop 2006, pages 214–223. ACM, 2006.

[43] D. Sch¨onefeld et A. Winkler : Reducing the complexity of syndrome coding for embedding. In Proc. International Workshop on Information Hiding, IH'07, Lecture Notes in Computer Science, Saint-Malo, France, juin 2007. Springer.

[44] D.A. Huffman : A method for the construction of minimum redundancy codes. In Proc IRE, volume 40, pages 1098–1101, septembre 1952.

[45] Dong, J., Wang, W. et Tan, T. (2009). Multi-class Blind Steganalysis Based on Image Run-Length Analysis. InDigitalwatermarking, the 8th international conference, IWDW'09, pages 199–210, Berlin, Heidelberg. Springer-Verlag.

[46] E.R. Ardabili, K.Maghooli. E.Fatemizadeh : Contourlet Features Extraction and adaboost Classification for Palmprint Verification , Journal of American Science.2011:7(7).

[47]E. H. Miller : A note on reflector arrays (Periodical style—Accepted for publication), IEEE Trans. Antennas Propagat., to be published.

[48] Eric Cole, (2003).Hiding in Plain Sight : Steganography and the Art of Covert Communication. ISBN : 0-471-44449-9

[49] Eiji. Kawaguchi et Richard O. Eason.: Principle and applications of BPCS-Steganography . University of Maine, Orono.

[50] ´E. Incerti : Compression d'image. Algorithmes et standards. Vuibert, 2003. ISBN : 2-7117-4815-4.

[51] E. Renold, S.J. Creighton, C. Atkinson et J. Carr : Images of abuse : A review of the evidence on child pornography. Rapport technique, National Society for the Prevention of Cruelty to Children (NSPCC), octobre 2003.

[52] Ettinger, J. M. (1998). Steganalysis and Game Equilibria. In Information Hiding - 2nd International Workshop, volume 1525 de Lecture Notes in Computer Science, IH'98, pages 319–328, Portland, Oregon, USA.

[53] F. Raynal, F. Petitcolas et C. Fontaine : L'art de dissimuler les informations. Pour la Science, été 2002. Dossier " L'art du secret ".

[54] F. Galand et G. Kabatiansky : Information hiding by coverings. In Proc. ITW'03, pages 151–154, 2003.

[55] Frezza-Buet, H. (2012). Machines à Vecteurs Supports Didacticiel. Support de cours, Supélec, France.

[56] Fabian Galand : Stéganaographie, Traité de Sécurité des systèmes d'information, Techniques de l'Ingénieur, Ch H 5870, 2004.

[57]Filler, T. (2011): IMPERFECT STEGOSYSTEMS – ASYMPTOTIC LAWS AND NEAR-OPTIMAL PRACTICAL CONSTRUCTIONS. Thèse de doctorat, Binghamton University.

[58] Fridrich, J., Kodovský, J., Holub, V. et Goljan, M. (2011a). Breaking HUGO – the Process Discovery. *In* Filler, T., Pevný, T., Craver, S. et Ker, A. D., éditeurs : *Information Hiding - 13th International Conference*, volume 6958 de *Lecture Notes in Computer Science, IH'11*, Prague, Czech Republic. Springer.

[59] Fridrich, J. et Kodovský, J. (2012). Rich Models for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882.

[60] Fridrich, J., Goljan, M., Hogea, D. et Soukal, D. (2003). Quantitative steganalysis of digital images: estimating the secret message length. *Multimedia Systems*, 9(3):288–302.

[61] F5 : http://www.inf.tu-dresden.de/~aw4

[62] Guorong Xuan1, Yun Q. Shi2: Steganalysis Based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions. IH 2005, LNCS 3727, pp. 262 – 277, 2005.

[63] G. Kipper : Investigator's guide to steganography. Information Security. Auerbach, 2004. ISBN : 0-8493-2433-5.

[64] G. Liang, S. Wang, and X. Zhang : Steganography in binary image by checking data-carrying eligibility of boundary pixels, Journal of Shanghai University, vol. 11, no. 3, pp. 272-277, 2007.

[65] G.K. Wallace : The JPEG still picture compression standard. Commun. ACM, 34(4):30–44, 1991.

[66] Gul, G. et Kurugollu, F. (2011). A New Methodology in Steganalysis : Breaking Highly Undetectable Steganograpy (HUGO). *In Information Hiding -13th International Workshop*, Lecture Notes in Computer Science, IH'11, pages 71–84,Prague, Czech Republic. Springer-Verlag.

[67] Hassan Masood, Mohammad Asim, Mustafa Mumtaz and Atif Bin Mansoor : Combined Contourlet and Non- subsampled Contourlet Transforms Based Approach for Personal Identification using Palmprint , International Conference on Digital Image Computing: Techniques and Applications, pp.408-415, 2009.

[68] Hasan,M. et Boris, F. (2006). Svm :Machines à vecteurs de support ou séparateurs à vastes marges. Rapport technique, Versailles St Quentin, France.

[69] I. Steinwart : Sparseness of support vector machines, Journal of Machine Learning Research, vol. 4, pp.1071–1105, 2003.

[70] I. Avcibas, N. Memon, and B. Sankur : Steganalysis Using Image Quality Metrics. In Security and Watermarking of Multimedia Contents, SPIE. San Jose, CA, 2001.

[71] I. Avcibas, N. Memon, and B. Sankur : Image Steganalysis With Binary Similarity Measures. In IEEE International Conference on Image Processing, Rochester, New York, 2002.

[72]J. Fridrich: Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes, Proceding of the 6th Information Hiding Workshop, Springer, vol. 3200 , pp.67-81, 2004.

[73] J. Fridrich , J. Kodovský : Calibration Revisited . MM&Sec'09, September 7–8, 2009, Princeton, New Jersey, USA. Copyright 2009 ACM 978-1-60558-492-8/09/07.

[74] Jon Yngve Hardeberg , Robert Jenssen : Image Analysis . 16th Scandinavian Conference, SCIA 2009 Oslo, Norway, June 15-18, 2009. chapitre33. A New Hybrid DCT and Contourlet Transform Based JPEG Image Steganalysis Technique.

[75] Jing-qu Lin. Wang : Reduction of markov extended features in JPEG image Steganalysis , 2nd Internetional congress on image and Signal processing,2009.

[76]J. Fridrich, M. Goljan and D. Hogea : Steganalysis of JPEG images: Breaking the F5 algorithm . In Information Hiding , 5th International Workshop, volume 2578 of Lecture Notes in Computer Science, Noordwijkerhout, The Netherlands, (Springer-Verlag, New York, pp310–323 ,2002).

[77] J. Barbier, ´E. Filiol et K. Mayoura : Universal JPEG steganalysis in the compressed frequency domain. In Y. Q. Shi et B. Jeon, éditeurs : Proc. Digital Watermarking, 5th International Workshop, IWDW 2006, volume 4283 de Lecture Notes in Computer Science, pages 253–267, Jeju Island, Korea, novembre 2006. Springer.

[78]J. Wang : Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication), IEEE J. Quantum Electron., submitted for publication.

[79]J. U. Duncombe, Infrared navigation—Part I: An assessment of feasibility (Periodical style), IEEE Trans. Electron Devices, vol. ED-11, pp. 34–39, Jan. 1959.

[80] J.A. Reeds : Solved : The ciphers in book III of Trithemius's Steganographia. Cryptologia, (22):291–319, octobre 1998.

[81] J.C. Judge : Steganography : Past, Present and Future. SANS, 2001.

[82] J. Kelley : Terror groups hide behind Web encryption. USA Today, mai 2001.

[83] J. Kelley : Terrorist instructions hidden online. USA Today, mai 2001.

[84] J.-P. Bay : Attention, une image peut en cacher une autre. Lci, septembre 2001.

[85] J. Barbier : La stéganographie moderne : d'Hérodote à nos jours. In Computer & Electronics Security Application Rendez-vous, CESAR 2007, Rennes, France, novembre 2007.

[86] J. Barbier, G. Sicot et S. Houcke : Algebraic approach for the reconstruction of linear and convolutional error correcting codes. In C. Ardil, éditeur : Proc. 3rd International Conference

on Computer Science and Engineering CISE 2006, volume 16, pages 66–71, Venice, Italy, novembre 2006. World Enformatika Society. ISBN : 975-00803-6-X.

[87] J. Barbier et S. Alt : Practical insecurity for effective steganalysis. In Proc. of 10th International Workshop on Information Hiding, IH 2008, Lecture Notes in Computer Science, Santa Barbara (CA), USA, mai 2008. Springer.

[88] J. Barbier, ´E. Filiol et K. Mayoura : Universal detection of JPEG steganography. Journal of Multimedia, 2(2):1–9, avril 2007. ISSN : 1796-2048.

[89] J. Fridrich, M. Goljan et R. Du : Reliable detection of LSB steganography in grayscale and color images. In Proc. ACM Workshop on Multimedia and Security, pages 27–30, Ottawa, Canada, octobre 2001.

[90] J. Fridrich, M. Goljan, P. Lisonek et D. Soukal : Writing on wet paper. IEEE Transactions on Signal Processing, 53(10):3923–3935, 2005. Special issue ”Supplement on Secure Media III”.

[91] J. Fridrich, M. Goljan et D. Soukal : Efficient wet paper codes. In Proc. 7th International Workshop on Information Hiding, volume 3727 de Lecture Notes in Computer Science, pages 204–218. Springer, 2005.

[92] J. Fridrich, M. Goljan et D. Soukal : Wet paper codes with improve embedding efficiency. IEEE Transactions on Security and Forensics, 1(1):102–110, 2006.

[93] J. Fridrich et D. Soukal : Matrix embedding for large payloads. IEEE Transactions on Security and Forensics, 1(3):278–294, 2006.

[94] Jessica Fridrich, Miroslav Goljan et Dorin Hogea, (2000): Attacking the Outguess. Departement of Electrical and Computer Engineering, SUNY Binghamton.

[95] J. Barbier, ´E. Filiol et K. Mayoura : New features for specific JPEG steganalysis. In C. Ardil, éditeur : Proc. 3rd International Conference on Computer, Information, and Systems Science, and Engineering, CISE 2006, volume 16 de Transactions on Engineering, Computing and Technology, pages 72–77. World Enformatika Society, novembre 2006. ISBN : 975-00803-6-X.

[96] J. Barbier et K.Mayoura : Steganalysis of the Multi Bit Plane Image Steganography. In Proc. of Digital Watermarking, 6th International Workshop, IWDW 2007, Lecture Notes in Computer Science, Guangzhou, China, décembre 2007. Springer.

[97] J. Fridrich, M. Goljan et R. Du : Detecting LSB steganography in color and gray-scale images. IEEE MultiMedia, 8(4):22–28, 2001.

[98] J. Z¨ollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke et GrittaWolf : Modeling the security of steganographic systems. In D. Aucsmith, éditeur :

Proc. Information Hiding. Second International Workshop, IH'98, volume 1525 de Lecture Notes in Computer Science, pages 344–354, Portland, Oregon, USA, avril 1998. Springer-Verlag.

[99] Johann BARBIER : Analyse de canaux de communication dans un contexte non coopératif. Thèse de doctorat, école Supérieure et d'Application des Transmissions école Polytechnique, Laboratoire de Virologie et Cryptologie le 28 novembre 2007.

[100] Joshua Silman: Steganography and Steganalysis: An Overview**,** gsec 1.2 f, 2001**.**

[101] J. Fridrich, M. Goljan, and D. Hogea : Steganalysis of JPEG Images: Breaking the F5 Algorithm. In Proceedings, Information Hiding, 5th International Workshop, IH 2002, pp, 310-323. Noordwijkerhout, The Netherlands, 2002.

[102] J. Pasquet, Bringay, S. et Chaumont, M. (2013). Des millions d'images pour la stéganalyse : inutiles ! *In COmpression et REprésentation des Signaux Audiovisuels, CORESA'13*, Creusot, France.

[103] JPHS : http://linux01.gwdg.de/~alatham/stego.html

[104] J.Kodovský, Fridrich, J. etHolub, V. (2012) : Ensemble Classifiers for Steganalysis of Digital Media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444.

[105] J. Kodovský et Fridrich, J. (2012a). JPEG-Compatibility Steganalysis Using Block-Histogram of Recompression Artifacts. *In Information Hiding - 14th InternationalWorkshop*, volume 7692 de *Lecture Notes in Computer Science, IH'12*, pages 78–93, Berkeley, CA, USA. Springer-Verlag.

[106] J. Kodovský et Fridrich, J. (2011) : Steganalysis in High Dimensions: Fusing Classifiers Built on Random Subspaces. In Media Watermarking, Security, and Forensics XIII, part of IS&T SPIE Electronic Imaging Symposium, volume 7880, paper. 21, pages L 1–12, San Francisco, CA.

[107] J .Kodovský, Filler, T., Fridrich, J. et Holub, V. (2011). On Dangers of Overtraining Steganography to Incomplete. CoverModel. *InMultimedia and Security Workshop, MM&Sec '11 Proceedings of the 13th ACM multimedia*, pages 69–76, Buffalo, NY, USA. ACM.

[108]K.-M. Lin and C.-J. Lin : A study on reduced support vector machines, IEEE Transactions on Neural Networks, vol. 14, pp. 1449–1459, 2003.

[109] Ker, A. D. (2006). Batch steganography and pooled steganalysis, *Proceeding of Information Hiding Workshop*, vol. 4437, pp. 265–281.

[110] K. Mayoura : Analyse stéganographique d'une image JPEG. Mémoire de DESS, Universit´e du mans, 2004.

[111] Ker, A. et Pevný, T. (2012b). Identifying a steganographer in realistic and heterogeneous data sets. *InMediaWatermarking, Security, and Forensics IV, part of IS&T SPIE Electronic Imaging Symposium*, volume 8303, San Francisco, California, USA.

[112] Ker, A. et Pevný, T. (2011). A New Paradigm for Steganalysis via Clustering. *In Media Watermarking, Security, and Forensics III, part of IS&T SPIE Electronic Imaging Symposium*, volume 7880, pages 0U01–0U13, San Francisco, California, USA.

[113]L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds: Large-Scale Kernel Machines , MIT Press, 2007.

[114] Les efforts de la NSA vis-à-vis du Web : la stéganographie. Le Monde du Renseignement , 26 octobre 2000.

[115] Lyu, S. & Farid, H. (2002): Detecting hidden messages using higher-order statistics and support vector machines, Proceeding of 5th International Workshop on Information Hiding.

[116] Lubenko, I. et Ker, A. (2012a). Going from Small to Large Data in Steganalysis. *In Media Watermarking, Security, and Forensics IV, part of IS&T SPIE Electronic Imaging Symposium*, volume 8303, pages 0M01–0M10, Burlingame, California, USA.

[117] L. von Ahn et N. J. Hopper : Public-key steganography. In C. Cachin et J. Camenisch, éditeurs : Proc. Eurocrypt 2004, volume 3027 de Lecture Notes in Computer Science, pages 323–341, Interlaken, Switzerland, mai 2004. Springer. ISBN : 3-540-21935-8.

[118] Ljupce Nikolov : Stéganographie Détection de messages cachés, Travail de diplôme, Haute Ecole d'ingénierie et de Gestion, 14 décembre 2008.

[119]M. Young: The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[120] M. Chapman : Hiding the Hidden : A Software System for Concealing Ciphertext in Innocuous Text. These de doctorat, The University of Wisconsin-Milwaukee, mai 1997.

[121] M. Cluzeau : Reconnaissance d'un code linéaire en bloc en utilisant un algorithme de décodage itératif. In Journées Codage et Cryptographie, Eymoutiers, France, octobre 2006.

[122] M. Cluzeau : Reconnaissance d'un schéma de codage. Thèse de doctorat, ´Ecole Polytechnique, Palaiseau, France, novembre 2006.

[123] M. Wu, E. Tang, and B. Lin : Data hiding in digital binary image, Proc. Of 2000 IEEE International Conference on Multimedia and Expo, vol. 1, pp. 393-396, 2000.

[124] M. Goljan J. Fridrich and D. Hogea : New Methodology for Breaking Steganographic Techniques for Jpegs. In EI SPIE Santa Clara, CA, 2003.

[125] Miche, Y., Bas, P. et Lendasse, A. (2010). Using Multiple Re-Embeddings For Quantitative Steganalysis and Image Reliability Estimation. Rapport technique TKK- ICS-R34, Aalto University School of Science and Technology, Aalto, Finland.

[126] Mbsteg : P. Sallee. Model-based steganography. In *Digital Watermarking*, volume 2939/2004 of *Lecture Notes in Computer Science*, pages 154–167. Springer Berlin / Heidelberg, 2004.

[127] MM3 : Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In *Information Hiding 2007*, volume 4437/2007, pages 314–327, 2007.

[128] N. PROVOS and P. HONEYMAN: Detecting steganographic content on the internet. In Network and Distributed System Security Symposium. The Internet Society, 2002.

[129] Neil F. Johnson et Sushil Jajodia, (1998). Steganalysis : The Investigation of Hidden Information. In proceding IEEE Information Technology Conference.

[130] N. Stephenson : Le Cryptonomicon. Payot, avril 2000.

[131] N.F. Johnson et S. Jajodia : Exploring steganography : Seeing the unseen. IEEE Computer, 31(2):26–34, 1998.

[132] N.F. Johnson, Z. Duric et S. Jajodia : Information Hinding - Steganography and watermarking - Attacks and countermeasures. Adavances in Information Security. Kluwer Academic. ISBN : 0-7923-7204-2.

[133] N. Provos : Universal steganography., aout 1998. http ://wwww.outguess.org/.

[134] N. Provos : Defending against statistical steganalysis. In 10th USENIX Security Symposium, Washington, DC, USA, 2001.

[135] Niels Provos et Peter Honeyman, (2003). Hide and Seek : An Introduction to Steganography. IEEE Computer Society.

[136] N.J. Hopper : On steganographic chosen covertext security. In Proc. International Colloquium on Automata Languages and Programming, ICALP 2005, volume 3580 de Lecture Notes in Computer Science, pages 311–323, Lisboa, Portugal, 2005. Springer. ISBN : 3-540-27580-0.

[137] N.J. Hopper, J. Langford et L. von Ahn : Provably secure steganography. In M. Yung, éditeur : Proc. Crypto 2002, volume 2442 de Lecture Notes in Computer Science, pages 77–92, Santa Barbara, CA, USA, aout 2002. Springer. ISBN : 3-540- 44050-X.

[138] N.J. Hopper : Toward a Theory of Steganography. These de doctorat, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, juillet 2004.

[139] N. Dedié, G. Itkis, L. Reyzin et S. Russel : Upper and lower bounds on blackbox steganography. In Proc. 2nd Theory of Cryptography Conference (TCC 2005), volume 3378 de Lecture Notes in Computer Science. Springer, 2005.

[140] N. Memon and R. Chandramouli: Analysis of LSB Based Image Steganography techniques. In Proceedings of the International Conference on Image Processing, Thessaloniki Greece, 2001.

[141] Nouha Kobsi, F.H.Merouani : Proposition d'un Multi-Classifieur pour une Stéganalyse d'Image, Infodays'08-Chlef, Algérie,15-16,2008.

[142] P. Wayner : Disappearing cryptography - Information Hiding : steganography & watermarking. Morgan Kaufmann, 2002. ISBN : 1-55860-769-2.

[143] P. Lu, X. Luo, Q. Tang et L. Shen : An improved sample pairs method for detection of LSB embedding. In J. Fridrich, éditeur : Proc. Information Hiding, 6th International Workshop, volume 3200 de Lecture Notes in Computer Science, pages 116–127, Toronto, Canada, mai 2004. Springer. ISBN : 3-540-24207-4.

[144] Pevny, T.; Fridrich, J. & Ker, A.D. (2009): From Blind to Quantities Steganalysis, Proceeding of SPIE, Electronic Imaging, Security and Forensics of Multimedia Contents, pp. 0C 1-0C 14.

[145] P. Sallee, (2003): Model-based steganography, Proceeding of International Workshop on Digital Watermarking.

[146]Pevný, T. et Fridrich, J. (2006) : Multi-class Blind Steganalysis for JPEG Images. In Media Watermarking, Security, and Forensics VIII, part of IS&T SPIE Electronic Imaging Symposium, volume 6072, pages 1–13, San Francisco, California, USA.

[147] Pevný, T., Filler, T. et Bas, P. (2010) : UsingHigh-Dimensional ImageModels to Perform. Highly Undetectable Steganography. In Information Hiding - 12th In- ternational Conference, volume 6387 de Lecture Notes in Computer Science, IH'10, pages 161–177, Calgary, AB, Canada. Springer-Verlag.

[148] R. Chandramouli : Data hiding capacity in the presence of an imperfectly known channel. In Proc. SPIE Security and Watermarking of Multimedia Contents II, volume 4314, 2001.

[149] R. Chandramouli et N.D. Memon : Steganography capacity : A steganalysis perspective. In Proc. SPIE, Security and Watermarking of Multimedia Contents V, volume 5020, pages 173–177, Santa Clara, CA, USA, janvier 2003.

[150] R. Chandramouli, M. Kharrazi et N.D. Memon : Image steganography and steganalysis : Concepts and practice. In T. Kalker, I. J. Cox et Y.M. Ro, éditeurs : Proc. Digital Watermarking, Second International Workshop, IWDW 2003, volume 2939 de Lecture Notes in Computer Science, pages 35–49, Seoul, Korea, octobre 2003. Springer. ISBN : 3-540-21061-X.

[151] R. Chandramouli : Mathematical theory for steganalysis. In Proc. SPIE Security and Watermarking of Multimedia Contents IV, 2002.

[152] R. Crandrall : Some notes on steganography. Posted on Steganography Mailing List, 1998. http ://os.inf.tu-dresden.de/~westfeld/crandall.pdf.

[153]Simmons, G.J., : The Prisoners problem and the subliminal channel , in Advances in Cryptology, Proceedings of CRYPTO 83, Plenum Press, 1984, p.51-67.

[154] Sharkas, M, El-Rube.I, Mostafa,M.A : The Contourlet Transform with the principal component analysis for palmprint Recognition , International conference on computational intelligence, communication systems&Networks, PP.262-267,2010 .

[155] S. Mika, G. Ratsch, J. Weston, B. Sch¨olkopf, and K. Muller : Fisher discriminant analysis with kernels, in IEEE Neural Networks for Signal Processing Workshop,1999, pp. 41–48.

[156] S. Wu : Support vertor machine classifiers by modifying kernel function. Journal of neural networks.No.12,pp.783 789,1999.

[157] Si Wu , S.Amari : conformal transformation of kernel function: A Data-dependent way to improve support vector machine clessifiers, Neural processing .Nu.15 .2002

[158] Schöttle, P. et Böhme, R. (2012) : A Game-Theoretic Approach to Content-Adaptive Steganography. In InformationHiding - 14th InternationalWorkshop, volume 7692 de Lecture Notes in Computer Science, IH'12, pages 125–141, Berkeley, CA, USA. Springer-Verlag.

[159] S. Katzenbeisser et F.A.P. Petitcolas : Information Hiding. Techniques for steganography and digital watermarking. Computer Science. Artech House. ISBN : 1-5853 035-4.

[160] S. Dumitrescu, X. Wu et Z. Wang : Detection of LSB steganography via sample pair analysis. In Fabien A. P. Petitcolas, éditeur : Proc. Information Hiding, 5th International Workshop, volume 2578 de Lecture Notes in Computer Science, pages 355–372, Noordwijkerhout, The Netherlands, octobre 2002. Springer. ISBN : 3-540-00421-1.

[161] S. Lyu et H. Farid : Steganalysis using higher-order image statistics. IEEE Transactions on Information Forensics and Security, 1, 2006.

[162] 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.

[163] Steganalysis with mismatched covers : do simple classifiers help ? *In Multimedia and Security Workshop, MM&Sec '12 Proceedings of the 14th ACMmultimedia*, pages 11–18, Coventry,United Kingdom. ACM.

[164] StegHide : http://steghide.sourceforge.net/

[165] Tomas Pevny and Jessica Fridrich: Merging markov and dct features for multi-class jpeg steganalysis. In Proceding. of SPIE: Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, pp. 3-14, 2007.

[166]T. Downs, K.E. Gates, and A. Masters : Exact simplification of support vector solutions, Journal of Machine Learning Research, vol. 2, pp. 293–297, 2001.

[167] T. Pevny and J. Fridrich : Towards Multi-class Blind Steganalyzer for JPEG Images. In Proc. IWDW, 2005, pp.39-53.

[168]Transformée en cosinus discrète.http ://fr.wikipedia.org/wiki/Transform%C3%A9e_en_cosinus_discr%C3%A8te.

[169] Upham, D. (1992-1997). Jpeg-Jsteg, modification of the independent JPEG's group's JPEG software (release 4) for 1-bit steganography in JFIF output files.

[170] van Damme, E. (1991). Stability and Perfection of Nash Equilibria. Springer-Verlag.

[171] Vapnik, V. N. (1995): The nature of statistical learning theory. Springer-Verlag New York, Inc, New York, NY, USA.

[172] V. Holub et Fridrich, J. (2012): Designing SteganographicDistortion Using Directional Filters. *In IEEE Workshop on Information Forensic and Security*, *WIFS'12*, Tenerife, Spain.

[173] Westfeld, A. (2001). F5–A Steganographic Algorithm: High Capacity Despite Better Steganalysis. *In Information Hiding - 4th International Workshop*, volume 2137, pages 289–302, New York, Pittsburgh, PA. Springer-Verlag.

[174] Xiang.Yang luo, Dao. Shun Wang : A preview on blind detection for image steganography ,signal processing.88(9):2138-2157,2008.

[175] X. Y. Luo, D. S.Wang, P.Wang, and F. L. Liu : A review on blind detection for image steganography, Signal Processing, vol. 88, no. 9, pp. 2138-2157, 2008.

[176] Y. Q. Shi, C. Chen, and W. Chen: A morkov process based approach to Rective attacking jpeg steganography. Proceding of the 8th Information Hiding Workshop, Springer, vol. 4437, pp. 249-264,2006.

[177] Y. Q. Shi, C. Chen ,et W. Chen : A Markov process based approach to effective attacking JPEG steganography. In Proceedings of the 8th Information Hiding Workshop, 2006.

[178] Yunhong Wang; Tieniu Tan; Lei Guo: Blind Image Steganalysis Based on Statistical Analysis of Empirical Matrix Xiaochuan Chen. Pattern Recognition, ICPR 2006,18th International Conference on Volume 3, 0-0 0 Page(s):1107 – 1110. (2006).

[179]Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa : Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style), IEEE

Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740–741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[180] Y. MICHE, P. BAS, A. LENDASSE, C. JUTTEN, and O. SIMULA: Extracting relevant features of steganographic schemes by feature selection techniques. In Wacha'07: Third Wavilla Challenge, June 14 2007.

[181] Ying Wang : Optimized Feature Extraction for Learning-Based Image Steganalysis Student Member, IEEE, and Pierre Moulin, Fellow, IEEE. 2006.

[182] Zhuo Li, Kuijun Lu, Xianting Zeng, Xuezeng Pan: A Blind Steganalytic Scheme Based on DCT and Spatial Domain for JPEG Images. JOURNAL OF MULTIMEDIA, VOL. 5, NO. 3, JUNE 2010.

Annexes A

Logiciels de stéganographie

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

Blindside: Blinside 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

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/p128/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

Annexes B

Base d'image UCID *et le Toolbox Matlab*

Pour validé notre approche de stéganalyse, il serait nécessaire d'enrichir la base d'image avec un nombre d'image steganographie, ce qui nous permettra d'élargir la taille de la basse d'image, cependant la performance et la fiabilité de seront meilleurs.

La basse UCID « an Uncompressed Color Image Database » constituée de 1338 image TIFF « Tagged Image Fille Format » UCID est une base très variée prises dans divers thèmes, elle contient une variété de scènes extérieures et intérieure prises pendant la journée et d'autre pendant la nuit, elle inclue aussi des images comme illustre la figure au dessous.

- **La nature** : paysages, foret, fleurs, animaux, mer, places publics…
- **Portraits** : Femmes, Hommes, enfants, jeunes, vieux, la foule, de profile, en face, de dos.
- **Objets synthétiques** : outils de cuisine, voitures, horloges, jouets pour enfants, statuts, bâtisses…
- **D'autres images**.

Les images ont ensuite été redimensionées en 800 × 600 (multiples de 8) afin d'éviter les possibles effets de bloc dus à la recompression JPEG sur une grille différente. Elles sont également transposées vers un espace de couleur en niveaux de gris, et finalement enregistrées sous un format sans pertes (pgm pour notre étude). Ce format sans perte est utilisé pour éviter toute recompression de l'image lors de l'opération de découpage. Les images sont découpées au format 512 × 512 et finalement enregistrées en JPEG, avec un facteur de qualité de 80 % (notre procédé d'extraction utilise des images de cette taille et avec ce facteur de qualité), dans l'espace de couleurs YCbCr. Même si les expériences suivantes, ainsi que la méthodologie générale sont effectuées avec une telle taille d'image, n'importe quelle taille conviendrait.

Traitement d'image sous MATLAB

1. image processing Toolbox
Les fonctions de la librairie « traitement d'image » de MATLAB :

- Lecture, écriture et affichage d'une couleur ou niveau de gris
- Transformations spatiales et transformation fréquentielles,
- Analyse, et restauration d'image
- Changement d'espace couleur
- Binarisation et restauration d'image

2. Codage d'une image, représentation spatiale
Une image est considérées comme un ensemble de points ou pixels (picture element), associé au quadrillage rectangulaire de l'image originale.la représentation d'une image se fait donc par l'intermédiaire d'une matrice d'entier codés entre 0 et 255.les images en niveau de gris sont représentées par des matrices 2D, les images couleurs représentés 3 composantes (Rouge, Vert, Bleu) sont représentés par 3 matrices 3D.On accède à un pixel grâce à son indice de ligne et son indice de colonne. Le premier pixel d'une image est le pixel en haut à gauche. Cette représentation est appelé spatiale de l'image.

3. Histogramme d'une image
Un histogramme est un graphique statistique permettant de représenter le nombre de pixels pour chaque intensité luminance. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (a droite).on calcul l'histogramme d'une image par la fonction « imhist ».

4. Détection de contours
La détection de contours permet de faire ressortir les variations importantes de l'image

- Fonction edge

Annexes C

Algorithme de stéganographie

Algorithme d'insertion MBPIS

Entrées : un message secret M de l bits,

I une image non compressée,

K_e,K_s les clés cryptographique et stéganographique.

Sortie : une image stéganographiée ou échec.

Paramètres : i_{max} le plan de bits maximum,

t, le seuil,

m × n, la taille de la fenêtre.

1 **Coder** *I* en *I′* dans le CGC selon (5.1)

2 **Décomposer** *I′* en N plans de bits

3 **Compresser** et **Chiffrer** *M* en *M′* avec K_e

4 **Initialiser** le GPA à l'aide de K_s

5 **Pour** *i* de i_{max} a 1

6 **Déterminer** les zones non homogènes de B_i taille m × n avec le seuil t selon (5.4)

7 **Insérer** les bits de *M′* dans les bits du plan B_i des zones non homogènes à l'aide du GPA

8 **Fin Pour**

9 **Si** il reste des bits de *M′* non insérés alors

10 **Renvoyer** échec

11 **Fin Si**

12 **Coder** *I′* dans le plan de bits usuel selon (5.2)

13 **Renvoyer** *I′*

Figure C.5 : Algorithme d'insertion MBPIS

Algorithme d'extraction MBPIS

Entrées : une stégo image I',

K_e,K_s les clés cryptographique et stéganographique.

Sortie : le message M dissimulé.

Paramètres : i_{max} le plan de bits maximum,

t, le seuil,

m × n, la taille de la fenêtre.

1 **Coder** I' dans le CGC selon (5.1)

2 **Décomposer** I' en N plans de bits

3 **Initialiser** le GPA a l'aide de K_s

4 **Pour** i de i_{max} a 1

5 **Déterminer** les zones non homogènes de B_i taille m × n
avec le seuil t selon (5.4)

6 **Lire** les bits de M' dans les bits du plan Bi des zones
non homogènes a l'aide du GPA

7 **Fin Pour**

8 **Déchiffrer** M'avec k_e puis le décompresser en M

9 **Renvoyer** M

Figure C.6 : Algorithme d'extraction MBPIS

Algorithme de Huffman

Entrées : une séquence *S* de *N* symboles.

Sortie : code de Huffman.

1 Ordonner les *N* symboles par ordre croissant de leur nombre d'occurences dans *S*,

Chaque couple (symbole, occurrence) étiquetant un arbre réduit à sa racine.

2 **Tant qu'**il reste plus d'un arbre binaire

3 Retirer de la liste les deux arbres binaires de poids minimum

4 Fusionner ces arbres en un troisième de poids la somme des poids de ces fils

5 Affecter `a l'arc vers le fils de droite la valeur 1 et vers celui de gauche la valeur 0

6 Insérer ce nouvel arbre dans la liste

7 **Fin Tant que**

8 Affecter à chaque feuille la concaténation de la valeur des arcs depuis la racine

9 **Renvoyer** pour chaque feuille, le couple (symbole, valeur binaire)

Figure C.14– Algorithme de construction du code de Huffman

Algorithme de correction statistique d'Outguess

Entrées : *f* l'histogramme des coefficients DCT, $(DCT_i)_{i=1...n}$ les coefficients DCT.

Sortie : $(DCT'_i)_{i=1...n}$ les coefficients DCT corrigés.

1 $T_i \leftarrow (150/n)f_i$

2 Mod[i] ← 0

3 **Pour** *i* de *1* à *n*

4 **Si** DCT_i non modifié **alors continué** en 3

5 Adj ← $DCT_i \oplus 1$

6 **Si** Mod[Adj] 6≠ 0 **alors**

7 Mod[Adj] ← Mod[Adj] − 1

8 **continué** en 3

9 **Fin Si**

10 **Si** $Mod[DCT_i] < T_{DCTi}$ **alors**

11 $Mod[DCT_i] \leftarrow Mod[DCT_i] + 1$

12 **continué** en 3

13 **Fin Si**

14 **Si corriger** (i,DCT_i) échoue **alors**

15 $Mod[DCT_i] \leftarrow Mod[DCT_i] + 1$

16 **continué** en 3

17 **Fin Si**

18 **Fin Pour**

19 **Pour** chaque Mod[*i*] 6≠ 0

20 **Tant que** Mod[*i*] 6≠ 0

21 Mod[*i*] ← Mod[*i*] − 1

22 **corriger** (n,DCT_i)

23 **Fin Tant que**

24 **Fin Pour**

25 **Renvoyer** (DCT'_i) = (DCT_i)

Figure C.19– Algorithme de correction statistique d'Outguess

Algorithme d'insertion d'Outguess

Entrées : *I* une image non compressée,

M un message,

K_s une clé stéganographique, IV un IV.

Sortie : *I′* une stégo image, *d* distortion.

1 **Compresser** *I* jusqu' à la quantification

2 $(DCT'_i) \leftarrow (DCT_i)$

3 $M' \leftarrow Enc\ (RC4\ (K_s, M))$

4 $l' \leftarrow$ longueur(M′)

5 **Initialiser** le GPA a l'aide de K_s

6 **Générer** (pos_i) selon (5.6)

7 $M_1 \leftarrow IV \,||l'$

8 **Pour** *i* de 1 à 32

9 $LSB(DCT'_{posi}) \leftarrow M_1(_i)$

10 **Fin Pour**

11 **Initialiser** le GPA a l'aide de IV

12 **Générer** (pos_i) selon (5.6)

13 **Pour** *i* de 1 à *l′*

14 $LSB\ (DCT'_{posi}) \leftarrow M'\ (i)$

15 **Fin Pour**

16 $E_1 \leftarrow \{DCT'_i \neq DCT_i\}$

17 $f \leftarrow$ histogramme des (DCT'_i)

18 $DCT' \leftarrow$ **Correction** $(f, (DCT'_i))$

19 **Terminer** la compression JPEG

20 $E_2 \leftarrow \{DCT'_i \neq DCT_i\} \setminus E_1$

21 $d \leftarrow \text{Card}(E_1 S E_2) + \sum_{c \in E1} D(c) - \sum_{c \in E2} D(c)$

22 **Renvoyer** I' et d

Figure C.20– Algorithme d'insertion d'Outguess

Algorithme d'extraction d'Outguess

Entrées: I′ une stégo image JPEG,

K_s une clé stéganographique.

Sortie : M un message.

1 **Décompresser** I′ jusqu' à la quantification

5 **Initialiser** le GPA a l'aide de K_S

6 **Générer** (pos_i) selon (5.6)

8 **Pour** i de 1 à 32

9 $M_1(i) \leftarrow LSB(DCT'_{posi})$

10 **Fin Pour**

7 $M_1 \rightarrow IV \,||\, l$

11 **Initialiser** le GPA a l'aide de *IV*

12 **Générer** (pos_i) selon (5.6)

13 **Pour** i de 1 à *l*

14 $M(i) \leftarrow LSB(DCT'_{posi})$

15 **Fin Pour**

3 $M \leftarrow Dec(RC4^{-1}(K_S, M))$

22 **Renvoyer** *M*

Figure C.21 : Algorithme d'extraction d'Outguess

Algorithme d'insertion de F5

Entrées : un message secret $M = m1m2 \ldots$ de l bits,

I une image non compressée,

K_s la clé stéganographique.

Sortie : une image stéganographiée.

1 **Compresser** I au format JPEG, jusqu''a la quantification

2 **Initialiser** le GPA a l'aide de K_s

3 **Mélanger** aléatoirement les coefficients DCT a l'aide du GPA

4 Déterminer le paramètre s à partir de l et I

5 **Pour** chaque bloc de message mi de longueur s

6 **Remplir** un buffer v avec les n *LSB* des coefficients *DCT* non nuls suivants

7 **Calculer** $j = m_i \oplus S(v)$ selon (5.9)

8 **Si** $j \neq 0$ alors décrémenter la valeur absolue du coefficient *DCT* de *LSB* v_j

9 **Fin Si**

10 **Si** un effondrement apparaît alors

11 $v_k \leftarrow v_k+1$ pour $k = j \ldots (n-1)$

12 $v_n \leftarrow$ *LSB* du prochain *DCT* non nul

13 **Retour** à l''étape 7

14 **Fin Si**

15 **Fin Pour**

16 **Finir** la compression JPEG en I'

17 **Renvoyer** I'

Figure C.27– Algorithme d'insertion de F5

Algorithme d'extraction de F5

Entrées : I' une image JPEG,

K_s la clé stéganographique,

s le paramètre du code,

l la longueur du message.

Sortie : le message *M = m1m2 . . . .*

1 **Décompresser** *I′* jusqu``a la quantification
2 **Initialiser** le *GPA* a l'aide de K_s
3 **Mélanger** aléatoirement les coefficients *DCT* a l'aide du *GPA*
5 **Pour** les *l* blocs de message mi de longueur *s*
6 **Remplir** un buffer *v′* avec les *n LSB* des coefficients *DCT* non nuls suivants
7 **Calculer** $m_i = S(v')$ selon (5.8)
8 **Fin Pour**
9 **Renvoyer** *M = m1m2 . . .*

Figure C.28 – Algorithme d'extraction de F5