



HAL
open science

Insertion adaptative en stéganographie : application aux images numériques dans le domaine spatial

Sarra Kouider

► **To cite this version:**

Sarra Kouider. Insertion adaptative en stéganographie : application aux images numériques dans le domaine spatial. Cryptographie et sécurité [cs.CR]. Université Montpellier II - Sciences et Techniques du Languedoc, 2013. Français. NNT : 2013MON20107 . tel-01020745

HAL Id: tel-01020745

<https://theses.hal.science/tel-01020745>

Submitted on 8 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER II
Sciences et Techniques du Languedoc

THÈSE

présentée au Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier pour
obtenir le diplôme de doctorat

Spécialité : **Informatique**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

Insertion adaptative en stéganographie application aux images numériques dans le domaine spatial

par

Sarra KOUIDER

Soutenue le 17 décembre 2013, devant le jury composé de :

Directeur de thèse

M. William PUECH, Professeur des universités LIRMM, Université Montpellier II

Co-Directeur de thèse

M. Marc CHAUMONT, Maître de conférences, HDR LIRMM, Université de Nîmes

Rapporteurs

M. Patrick BAS, Chargé de recherche CNRS LAGIS, École Centrale Lille

M. Lionel FILLATRE, Professeur des universités I3S, Université de Nice Sophia-Antipolis

Examineurs

M. Jean-Marie MOUREAUX, Professeur des universités CRAN, Université de Lorraine

M. Laurent IMBERT, Chargé de recherche CNRS, HDR LIRMM, Université Montpellier II



Remerciements

La valeur d'un homme tient dans sa capacité à donner et non dans sa capacité à recevoir.

ALBERT EINSTEIN - PHYSICIEN

Bien qu'une thèse soit le fruit d'un travail personnel, je n'aurais jamais abouti à de tels résultats sans le soutien et l'aide des personnes qui m'ont entouré durant ces 3 années. Ces dernières années avaient été pour moi une véritable aventure très enrichissante, que ce soit sur le plan professionnel ou humain. A travers ces quelques paragraphes, je vais essayer de remercier ces personnes à leur juste valeur et je les prie de m'excuser par avance pour ceux que j'oublierais.

Tout d'abord, je tiens à exprimer ma profonde gratitude envers M. Jean-Marie Moureaux, professeur à l'université de Lorraine, et M. Laurent IMBER, chargé de recherche CNRS au LIRMM, pour avoir accepté d'examiner cette thèse.

J'exprime également ma reconnaissance envers M. Patrick Bas, chargé de recherche CNRS au LAGIS, et M. Lionel Fillatre, professeur à l'université de Sophia-Antipolis, 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.

Je remercie particulièrement M. Marc Chaumont, qui a été pour moi plus qu'un encadrant de thèse. Son investissement personnel et sa passion pour le travail sont pour moi un véritable modèle. Toutes ses propositions, ses remarques avisées et nos discussions sans fin m'ont été extrêmement bénéfiques. Je voudrais le remercier également pour sa patience, la confiance qui m'a toujours octroyé, ainsi que pour m'avoir permis de m'affirmer.

Je n'oublie pas de remercier également mon directeur de thèse M. William Puech pour

sa bienveillance, son enthousiasme, et ces conseils. Je le remercie aussi pour m'avoir accueilli chaleureusement au sein de l'équipe ICAR, et pour avoir veillé au bon déroulement et l'aboutissement de ce travail.

J'ai eu la chance d'effectuer ma thèse au sein de l'équipe ICAR du LIRMM, qui est une équipe riche de sa diversité offrant un cadre idéal pour se réaliser pleinement. Je tiens à remercier donc, pour tous les conseils qu'ils m'ont prodigués, ses permanents que j'ai eu le plaisir de côtoyer. Je remercie aussi tous mes camarades thésards que j'ai eu la chance de connaître. Je garderai des instants inoubliables de nos sorties de 13 juillet et de nos journées café croissant.

Je tiens à adresser un remerciement particulier à Abderrahmene la personne, qui était la plus proche à mon cœur ces 5 années que j'ai passé en France, et qui m'a vraiment soutenu pour achever ce travail. Je tiens également à m'excuser auprès de lui car je sais que je n'étais pas toujours facile à vivre surtout durant les derniers mois de thèse.

Je n'oublie pas de dédier un remerciement particulier à tata Assia que je considère comme une seconde mère ici en France et qui m'a accueillie chaleureusement au sein de sa petite famille. UN GRAND MERCI.

Finalement, 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.



Table des matières

Remerciements	i
Table des matières	iii
1 Introduction	1
1.1 La stéganographie des média empiriques	2
1.2 Positionnement de la thèse	3
1.3 Plan du manuscrit	3
1.4 Symboles et notations mathématiques	4
I État de l’art	7
2 La Stéganographie	9
2.1 Contexte de la steganographie	10
2.1.1 La stéganographie au cours des siècles	10
2.1.2 La stéganographie de nos jours	13
2.2 La stéganographie moderne appliquée aux images numériques	14
2.3 Philosophies de conception d’un schéma stéganographique	16
2.3.1 La stéganographie par sélection du médium de couverture	16
2.3.2 La stéganographie par synthèse du médium de couverture	16
2.3.3 La stéganographie par modification du médium de couverture	17
2.4 Propriétés d’un schéma stéganographiques	18
2.4.1 Sécurité d’un schéma stéganographique	19
2.4.2 Capacité d’insertion	20

2.4.3	Capacité stéganographique	21
2.4.4	L'efficacité d'insertion	23
2.5	Méthodes de stéganographie usuelles	24
2.5.1	Insertion dans le domaine spatial	24
2.5.2	Insertion dans le domaine transformé	27
2.6	Méthodes de stéganographie adaptatives	28
2.6.1	l'impact d'insertion (la distortion)	29
2.6.2	Le problème de minimisation d'impact d'insertion	31
2.6.3	La carte de détectabilité	32
2.7	Synthèse	35
3	Les codes correcteurs d'erreurs en stéganographie	37
3.1	Utilité des codes correcteurs d'erreurs	38
3.2	Les codes correcteurs linéaires	39
3.2.1	Définition d'un code linéaire	39
3.2.2	Distance minimale d'un code linéaire	39
3.2.3	Matrice génératrice d'un code linéaire	39
3.2.4	Matrice de contrôle d'un code linéaire	40
3.3	Stéganographie et codes correcteurs	41
3.3.1	Technique de matrix embedding	41
3.3.2	Les codes à papier mouillé	44
3.3.3	Les codes STC	46
3.4	Synthèse	48
4	La Stéganalyse	49
4.1	Attaque d'un schéma de stéganographie	50
4.2	Les principaux scénarios de la stéganalyse	51
4.2.1	Stéganalyse à clairvoyance	51
4.2.2	Stéganalyse à payload inconnu	52
4.2.3	Stéganalyse universelle	52
4.2.4	Stéganalyse avec cover-source mismatch	53
4.2.5	Stéganalyse par mise en commun	53
4.3	Analyse ciblée d'un schéma stéganographique	54
4.3.1	Caractéristiques utilisées	55
4.3.2	Quelques méthodes d'analyse ciblée	55
4.4	Analyse aveugle d'un schéma stéganographique	59
4.4.1	Caractéristiques utilisées	61
4.4.2	Quelques outils pour la classification	63
4.5	La stéganalyse sous d'autres angles	70
4.6	Synthèse	72

II Contributions	73
5 La Stéganographie Adaptative par Oracle (ASO)	75
5.1 Motivation	76
5.2 Vue d'ensemble de la méthode ASO	78
5.3 Calcul de la carte de détectabilité	79
5.3.1 Aspect théorique	79
5.3.2 Aspect pratique	83
5.4 Insertion du message secret	84
5.5 Détails techniques de l'implémentation de ASO	85
5.6 Tests et résultats	86
5.6.1 Évaluation de la sécurité de ASO	86
5.6.2 Évaluation du processus itératif de ASO	90
5.6.3 Analyse de la carte de détectabilité ASO	92
5.7 Conclusion	96
6 Le paradigme de stéganographie par base	97
6.1 Le paradigme de la stéganographie par base	98
6.2 Mesures de sélection en stéganographie	98
6.3 Mesure de sécurité basée oracle pour la sélection	101
6.4 Tests et résultats	102
6.5 Discussion et conclusion	106
7 Conclusions et perspectives	107
7.1 Résumé des contributions	108
7.2 Perspectives	109
Liste des publications	113
Bibliographie	114
Table des figures	127
Liste des tableaux	128

Introduction

Soignez le commencement,
pensez à la fin, la fin viendra sans
fatigue. Si vous oubliez le but,
vous succomberez avant la fin.

CHOU KING - *Philosophe chinois*

Préambule

Le besoin de communication secrète ou discrète n'est pas une quête nouvelle : depuis l'antiquité, l'être humain, de part sa nature méfiante, a toujours chercher à protéger et à dissimuler ses données avec différentes méthodes. Avec l'avènement d'Internet, des méthodes numériques adaptées ont alors été mises en places. Dans ce manuscrit, nous allons nous intéresser plus particulièrement à la stéganographie qui est un procédé de communication secrète, et à sa discipline duale la stéganalyse. Dans ce chapitre d'introduction, nous commençons d'abord par détailler le problème de stéganographie dans les média empiriques (section 1.1). Ensuite, nous positionnons nos travaux de thèse, par rapport à l'état de l'art actuel dans ce domaine (section 1.2). Enfin, nous présentons le plan de ce manuscrit, ainsi que les différentes annotations mathématiques utilisées tout au long de ce document (sections 1.3 et 1.4).

Sommaire

1.1	La stéganographie des média empiriques	2
1.2	Positionnement de la thèse	3
1.3	Plan du manuscrit	3
1.4	Symboles et notations mathématiques	4

1.1 La stéganographie des média empiriques

Le problème d'échange de données secrètes a toujours existé, et ce depuis la naissance des grandes civilisations. Bien que la cryptographie offre un moyen efficace pour protéger les données secrètes en les rendant inintelligible aux yeux de ceux qui n'ont pas les droits nécessaires, dans la plupart des cas, le simple fait de communiquer avec des messages chiffrés peut attirer l'attention. Cela peut être problématique lorsqu'il s'agit d'un canal de communication monitoré par une tierce personne, qui peut stopper la communication entre les deux parties, au moindre soupçon. Dans un tel cas de figure, il est important du côté de l'émetteur et du récepteur, de cacher le fait même qu'il y a communication [Simmons, 1983]. Pour ce genre de scénario, la stéganographie représente une meilleure alternative que la cryptographie. La stéganographie, ou la science de communication secrète, est un procédé permettant de cacher un message secret au sein d'un document hôte anodin, de tel sorte à rendre le processus de dissimulation indétectable. Autrement dit, l'objectif est de rendre difficile, ou impossible, la distinction entre un document originale et un document modifié comportant un message secret. De manière analogue à la cryptographie, dont la discipline duale est la cryptanalyse visant à décrypter le message chiffré, la stéganographie a également comme discipline duale la stéganalyse. L'objectif de la stéganalyse est de détecter la présence d'un message caché. Ainsi, en stéganalyse le but principal n'est pas d'extraire le message caché, mais plutôt de détecter sa présence.

Le concept clé de la sécurité d'un système de stéganographie est donc son indétectabilité visuelle mais aussi et surtout statistique. En d'autres termes, lors du processus de dissimulation, la distribution du *stégo-objet*, contenant le message secret, doit être gardée la plus proche possible de la distribution originale. Pour atteindre cet objectif, le modèle de la source de couverture, employé pour la dissimulation, joue un rôle important. En effet, pour permettre une communication secrète qui n'est pas trivialement détectable, et afin de garantir un certain niveau de sécurité, il est nécessaire que la distribution *cover* soit proche ou même confondue à celle *stégo* [Cachin, 1998].

De nos jours, avec le développement d'Internet, et l'explosion des média numériques partagés sur les réseaux (images, sons et vidéos, ...), la dissimulation d'informations secrètes devient une pratique populaire et accessible à toute personne souhaitant communiquer de façon discrète. La communauté scientifique s'est alors particulièrement intéressée à cette discipline. Les chercheurs ont mis en évidence que la stéganographie appliquée aux média numériques actuels représente un véritable challenge faisant appel à de nombreuses disciplines : mathématiques, statistiques, traitement du signal, théorie de l'information, fouille de données, et théorie des jeux... Par ailleurs, les média numériques de nature empirique, variée et complexe, représentent des supports idéals pour la stéganographie [Fridrich, 2009].

Dans ce manuscrit, nous nous sommes intéressés également à la stéganographie, et plus particulièrement à la dissimulation d'informations secrètes dans les images numériques naturelles.

1.2 Positionnement de la thèse

Ces dernières années, le domaine scientifique a connu une explosion de publications en méthodes de dissimulation d'informations. Parmi les différentes méthodes de stéganographies, on trouve les algorithmes d'insertion adaptatifs, dont le principe consiste à modifier le médium hôte, de façon intelligente, pour pouvoir insérer le message secret de manière sécurisée. En d'autres termes, trouver le moyen d'insérer le message, de façon à garder la distribution du médium stéganographié la plus proche possible de la distribution originale. Pour ce faire, les méthodes adaptatives actuelles utilisent des cartes de détectabilité qui reflètent le niveau de sécurité de chaque élément de couverture. Ces méthodes permettent d'être plus efficaces en terme de sécurité, que les méthodes précédentes.

Dans le cadre de nos travaux de thèse, pour répondre à la tendance actuelle et au besoin de communication secrète, nous nous sommes intéressés aux différentes méthodes de stéganographie par modification de médium de couverture, plus particulièrement aux algorithmes de dissimulation adaptatifs.

L'objectif principal de cette thèse, est donc d'étudier et comprendre le mécanisme de fonctionnement de ces méthodes d'insertion adaptatives, afin de pouvoir mettre en place un nouveau schéma adaptatif pour la dissimulation d'informations dans les images numériques naturelles, encore plus performant et plus sûre que les méthodes existantes.

1.3 Plan du manuscrit

Ce manuscrit est composé de deux parties : une première partie état de l'art, où sont exposées les différentes notions de base (chapitres 2 à 4), puis une deuxième partie contributions, présentant le schéma d'insertion adaptatif par oracle (chapitres 5 à 6).

Le chapitre 2 expose la stéganographie sous divers angles, et présente les différentes méthodes d'insertion existantes dans l'état de l'art.

Le chapitre 3, présente un bref rappel des concepts clés de l'utilisation des codes correcteurs d'erreurs dans le cadre de la stéganographie.

Le chapitre 4 est dédié à la stéganalyse. Nous passons en revue les différentes méthodes d'attaques d'un schéma stéganographique de l'état de l'art.

Dans le chapitre 5, nous présentons le schéma d'insertion adaptatif par oracle (ASO) que nous avons proposé, ainsi que les différents résultats expérimentaux.

Dans le chapitre 6, nous introduisons les nouveaux concepts qui découlent de notre proposition. Nous présentons et étudions en particulier le concept de stéganographie par base, ainsi qu'une nouvelle mesure de sécurité pour la sélection des images une fois stéganographiées.

Dans le chapitre 7, nous clôturons ce manuscrit en présentant une conclusion générale à ces travaux, ainsi que les différentes perspectives envisagées.

1.4 Symboles et notations mathématiques

Afin de faciliter la compréhension, et pour éviter toute confusion, nous fixons dans cette section les différentes notations que nous utiliserons tout au long de ce manuscrit.

Pour commencer, nous utiliserons la police calligraphique pour la notation des ensembles. Les matrices et vecteurs seront identifiés par des caractères en gras respectivement en majuscule et minuscule.

Nous définirons par \mathcal{C} l'ensemble des médiums de couverture¹ de distribution $P_{\mathcal{C}}$, et \mathcal{S} l'ensemble des stégo-médiums² de distribution $P_{\mathcal{S}}$. Nous noterons par \mathbb{N} , et \mathbb{R} l'ensemble des nombres naturels et réels.

Les symboles $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{C} \triangleq \mathcal{I}^n$ ou $\mathbf{X} = (X_{i,j}) \triangleq \mathcal{I}^{n_1 \times n_2}$ désigneront de manière équivalente, l'image de couverture composée de $n = n_1 \times n_2$ éléments. Pour une image de couverture en niveau de gris, nous avons $\mathcal{I} = \{0, \dots, 255\}$.

Nous dénoterons par $\mathbf{y} = (y_1, \dots, y_n) \in \mathcal{S} \subset \mathcal{C} \triangleq \mathcal{I}^n$ ou $\mathbf{Y} = (Y_{i,j}) \triangleq \mathcal{I}^{n_1 \times n_2}$ l'image stégo composée de $(n = n_1 \times n_2)$ éléments. Dans le cas d'une image en niveau de gris, nous aurons $\mathcal{I} = \{0, \dots, 255\}$.

Nous noterons par $\mathbf{y}_{\mathbf{x} \sim x_i}$ l'image stégo obtenue après modification du $i^{\text{ème}}$ élément x_i de l'image de couverture \mathbf{x} .

Nous désignerons exclusivement par l'annotation $\mathbf{m} = (m_1, \dots, m_m) \in \{0, 1\}^m$ le message secret à m bits que l'on souhaite dissimuler pour le transmettre secrètement.

La fonction $D : \mathcal{C} \times \mathcal{S} \rightarrow \mathbb{R}$ désignera une fonction de distorsion, i.e., $D(\mathbf{x}, \mathbf{y})$ dénotera la distorsion entre l'image de couverture $\mathbf{x} \in \mathcal{I}^n$ et l'image stégo $\mathbf{y} \in \mathcal{I}^n$.

Nous dénoterons par $f(\mathbf{x})$ (resp. $f(\mathbf{y})$) la fonction qui retourne le vecteur caractéristique de l'image de couverture (resp. de l'image stégo) passé en paramètre.

Les symboles $\mathbf{f}_{\mathbf{x}}$, $\mathbf{f}_{\mathbf{y}}$ dénoteront respectivement le vecteur caractéristique de l'image de couverture \mathbf{x} , et le vecteur caractéristique de l'image séganographiée \mathbf{y} .

1. Le médium de couverture : il s'agit du support hôte, dans le quel seront dissimulées les informations secrètes. Il peut s'agir d'un texte, d'un son, d'une vidéo, ou d'une image dans notre cas.

2. Le stégo-médium ou le stégo-object : est le support obtenu après dissimulation des données secrètes.

Nous désignerons par $f_{x \sim x_i}^{(+)}$ (resp. $f_{x \sim x_i}^{(-)}$) le vecteur caractéristique de l'image x obtenu après modification $+1$ (resp. -1) du $i^{\text{ème}}$ pixel x_i .

Nous définirons par $\rho \in \mathbb{R}_+^n$ une carte de détectabilité associée l'image de couverture x . Cette carte de détectabilité attribue, à chaque élément de couverture x_i avec $i \in \{1, \dots, n\}$, un coût de détectabilité $\rho_i \in \mathbb{R}_+$ modélisant l'impact sur la sécurité dû à la modification de cet élément.

Nous noterons par $\rho_i^{(+)}$ (resp. $\rho_i^{(-)}$) la détectabilité après modification $+1$ (resp. -1) du pixel x_i .

Les termes P_E , P_{FP} , et P_{FN} désigneront respectivement la probabilité d'erreur, la probabilité de faux positif, et la probabilité de faux négatif. Ces notations seront utilisées afin de comparer la performance des différents algorithmes stéganographiques.

Les symboles E , V , et cov , désigneront respectivement l'espérance, la variance et la covariance mathématique.

Nous utiliserons $\log_2(x)$ pour dénoter le logarithme en base 2, et réserverons $\ln(x)$ pour désigner le logarithme naturel (népérien).

Enfin, nous noterons par $H(p) = -\sum_{i=1}^k p_i \log_2 p_i$ l'entropie au sens de Shannon d'une distribution de probabilités $p = (p_1, \dots, p_k) \in [0, 1]^k$.

Première partie

État de l'art

La Stéganographie

J'aurais voulu être espion, mais il fallait avaler des microfilms et mon médecin me l'a interdit.

WOODY ALLEN - RÉALISATEUR

Préambule

La stéganographie est l'art de la dissimulation d'un message secret. L'objectif est de cacher une information secrète au sein d'un support d'apparence anodine, de sorte que le message soit indétectable. De nos jours, avec la propagation d'Internet, et la généralisation des réseaux sociaux, la stéganographie a pris de l'ampleur dans les supports numériques (fichiers audio, vidéos ou images), qui représentent des supports privilégiés pour la transmission d'informations. Dans ce chapitre, nous présentons cette discipline, ainsi que les méthodes d'insertion de l'état de l'art. Tout d'abord, Nous introduisons le contexte général de la stéganographie (section 2.1), ainsi que les bases de la stéganographie moderne (section 2.2). Ensuite, nous présentons les différentes philosophies de dissimulation, en particulier les méthodes de stéganographie par modification (section 2.3). Nous définissons alors les caractéristiques d'un schéma stéganographique (section 2.4). Nous passons en revue quelques unes des techniques classiques de stéganographie (section 2.5). Enfin, nous nous intéressons plus particulièrement aux méthodes d'insertion adaptatives (section 2.6).

Sommaire

2.1	Contexte de la steganographie	10
2.2	La stéganographie moderne appliquée aux images numériques	14
2.3	Philosophies de conception d'un schéma stéganographique	16
2.4	Propriétés d'un schéma stéganographiques	18
2.5	Méthodes de stéganographie usuelles	24
2.6	Méthodes de stéganographie adaptatives	28
2.7	Synthèse	35

2.1 Contexte de la stéganographie

Dans cette section nous abordons le contexte général de l'utilisation de la stéganographie. Nous passons en revue le contexte historique : de la stéganographie *ancienne* (section 2.1.1), à la stéganographie *moderne* (section 2.1.2).

2.1.1 La stéganographie au cours des siècles

Tout comme la cryptographie, la stéganographie, ou l'art de communication secrète, est une discipline qui remonte à l'antiquité. Tandis que la première permet de communiquer secrètement, la seconde offre en plus la discrétion de la communication. En effet on peut dire que la stéganographie repose sur l'idée de la sécurité par obscurité : si personne ne sait qu'il y a un message caché à l'intérieur d'un support quelconque, personne ne cherchera à le regarder ou à le récupérer.

D'origine grecque, le mot stéganographie ("*stego*" : secret, et "*graphia*" : écriture) est apparu pour la première fois dans l'histoire vers 445 av J.-C, à travers les récits de Hérodote. Dans son œuvre *l'Enquête*¹ [Hérodote, 1985] [Hérodote, 1990], Hérodote rapporta l'histoire de Histiée, conseiller du roi de Perse, qui incita son gendre Aristagoras, le roi de Milet, à se rebeller contre les Perses vers 500 av J.-C. Pour ce faire, il fit raser la tête de son esclave, lui tatoua le message sur le crâne, puis attendit la repousse de ses cheveux avant de l'envoyer à Milet. Une fois l'esclave arrivé à destination, il n'eut qu'à se faire raser la tête une deuxième fois, pour transmettre le message secret.

Plus loin encore, dans ce même ouvrage, on retrouve aussi l'histoire de Démarate en 480 av J.-C, qui réussit à déjouer le plan de Xerxès, roi du perse, visant à envahir la Grèce. Démarate, ancien roi de Sparte, fut au courant du plan d'invasion de Xerxès. Il décida alors de prévenir les Grecques, ceci en leur envoyant un message gravé sur le bois d'une tablette d'écriture recouverte de cire, et donc d'apparence extérieure vierge.

Au fil des siècles qui ce sont écoulés depuis Hérodote, différentes formes de stéganographie, de plus en plus évoluées, ont été utilisées dans le monde :

En chine antique, on écrivait les messages secrets sur de très fins rubans de soie, qu'on enrobait ensuite dans des petites boules de cire. Ces boules, ensuite avalées par le messager, pouvaient voyager jusqu'au destinataire, d'une manière totalement discrète.

Plus subtile encore, l'invention de l'encre sympathique fut l'un des procédés stéganographiques le plus utilisé. Rapportée par le naturaliste Pline l'Ancien dès le 1^{er} siècle avant J.-C, l'encre sympathique, ou l'encre invisible, est un procédé chimique qui consiste à uti-

1. *l'Enquête* (intitulée aussi *Histoires*) : Chronique composée de 9 livres, portant chacun le nom d'une muse, et rapportant l'histoire et les accomplissements des civilisations anciennes (Perse, Grecque, égyptienne, ...).



 <p>Lettre de George Sand à Alfred de Musset</p> <p>Un texte peut caché un autre : En apparence il s'agit d'une simple lettre d'amour. En revanche, en lisant une phrase sur deux, on découvrira un texte caché plus coquin.</p>	<p>Cher ami, Je suis toute émue de vous dire que j'ai bien compris l'autre jour que vous aviez toujours une envie folle de me faire danser. Je garde le souvenir de votre baiser et je voudrais bien que ce soit une preuve que je puisse être aimée par vous. Je suis prête à montrer mon affection toute désintéressée et sans calcul, et si vous voulez me voir ainsi vous dévoiler, sans artifice, mon âme toute nue, daignez me faire visite, nous causerons et en amis franchement je vous prouverai que je suis la femme sincère, capable de vous offrir l'affection la plus profonde, comme la plus étroite amitié, en un mot : la meilleure épouse dont vous puissiez rêver. Puisque votre âme est libre, pensez que l'abandon où je vis est bien long, bien dur et souvent bien insupportable. Mon chagrin est trop gros. Accourez bien vite et venez me le faire oublier. À vous je veux me soumettre entièrement.</p> <p>Votre poupée</p>
 <p>Réponse de Alfred de Musset à George sand</p> <p>Un texte peut caché un autre (Acrostiche): Lisez seulement le premier mot de chaque ligne.</p>	<p>Quand je mets à vos pieds un éternel hommage Voulez-vous qu'un instant je change de visage ? Vous avez capturé les sentiments d'un cœur Que pour vous adorer forma le Créateur. Je vous chéris, amour, et ma plume en délire Couche sur le papier ce que je n'ose dire. Avec soin, de mes vers lisez les premiers mots Vous saurez quel remède apporter à mes maux.</p>

FIGURE 2.1 – Lettres de George Sand et Alfred de Musset.

liser du jus de citron, du lait, ou même du chlorate de soude, pour écrire le message secret qui sera invisible à l'œil nu. Un simple passage sous une source chaude (flamme de bougie, fer à repasser chaud...) ou un bain dans un réactif chimique, révèle le message. Cette technique reste encore présente, puisque on trouve aujourd'hui sur les billets de banque des encres ultraviolettes, qui créent une réaction, en cas de photocopie, en inscrivant un message permettant à lutter contre la contrefaçon.

Une autre forme de dissimulation de messages est la stéganographie linguistique. Pour cacher un message secret dans un texte, on peut utiliser le langage, l'espace entre les mots, la ponctuation, l'orthographe, ou encore des repères au niveau des caractères.

Parmi les méthodes de stéganographie linguistique on trouve l'acrostiche, qui n'est autre qu'un poème dont la première lettre de chaque vers compose un mot ou une phrase. L'exemple le plus célèbre se trouve sans doute dans le livre *Hypnerotomachia Poliphili* publié en 1499. Dans cet ouvrage, si on regroupe la première lettre de chaque chapitre, au

nombre de 38, on peut reconstruire la phrase suivante "*Poliam frater Franciscus Columna peramavit*", ce qui veut dire "*Frère Francesco Colonna aime Polia passionnément*".

Les utilisateurs de stéganographie linguistique, les plus connus, restent à nos jours George Sand (entre 1833 et 1834) et Alfred de Musset. La Figure 2.1 illustre un extrait de correspondance amoureuse attribuée à George Sand et supposée avoir été adressée à Alfred de Musset.

Durant la Seconde Guerre mondiale, on nota l'apparition de la technique du micro-point de Zapp. Très appréciée par les agents allemands, cette technique consiste à réduire une photo, d'une page en un point d'un millimètre ou moins, qui est ensuite placé discrètement dans du texte anodin. Cette technique a été très longtemps utilisée dans les billets de banque Suisse.



FIGURE 2.2 – Peinture artistique contenant l'anamorphose d'un crâne qui ne peut être vu qu'en regardant le tableau avec une vue rasante.

La stéganographie est aussi présente dans le domaine artistique. Parmi les applications artistiques possibles, on trouve *l'anamorphose* qui est une déformation réversible d'une image à l'aide d'un système optique (miroir, courbe ..) ou d'un procédé mathématiques. S'inspirant de la stéganographie, le procédé consiste à cacher une image dans une autre, et qui ne sera découverte que si l'image de couverture est penchée selon un certain angle. L'exemple le plus célèbre est le tableau de Hans Holbein, *les ambassadeurs* (Figure 2.2). Une autre application artistique, qui peut aussi faire appel à la stéganographie, est l'élaboration de tours de télépathie. Ainsi deux complices peuvent communiquer à l'insu de leur public, et prétendre que l'un d'entre eux a la capacité de lire dans les pensées de l'autre.

Plus subtil encore, on retrouve aussi la stéganographie par partition musicale. Introduite par le scientifique allemand Gaspar Schott (1608 - 1666), le principe consiste à coder le message secret en utilisant des notes de musique. Ainsi, comme illustré sur la Figure 2.3,

le message apparait comme une partition musicale toute à fait normale, et peut donc passer inaperçu. Toutefois, si on joue la partition, il y a peu de chance que la mélodie soit harmonieuse.

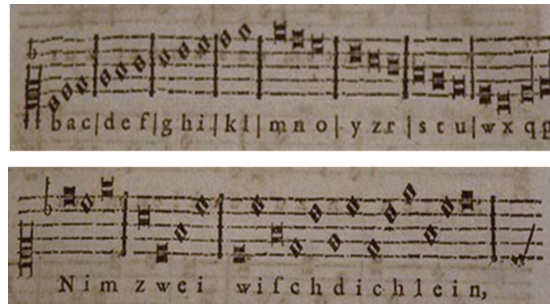


FIGURE 2.3 – Utilisation des partitions musicales pour la stéganographie.

Les lecteurs férus d'anecdotes historiques consacrées à la stéganographie trouveront satisfaction dans les ouvrages suivant [Kahn, 1996] [Judge, 2001] [Fridrich, 2009].

2.1.2 La stéganographie de nos jours

Contrairement à la stéganographie *ancienne*, la stéganographie *moderne*, ou *numérique*, est une science jeune, qui date seulement d'une quinzaine d'années. La stéganographie *moderne* passe par l'utilisation des supports numériques pour la transmission de données secrètes. L'essor d'Internet, et le développement des échanges électroniques via les réseaux sociaux a rendu très simple la dissimulation de messages secrets dans des supports comme : les fichiers audio, le texte, les images, les vidéos, les programmes, les sites internet. Les fichiers multimédia représentent des supports privilégiés pour l'échange de données. La stéganographie numérique constitue un excellent moyen pour la communication secrète. Elle est, en effet, très adaptée pour la dissimulation de données confidentielles. Dans certains pays non démocratiques où la liberté d'expression est réprimée, la stéganographie représente un excellent moyen pour communiquer librement dans des conditions de censure ou de surveillance. Toutefois, il arrive également que la stéganographie soit utilisée à des fins illicites. Certaines sources avancent l'hypothèse selon laquelle, la stéganographie a été utilisée pour de la pornographie infantile [Astrowsky, 2000] [Renold *et al.*, 2003], ainsi que pour des actes de terrorisme [MR, 2000]. Plusieurs acteurs des médias prétendent même que les attentats du 11 septembre, dirigés par Ben Laden en 2001, auraient été organisés en utilisant des messages secrets cachés au sein d'images numériques à caractère pornographique [AFP, 2001] [Kelley, 2001] [Sieberg, 2001]. Par ailleurs, la stéganographie intéresse également les hackers, qui peuvent l'utiliser par exemple pour le camouflage de code malveillant

fragmenté, qui sera par la suite rassemblé directement sur l'ordinateur de la victime. Finalement, la stéganographie est aussi adaptée pour l'espionnage et la fuite de données industrielles. Elle constitue un moyen efficace pour le vol d'informations confidentielles d'une manière discrète. En effet, il est très difficile de détecter ce genre de fuites au sein des entreprises.

Face à une telle prolifération et à un tel engouement pour la stéganographie numérique, il devient important d'étudier ces nouvelles voies de dissimulation de données. Le but étant 1) de comprendre et maîtriser le processus de dissimulation stéganographique, et 2) de développer par la suite des outils de stéganalyse performants qui permettent de dissuader les usages malveillants. Ces dernières années la communauté scientifique s'est donc particulièrement intéressée à cette jeune discipline. C'est d'ailleurs en 1996, lors de la première édition *d'Information Hiding*, que l'on peut situer la naissance de la première communauté stéganographique [Pfitzmann, 1996]. Depuis, le nombre de conférences scientifiques a augmenté, pour enfin se stabiliser ces cinq dernières années. Parmi les plus importantes conférences du domaine, on peut citer : WIFS², IH&MMSec³, SPIE EI⁴.

Aujourd'hui les messages secrets se transmettent de manière numérique avec des méthodologies plus rigoureuses. De nombreuses méthodes d'insertion sont apparues, ainsi qu'une meilleure formalisation de la stéganographie.

2.2 La stéganographie moderne appliquée aux images numériques

La stéganographie moderne consiste à exploiter les supports numériques actuels pour transmettre le message secret. La définition du problème peut s'expliquer par le *scénario des prisonniers*, posé par G.J. Simmons en 1983 [Simmons, 1983]. Soit Alice et Bob deux prisonniers enfermés dans deux cellules différentes, et souhaitant communiquer un message d'évasion. Comme dans toute prison, Alice et Bob ne sont autorisés à communiquer qu'à travers un intermédiaire. Eve est la gardienne chargée de la surveillance des échanges de message entre Alice et Bob. Si Eve suspecte le moindre signe de conspiration entre les deux détenus, elle s'autorisera à mettre fin à leur échange. Alice et Bob sont conscients de cette situation, et savent très bien que l'utilisation de messages chiffrés éveillerait les soupçons de Eve. Ils doivent donc utiliser une technique de dissimulation pour cacher leur plan dans

2. WIFS : IEEE International Workshop on Information Forensics and Security.

3. IH&MMSec : Information Hiding, et ACM Multimedia and Security workshop, deux importantes conférences qui ont fusionné en 2013.

4. SPIE EI : IS&T/SPIE Electronic Imaging.

des messages innocents. Ils pourront ainsi planifier leur évasion, sans attirer la suspicion de Eve.

Dans ce scénario on distingue deux parties : les stéganographes représentés par les deux prisonniers Alice et Bob, et la stéganalyste modélisée par la gardienne Eve. Du côté stéganographie, Alice et Bob ont pour but de se communiquer discrètement des informations secrètes de manière totalement indétectable. Du côté stéganalyse, Eve la gardienne est libre d'examiner le médium intercepté de manière passive, active ou malicieuse. Une inspection passive consiste simplement à décider si le médium intercepté contient ou non un message secret. Lorsque Eve conclut qu'il y a présence d'un message caché, elle coupe la communication. Dans le cas inverse, elle laisse passer le médium vers son destinataire. Une inspection active consiste à altérer suffisamment le médium intercepté, pour en conserver que le contenu perceptible et empêcher la lecture d'un éventuel message caché. Enfin, une inspection malicieuse consiste à comprendre la technique stéganographique et extraire le message caché, et pourquoi pas réintroduire un message falsifié.

Dans ce manuscrit nous nous intéressons uniquement à la stéganalyse passive qui, comme son nom l'indique, ne s'autorise pas la modification du médium intercepté lors de l'analyse. L'objectif principale est alors de détecter la présence ou non d'informations dissimulées.

La stéganographie moderne est potentiellement applicable à différents supports numériques : fichiers audio, vidéos, textes, ...etc. Parmi les fichiers qui sont très adaptés pour la dissimulation d'information, on retrouve également les images numériques. Ce type de fichier étant très couramment échangé sur Internet, une grande majorité des travaux de recherches lui sont consacrés. Dans ce manuscrit nous nous intéressons également à ce type de fichier. Nous nous focalisons principalement sur les schéma stéganographiques utilisant les images numériques naturelles comme support de transmission, c'est à dire celles qui sont acquises à l'aide d'un appareil d'enregistrement de type appareil photographique ou scanner. Elles peuvent être de nature compressée ou non-compressée. Formellement, nous noterons par $\mathbf{x} = (x_1, \dots, x_n) \triangleq \mathcal{I}^n$ une image de couverture numérique composé d'une succession de n échantillons. Respectivement, le symbole $\mathbf{y} = (y_1, \dots, y_n) \triangleq \mathcal{I}^n$ désignera l'image obtenue après dissimulation du message secret (voir symboles et annotations dans la section 1.4). De manière générale les échantillons composant une image numérique naturelle appartiennent à \mathbb{R} . Dans le cas d'une image en niveaux de gris, les échantillons sont des pixels, avec $\mathcal{I} = \{0, \dots, 255\}$, et forment un matrice de taille $n = n_1 \times n_2$. Dans le cas d'une image en couleur, nous avons $\mathcal{I} = \{0, \dots, 255\}^3$ codé généralement avec trois canaux de couleurs : rouge, vert et bleu.

Les travaux présentés dans ce document considèrent uniquement les images numérique en niveau de gris, c'est à dire celles codées uniquement sur un seul canal.

2.3 Philosophies de conception d'un schéma stéganographique

En stéganographie, il existe trois philosophies principales pour l'insertion d'un message secret [Fridrich, 2009] : la stéganographie par sélection du médium de couverture (section 2.3.1), la stéganographie par synthèse du médium hôte (section 2.3.2), et enfin la stéganographie par modification d'un médium de couverture déjà existant (section 2.3.3). Dans cette section, nous présentons le principe de chacune de ces trois philosophies d'insertion.

2.3.1 La stéganographie par sélection du médium de couverture

En stéganographie par sélection du médium de couverture, Alice, l'émetteur, dispose au préalable d'une base fixe d'images. Pour communiquer secrètement avec Bob, Elle sélectionne, à partir de sa base, l'image qui communique au mieux le message désiré. Par exemple, Alice peut transmettre à Bob un bit d'information, simplement en jouant sur l'orientation de l'image envoyée (portrait ou paysage). De même, la présence d'un animal ou d'un objet particulier, dans l'image envoyée, peut avoir un sens caché, partagé uniquement entre l'émetteur et le récepteur. Par ailleurs, Alice peut également utiliser une fonction de hachage, avec une clé secrète commune entre elle et Bob, pour transmettre son message. Dans un tel cas de figure, Alice parcourt sa base d'images, jusqu'à ce qu'elle tombe sur une image, dont l'empreinte digitale coïncide avec le message désiré. Une fois trouvée elle envoie cette image à Bob, qui pourra facilement lire le message secret en ré-applicant la fonction de hachage avec sa clé secrète. Bien évidemment, cette dernière méthode de dissimulation devient très vite impraticable dans la réalité. En effet, plus le message est long, plus le nombre d'images à parcourir est important.

L'avantage de ce genre d'approches est qu'elles sont quasiment indétectables. En effet, le médium de couverture n'ayant subi aucune modification, il est impossible de deviner qu'il y a un message caché. Le problème majeur de ces méthodes reste cependant celui de la capacité d'insertion très limitée [Fridrich, 2009].

2.3.2 La stéganographie par synthèse du médium de couverture

La stéganographie par synthèse du médium de couverture consiste à créer le support hôte qui embarque au mieux le message secret. En théorie, si Alice est capable de créer un médium de couverture, avec un distribution connue entre elle et Bob, elle devrait pouvoir cacher son message de manière parfaitement sûre. En d'autres termes, Alice pourra dissimuler son message secret tout en préservant parfaitement la distribution originale.

Alice peut par exemple prendre plusieurs images de la même scène. Pour envoyer un message secret, Alice crée une nouvelle stégo image en échantillonnant simplement les

différentes images acquises. En stéganographie linguistique, on retrouve également ce concept de dissimulation au travers de l'acrostiche (voir les correspondances de George Sand à Alfred de Musset sur la Figure 2.1). Le lecteur intéressé par les différentes méthodes de la stéganographie par synthèse trouvera sûrement intéressantes les références suivantes [Fridrich, 2009] [Wang et Moulin, 2008].

L'inconvénient majeur de ces méthodes de dissimulation est qu'elles sont généralement très théoriques, et complexes à mettre en œuvre. Par ailleurs, elles sont également très limitées en capacité d'insertion.

2.3.3 La stéganographie par modification du médium de couverture

Le stéganographie par modification du médium de couverture est la méthode d'insertion la plus pratique et la plus utilisée dans la littérature. Le principe de cette méthode consiste à altérer un médium de couverture (déjà existant), pour dissimuler un message, de sorte que celui ci soit *indélectable visuellement et statistiquement*. Autrement dit, le message est inséré en modifiant le support, de manière à préserver "le plus possible" la statistique originale de ce support.

Soit \mathcal{K} l'ensemble des clés possibles, \mathcal{M} l'ensemble des messages possibles insérables, et \mathcal{C} l'ensemble des supports.

Formellement, un schéma stéganographique par modification du médium de couverture est caractérisé par deux fonctions :

- Une fonction d'insertion, notée Emb , utilisée par l'émetteur, Alice, et qui prend en entrée une clé privée $\mathbf{k} \in \mathcal{K}$, un message à dissimuler $\mathbf{m} \in \mathcal{M}$, et un médium de couverture $\mathbf{c} \in \mathcal{C}$ appelé le *cover-médium*, et qui retourne en sortie un nouveau élément de \mathcal{C} , contenant le message secret, appelé le *stégo-médium*.

$$\text{Emb} : \mathcal{C} \times \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}. \quad (2.1)$$

- Une fonction d'extraction, notée Ext , utilisée par le récepteur, Bob, et qui prend en paramètre une clé stéganographique, et le stégo-médium reçu, et qui retourne le message secret en sortie.

$$\text{Ext} : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}. \quad (2.2)$$

L'objectif d'un schéma stéganographique est qu'après qu'Alice ait inséré le message, Bob puisse l'extraire. Cela se traduit par :

$$\text{Ext}(\text{Emb}(\mathbf{c}, \mathbf{m}, \mathbf{k}), \mathbf{k}) = \mathbf{m}, \quad \forall (\mathbf{c}, \mathbf{m}, \mathbf{k}) \in \mathcal{C} \times \mathcal{M} \times \mathcal{K}. \quad (2.3)$$

De manière similaire à la cryptographie, il existe deux types d'algorithmes stéganographiques par modification : à clé privée et à clé publique. Dans le cas d'un schéma de stéganographie à clé privée, l'émetteur et le récepteur (Alice et Bob), doivent partager au préalable une clé secrète commune. Cette clé est alors utilisée par la suite pour l'insertion et

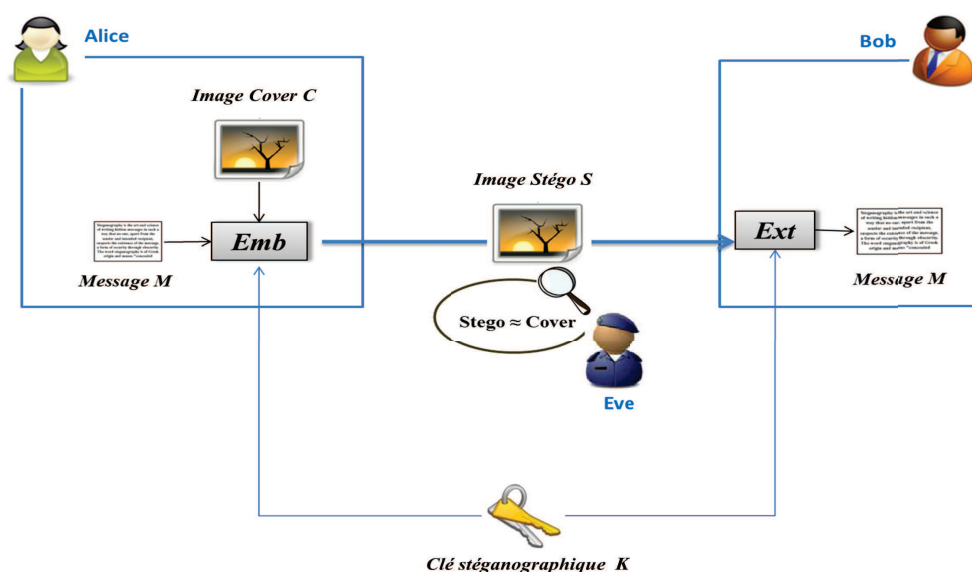


FIGURE 2.4 – Étapes de dissimulation d’un message secret pour un schéma stéganographique par modification.

l’extraction du message secret. Dans le cas d’un schéma de stéganographie à clé publique, pour des raisons de simplification, on supposera que l’émetteur, Alice, a accès à un annuaire de clés publiques stéganographiques. Ainsi, lors de la transmission, Alice utilisera la clé publique de Bob pour l’insertion du message secret, que seul Bob pourra le déchiffrer par la suite avec sa clé privée.

Dans ce manuscrit nous considérons uniquement les méthodes de stéganographie par modification de couverture. La Figure 2.4 illustre les étapes de dissimulation d’un message secret dans une image numérique.

2.4 Propriétés d’un schéma stéganographiques

Comme mentionné précédemment (section 2.3.3), l’objectif d’un schéma de stéganographie par modification est de dissimuler un message secret au sein d’un support hôte, de sorte qu’il soit *indélectable*. La difficulté majeure est donc de trouver le moyen de modifier le médium de couverture hôte, pour que le stégo-médium soit le plus identique possible au médium de couverture. Pour cela, il est important de définir en premier lieu la notion d’indélectabilité, ainsi que les propriétés caractérisantes d’un schéma stéganographique, qui nous permettront d’évaluer l’efficacité et la sécurité du processus d’insertion.

2.4.1 Sécurité d'un schéma stéganographique

La sécurité du point de vue théorique

En stéganographie, la notion de sécurité ou d'indéfectabilité d'un schéma se réfère à la résistance du processus de dissimulation face à une attaque au sens stéganographique⁵. Autrement dit, la sécurité d'un schéma mesure la capacité ou l'incapacité de l'attaquant (la gardienne Eve) de détecter la présence même du message secret au sein du support hôte. Ce critère a été introduit, pour la première fois en stéganographie, par [Cachin, 1998].

Soit \mathcal{C} l'ensemble des médiums de couverture de distribution $P_{\mathcal{C}}$, et \mathcal{S} l'ensemble des stégo-médiums de distribution $P_{\mathcal{S}}$. Afin de définir la sécurité d'un schéma stéganographique, C. Cachin, utilise l'entropie relative $D_{\text{KL}}(P_{\mathcal{C}}\|P_{\mathcal{S}})$ entre les deux distributions $P_{\mathcal{C}}$ et $P_{\mathcal{S}}$. Cette pseudo-distance, qui n'est pas une distance au sens mathématique, est appelée *distance de Kullbak-Liebler* (D_{KL}) et définie par :

$$D_{\text{KL}}(P_{\mathcal{C}}\|P_{\mathcal{S}}) = \sum_{\mathbf{x} \in \mathcal{C}} P_{\mathcal{C}}(\mathbf{x}) \log_2 \frac{P_{\mathcal{C}}(\mathbf{x})}{P_{\mathcal{S}}(\mathbf{x})}. \quad (2.4)$$

Cette formulation du problème consiste à dire que la sécurité d'un schéma stéganographique dépend de l'incapacité de l'adversaire à distinguer entre les deux distributions $P_{\mathcal{C}}$ et $P_{\mathcal{S}}$. Ainsi, un schéma de stéganographie est considéré comme étant parfaitement sûr si $D_{\text{KL}}(P_{\mathcal{C}}\|P_{\mathcal{S}}) = 0$, et comme ε -sûr si $D_{\text{KL}}(P_{\mathcal{C}}\|P_{\mathcal{S}}) < \varepsilon$. Plus ε est grand, plus la probabilité qu'un message secret soit détecté est grande. L'utilisation d'autres distances du même genre est également possible [Hopper, 2004].

À ce stade, nous pouvons aisément constater que cette formulation de la sécurité suppose que l'on sache définir précisément ce que c'est une distribution. En pratique, un attaquant, dont la puissance (les ressources matérielles, la capacité de calcul, le temps de calcul) est limitée, ne dispose que d'une approximation de ces distributions. De ce fait, et pour évaluer la sécurité d'un schéma face aux différentes formes d'attaques, auquel il doit résister, plusieurs modèles de sécurité ont été proposés dans la littérature. Certains modèles sont consacrés aux schémas stéganographique à clés privées [Hopper *et al.*, 2002] [Cachin, 2004], d'autres aux schémas à clés publiques [Le et Kurosawa, 2003] [Ahn et Hopper, 2004]. Par ailleurs, chaque modèle de sécurité est associé à un type particulier d'adversaire (passif, actif ou malicieux). Plusieurs modèles sont dédiés aux adversaires passifs [Le et Kurosawa, 2003] [Cachin, 2004] [Ahn et Hopper, 2004], d'autres aux adversaires actifs [Hopper, 2005] [Cachin, 2011]. Des travaux récents ont même adapté les modèles de sécurité aux adversaires réalistes. Ils ont défini formellement les modèles

5. Rappelons que dans ce manuscrit, nous considérons uniquement les attaques passives. L'objectif d'une attaque stéganographique, dans notre cas, est donc de détecter simplement la présence ou non d'un message secret.

de sécurité qui sont adaptés à la stéganalyse actuelle (spécifique et universelle) utilisant les outils d'apprentissage et de classification, mise en œuvre dans la littérature [Ker, 2007b] [Barbier et Alt, 2008] [Barbier *et al.*, 2009].

Pour conclure, on peut dire que la formalisation théorique de la sécurité d'un schéma stéganographique est encore un problème ouvert à la recherche. Pour les images naturelles de nature complexe, et dont la distribution statistique est difficilement modélisable (inconnue), la définition ainsi que formalisation de la sécurité représente encore un véritable challenge.

La sécurité du point de vue pratique

En pratique, compte tenu de l'absence d'une formalisation précise de la notion de sécurité, il est difficile pour un stéganographe de garantir la sécurité d'un schéma stéganographique face à toutes les attaques existantes. Toutefois, il est possible d'éviter de mettre en défaut le processus de dissimulation par des attaques triviales, ceci en respectant quelques règles de base. Parmi ces règles, il est important d'abord de s'assurer que le support de couverture est utilisé une seule fois, et qu'il est détruit dès son utilisation, afin d'éviter toutes les attaques par différence. En effet, si le support original tombe dans les mains d'une tierce personne la probabilité de détection du stégo-support sera égale à 1. Par ailleurs, afin de se prémunir contre les attaques exhaustives sur la clé stéganographique, il faut vérifier que la taille de la clé est suffisamment grande. De plus, pour éviter les attaques visuelles, l'émetteur doit s'assurer que le processus de dissimulation du schéma stéganographique soit imperceptible à l'œil nu. Autrement dit, il ne doit pas altérer visuellement le support hôte. De même, pour prévenir des attaques statistiques classiques, tel que le test du χ^2 , il est essentiel de préserver au mieux la distribution et la statistique des éléments de couverture à modifier, à l'ordre 1 et supérieur. Enfin, pour éviter les attaques utilisant l'image résidu de bruit pour l'estimation de la localisation de l'insertion, il est important également de vérifier que les endroits d'insertion de l'algorithme sont différents d'un médium à un autre.

2.4.2 Capacité d'insertion

La capacité d'insertion (ou *payload*) est le nombre de bits significatifs qui peuvent être dissimulés dans un support numérique hôte [Fridrich, 2009].

Pour un système stéganographique avec \mathcal{M} l'ensemble des messages possibles, la capacité d'insertion est donnée par :

$$\log_2 |\mathcal{M}(\mathbf{x})|, \quad (2.5)$$

où $|\mathcal{M}|$ est le nombre de messages possibles, et $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{C}$ le médium de couverture hôte composé de n éléments.

De la même façon, on peut également définir la capacité d'insertion relative par :

$$\frac{\log_2 |\mathcal{M}(\mathbf{x})|}{n}. \quad (2.6)$$

Ce n'est rien d'autre que le rapport entre la taille du message à cacher (la capacité d'insertion) et la taille du médium utilisé. Ce rapport peut être exprimé en nombre de bits de message insérés par pixel (bpp) quand il s'agit d'une image numérique dans le domaine spatial, ou en nombre de bits de message insérés par chaque coefficient non-nul (bpnc) lorsqu'il s'agit d'une insertion dans les coefficients quantifiés d'une image JPEG. Par la suite, tout au long de ce document, nous utiliserons ce rapport, afin de pouvoir comparer la quantité d'information cachées entre les différents algorithmes stéganographiques.

Notons que la capacité d'insertion est une mesure qui dépend fortement de l'algorithme d'insertion, mais aussi du médium de couverture. Par exemple, si nous prenons une image PGM⁶ à niveaux de gris de taille 512×512 , et en insérant un bit de message par pixel, nous obtenons alors $\mathcal{M} = \{0, 1\}^{512 \times 512}$ et $|\mathcal{M}(\mathbf{x})| = 2^{512 \times 512}$ messages possibles. En revanche, si nous utilisons une image JPEG⁷ en niveaux de gris de taille 512×512 avec un facteur de qualité $q_f = 75$, pour insérer un bit de message par coefficient DCT⁸ quantifié non-nul, alors la capacité d'insertion dépendra directement du médium de couverture utilisé et non de son format. En JPEG le nombre de coefficients non-nuls dépend du contenu de l'image, ce qui implique une capacité d'insertion différente d'une image à une autre.

Le but de la stéganographie est de cacher l'existence même du message secret en le rendant *indétectable*. Plus la capacité d'insertion est grande, plus le risque de détectabilité est fort. En stéganographie d'image on cherchera donc à maintenir un certain compromis entre "la capacité d'insertion" et "l'indétectabilité", afin de construire un schéma de stéganographie ε -sûr avec un ε suffisamment petit. Par ailleurs, en stéganographie avec gardien passif, on considère que si le message secret est altéré au cours de la transmission, il sera retransmis. L'objectif est donc de transmettre le maximum d'information secrète sans attirer la suspicion.

2.4.3 Capacité stéganographique

La capacité stéganographique est le nombre maximale de bits qui peuvent être insérés dans le médium de couverture, de sorte que la probabilité de détection soit insignifiante. C'est une mesure qui est strictement plus petite que la capacité d'insertion [Fridrich, 2009].

6. PGM : Portable Grey Map.

7. JPEG : Joint Photographic Experts Group.

8. DCT : Discrete Cosine Transform.

Par définition, la capacité stéganographique est peut être l'un des plus importants concepts de la stéganographie. Hélas, déterminer la quantité d'information, qui peut être cachée dans le support hôte, d'une manière totalement sûre, est une tâche extrêmement difficile, et ce pour le plus simple des algorithmes stéganographiques. La raison principale, est le manque de modèles statistiques précis représentant les images numériques naturelles. Par ailleurs, un grand dilemme se pose : faut-il définir la capacité stéganographique uniquement en fonction de l'image de couverture choisie, ou aussi en fonction de l'algorithme d'insertion utilisé.

La capacité stéganographique est un concept qui découle de la théorie de l'information. Pour un schéma stéganographique parfaitement sûr⁹, et où la distribution du médium de couverture est connue, [Comesana et Pérez-González, 2007] [Wang et Moulin, 2008] définissent la capacité stéganographie comme étant égale à l'entropie (au sens de Shannon) de la source de couverture. Bien évidemment, plus la taille du support hôte est grande, plus l'entropie (et donc la capacité stéganographique) du médium est grande aussi. De cette définition, nous pouvons conclure que la capacité stéganographique est en relation linéaire avec la taille du médium de couverture.

Compte-tenu de l'absence de schéma stéganographique parfaitement sûr pour les images naturelles, dont la distribution est partiellement ou totalement inconnue, le concept de la capacité stéganographique a été revisité pour un tel cas de figure. Pour un schéma stéganographique ε -sûr, et ce pour la première fois, [Ker, 2007a] introduit alors la loi de la "racine carrée de la capacité stéganographique" (Théorème 1). Pour cela, il définit d'abord la notion de risque pour un stéganographe (Définition 1) :

Définition 1 (Notion de risque pour un stéganographe [Ker, 2007a]). *Soit (P_{FP}^*, P_{FN}^*) un couple fixe de probabilités définissant le risque acceptable pour un stéganographe, et correspondant respectivement aux probabilités d'erreur de faux positif et faux négatif acceptables, avec $0 < P_{FP}^* < 1$ et $0 < P_{FN}^* < 1 - P_{FP}^*$. Un stéganographe est alors dit en situation de risque si Eve, la gardienne, dispose d'un détecteur δ dont la probabilité de faux positif (P_{FP}) et de faux négatif (P_{FN}) vérifient respectivement : $P_{FP} < P_{FP}^*$ et $P_{FN} < P_{FN}^*$.*

Théorème 1 (Loi de la racine carrée de la capacité stéganographique [Ker, 2007a]). *Soit n la taille du support de couverture et m la longueur du message à insérer, alors pour quelles que soient les probabilités (P_{FP}^*, P_{FN}^*) définissant le risque acceptable pour le stéganographe :*

- *si $m/\sqrt{n} \rightarrow \infty$ lorsque $n \rightarrow \infty$ alors il existe une taille du support de couverture n à partir de laquelle le stéganographe n'est pas en situation de risque.*

9. Pour rappel, un schéma stéganographique est dit parfaitement sûr si $D_{KL}(P_C \| P_S) = 0$, et ε -sûr si $D_{KL}(P_C \| P_S) < \varepsilon$ (voir Eq 2.4, dans la section 2.4.1).

- si $m/\sqrt{n} \rightarrow 0$ lorsque $n \rightarrow \infty$ alors il existe une taille du support de couverture n à partir de laquelle le stéganographe est en situation de risque.

De façon informelle, la loi de la racine carré indique qu'il est possible d'envoyer m bits de message en toute sécurité (sans risque), si m ne croît pas plus rapidement que \sqrt{n} . Par contre, si l'on fait croître le taille du message, m , plus rapidement que la racine carrée de taille du support de couverture, alors le stéganographe est en risque de détection.

Par la suite dans la littérature, ce théorème a été étendu aux supports de couverture dont la distribution peut être modélisée par un champ de Markov dans [Filler *et al.*, 2009], et au cas où la distribution du support de couverture est partiellement connue [Ker, 2010].

Remarque : En pratique, la loi de la racine carrée est utilisée, lorsque l'on construit une base d'images contenant des images de couverture et des image stégo, et dont les images sont de tailles différentes. Dans ce cas, pour avoir une même capacité stéganographique, le *payload* m est adapté, en fonction de la taille n de l'image de couverture.

2.4.4 L'efficacité d'insertion

L'efficacité d'insertion est le nombre de bits de message secret insérés par unité de distorsion. Autrement dit, c'est le nombre de bits de message insérés pour une modification du médium de couverture :

$$e = \frac{E_{\mathbf{x}}[\log_2 |\mathcal{M}(\mathbf{x})|]}{E_{\mathbf{x}, \mathbf{m}}[D(\mathbf{x}, \mathbf{y})]}, \quad (2.7)$$

avec E l'espérance mathématique, et $D(\mathbf{x}, \mathbf{y})$ la distorsion entre le support de couverture \mathbf{x} et le stégo-support \mathbf{y} , causée par l'insertion du message \mathbf{m} .

Par définition, l'efficacité d'insertion est une mesure qui dépend du processus d'insertion choisi, du support hôte \mathbf{x} , et de la mesure de distorsion $D(\mathbf{x}, \mathbf{y})$ utilisée.

Intuitivement, plus ce nombre est grand, plus nous avons la possibilité d'effectuer peu de modification tout en insérant un grand nombre de bits de message. Ce critère a été utilisé durant les années 2000, car on estimait que la détectabilité d'un schéma est liée au nombre de modifications (moins il y a de modifications, moins le schéma est détectable). Cependant, comme nous allons le voir en section 2.6, le nombre de modifications, apportées sur le médium hôte, n'est plus actuellement (en 2013) le critère le plus adapté pour juger de la sécurité d'un schéma stéganographique. Nous verrons que la manière de modifier le support est plus importante pour préserver la sécurité des données cachées, quitte à modifier plus de pixels.

Si l'efficacité d'insertion n'est pas le critère majeur pour évaluer la sécurité d'un schéma d'insertion par modification, elle permet néanmoins de donner un ordre de grandeur de

la détectabilité et de l'efficacité du code correcteur utilisé (voir le chapitre 3). Dans cet esprit, nombreux sont les travaux sur les codes qui ont utilisé cette mesure. En 1988, R. Crandall souligne l'importance des codes correcteurs d'erreurs en stéganographie (chapitre 3.3). Il suggère, pour augmenter l'efficacité d'insertion, d'utiliser la technique de *Matrix embedding* [Crandall, 1998]. Quelques années plus tard, en 2001, A. Westfeld concrétise pour la première fois cette approche à travers son algorithme F5 [Westfeld, 2001]. Un peu plus tard, [Fridrich *et al.*, 2005] proposent une nouvelle amélioration de l'algorithme F5. Pour leur algorithme nsF5, ils utilisèrent les codes à *codes à papier mouillé*. Par la suite plusieurs études ont été menées pour revisiter l'apport des codes correcteurs à l'efficacité d'insertion des schémas stéganographiques par modification. Parmi ces travaux on trouve [Dijk et Willems, 2001] [Galand et Kabatiansky, 2003] [Bierbrauer, 2004] [Bierbrauer et Fridrich, 2008].

2.5 Méthodes de stéganographie usuelles

2.5.1 Insertion dans le domaine spatial

L'insertion dans le domaine spatial concerne les images fixes non compressées, qui peuvent être représentées par différents formats BMP, RAW, TIFF, PGM ... etc. Une image non compressée, $\mathbf{x} = (x_1, \dots, x_n) \triangleq \mathcal{I}^n$, est une succession de n échantillons appelés *pixels*. Elle peut être en noir et blanc avec $\mathcal{I} = \{0, 1\}$, en niveaux de gris avec $\mathcal{I} = \{0, \dots, 255\}$, ou en couleur avec $\mathcal{I} = \{0, \dots, 255\}^3$. Pour chaque canal de couleur, la valeur de chacun des pixels, $x_i \in \mathbb{F}_{2^b} = \{0, \dots, 2^b - 1\}$ avec $i = \{1, \dots, n\}$, est représentée numériquement par un entier non-signé (positif ou nul) codé sur b bits, et donnée par :

$$x_i = \sum_{l=0}^{b-1} b_{i,l} 2^l, \quad (2.8)$$

tel que $b_{i,l} \in \{0, 1\}$ représente le $l^{\text{ème}}$ bit codant le pixel x_i . Cette formulation mathématique met en évidence que le degré informatif des bits, pour le codage d'un pixel x_i , est différent d'un plan à un autre. En effet, on peut constater que le premier bit $b_{i,0}$ est pondéré par 2^0 , alors que dernier bit $b_{i,b-1}$ est pondéré par 2^{b-1} . Cette propriété est à l'origine des premiers algorithmes stéganographiques de l'état de l'art, qui sont décrits ci-dessous.

La Figure 2.5 illustre les différents plans de bit d'une image en niveau de gris, en partant du bit de poids fort (MSB pour Most Significant Bit) jusqu'au bit de poids faible (LSB pour Least Significant Bit).

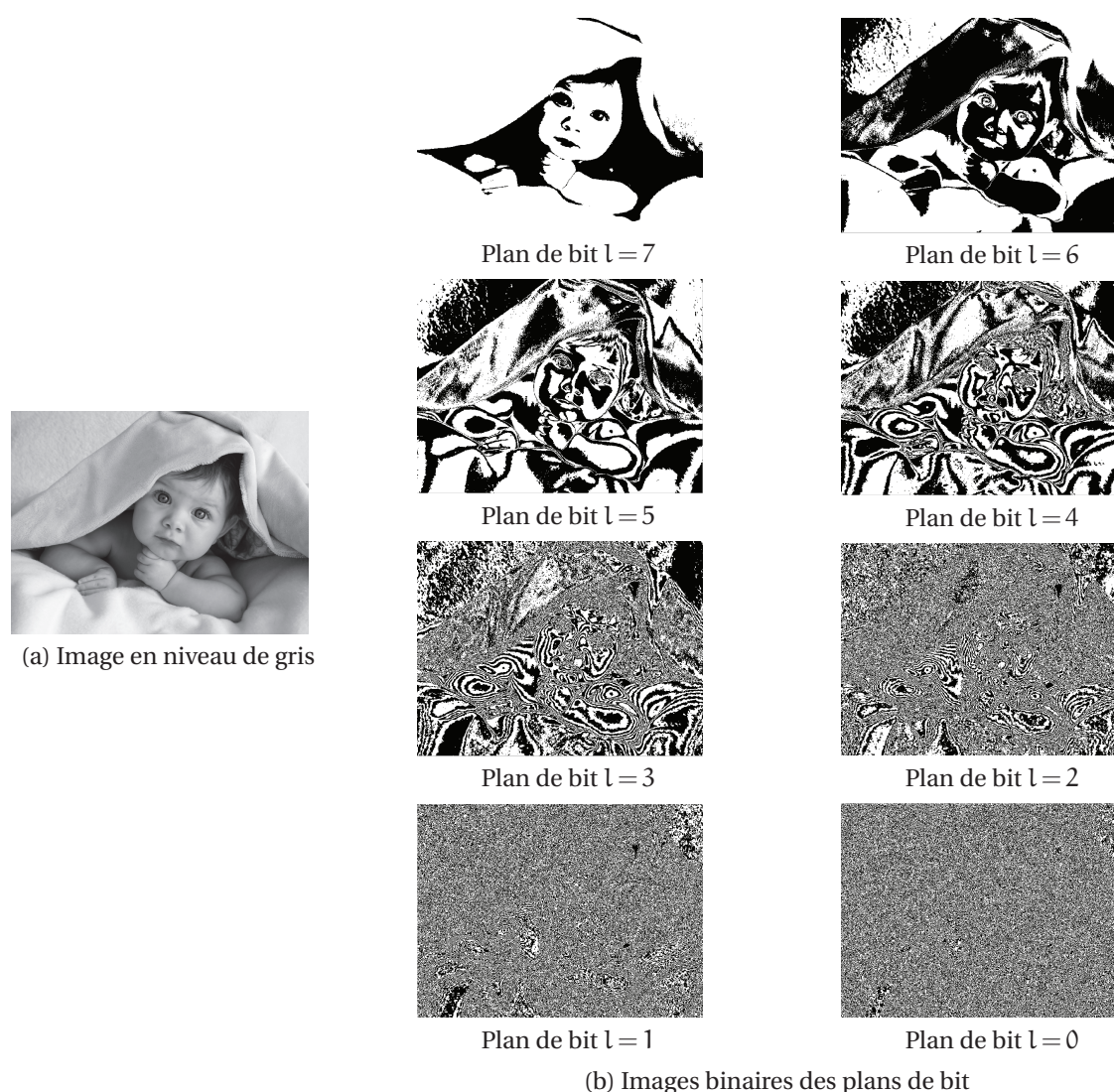


FIGURE 2.5 – Décomposition en plans de bits d’une image en niveaux de gris.

Stéganographie par substitution de LSB (*LSB Replacement*)

Historiquement, la technique de substitution des bits de poids faible (*LSB Substitution*) est la première méthode de stéganographie dans la littérature. Elle reste encore aujourd’hui la méthode la plus utilisée sur Internet, sans doute pour sa simplicité d’implémentation. Cette technique consiste à substituer les bits de poids faibles (les LSB) des pixels par les bits de message à insérer. Autrement dit, pour insérer un message $\mathbf{m} = (m_1, \dots, m_m)$, le dernier bit de poids faible, $b_{i,0}$ (équation 2.8), de chaque pixel est remplacé par un bit du message à dissimuler. Le sens de parcours des pixels est usuellement choisi par un parcours pseudo-

aléatoire. Pour ce faire, l'émetteur et le récepteur doivent préalablement échanger une clé k , utilisée comme graine d'un générateur de nombre pseudo-aléatoire.

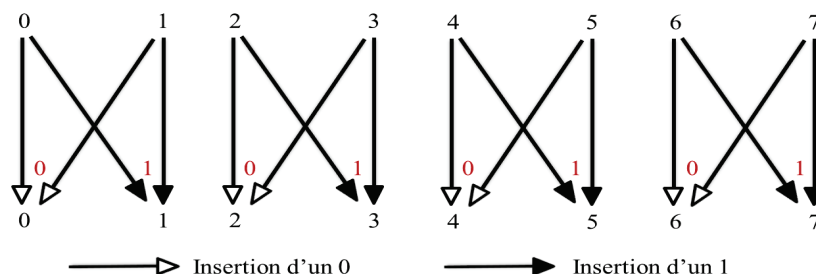


FIGURE 2.6 – Exemple de modification des LSB des pixels par la technique de substitution. Pour rappel, les pixels pairs ont un LSB égal à 0, alors que les pixels impairs ont LSB égal à 1.

Comme illustré sur l'exemple de la Figure 2.6, la substitution du LSB, pour un pixel donné, peut entraîner au plus une modification d'amplitude 1. Cette modification est imperceptible à l'œil nu. Par ailleurs, si nous observons les différents plans de bits d'une image en niveaux de gris, comme l'illustre la Figure 2.5, nous remarquons que les plans des bits de poids faibles contiennent moins d'information pertinente. Nous constatons également que les plans LSB des pixels sont nettement moins structurés que ceux du poids fort (les MSB).

La stéganographie par substitution des LSB est cependant une technique très facilement attaquable. En effet, même si les modifications apportées n'affectent pas l'apparence extérieure du support, elles altèrent considérablement la distribution statistique de celui-ci. Le changement dans la distribution des valeurs de pixels est si fort, qu'une simple analyse statistique par paires, tel que le χ^2 , permettrait facilement d'attaquer le système [Westfeld et Pfitzmann, 1999].

Stéganographie par correspondance de LSB (*LSB Matching*)

La stéganographie par correspondance des LSB, encore appelée *LSB Matching* ou ± 1 *embedding*, est l'amélioration la plus courante de la stéganographie par substitution des LSB. Cet algorithme d'insertion, qui est très proche de la technique par substitution des LSB, insère également le message $\mathbf{m} \in \{0, 1\}^m$ dans les LSB des pixels, mais en incrémentant ou décrémentant aléatoirement la valeur du pixel. Ici encore, le sens de parcours des pixels est habituellement choisi aléatoirement.

La méthode de stéganographie par correspondance des LSB a été proposée pour la première fois par [Sharp, 2001]. Le but de cette technique d'insertion est d'apporter une solution au problème des artefacts statistiques de la stéganographie par *LSB substitution*. En

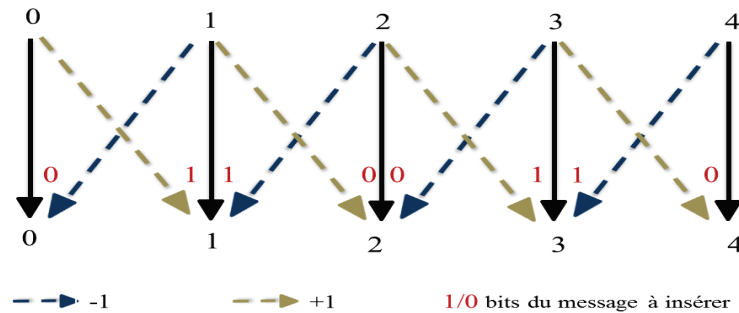


FIGURE 2.7 – Exemple de modification des LSB des pixels par la technique de correspondance.

effet, contrairement à la stéganographie par substitution des LSB, la méthode de stéganographie par correspondance des LSB n'altère pas la distribution statistique du premier ordre du support hôte. Ainsi, toutes les attaques ciblées, spécifiquement dédiées à la détection de la stéganographie par substitution des LSB et n'utilisant qu'une statistique de 1^{er}-ordre, sont inefficaces pour détecter la méthode d'insertion par correspondance des LSB.

La Figure 2.7 illustre un exemple de modification des bits de poids faible des pixels, par la technique de correspondance.

2.5.2 Insertion dans le domaine transformé

En stéganographie, la dissimulation d'informations secrètes dans un domaine transformé de l'image, est très couramment utilisée, car les images échangées sur Internet sont le plus souvent compressées avec pertes au format JPEG ou JFIF. Pour cela, des schémas sténographiques adaptés à ce genre de format ont été élaborés. Ces formats d'images, qui reposent sur une transformé discrète, possèdent des propriétés statistiques particulières. Le fait de ne pas en tenir compte peut rendre le système de dissimulation détectable.

À regarder de plus près, la plupart des méthodes de stéganographie qui opèrent dans un domaine transformé, sont des variantes des méthodes sténographiques spatiales, qui sont décrites auparavant [Fridrich, 2009]. À titre d'exemple, les algorithmes de stéganographie, classiquement utilisés pour les images JPEG, tels que F_5 [Westfeld, 2001], Jsteg [Upham, 1997] ou Outguess [Provos, 2001], reposent principalement sur la méthode d'insertion par modification des LSB. Pour ces algorithmes, la méthode de modification utilisée est appliquée aux coefficients DCT quantifiés et non plus directement aux valeurs des pixels. La Figure 2.8 illustre un exemple de modification des coefficients DCT, pour les algorithmes F_5 et Jsteg.

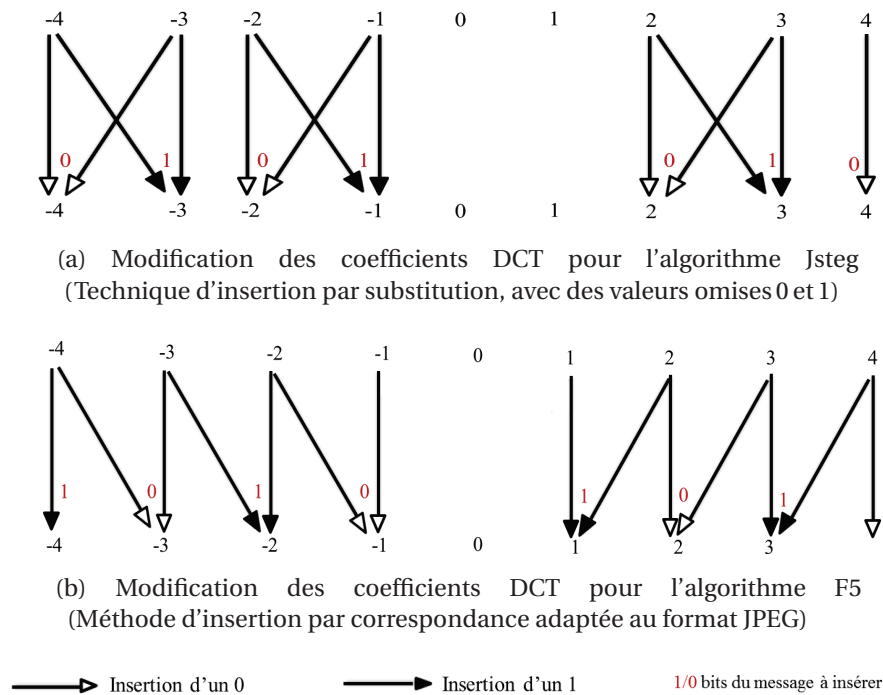


FIGURE 2.8 – Exemple de modification des coefficients DCT pour les algorithmes Jsteg et F5.

2.6 Méthodes de stéganographie adaptatives

Parmi les différentes méthodes de dissimulation de données existantes dans la littérature actuelle, on distingue deux approches d'insertion principales à savoir : les méthodes basées sur la préservation de modèles stéganographiques FCM-[Kodovský *et al.*, 2008], et les méthodes dites adaptatives qui reposent sur la minimisation d'impact d'insertion MOD-[Filler et Fridrich, 2011] HUGO-[Pevný *et al.*, 2010] WOW-[Holub et Fridrich, 2012] UNIWARD-[Holub et Fridrich, 2013]. La stéganographie par préservation de modèles statistiques a pour but de préserver un type particulier d'images de couverture, ceci en conservant au mieux la distribution originale des vecteurs caractéristiques. La stéganographie adaptative, quand à elle, consiste à dissimuler le message secret dans les zones de l'image de couverture qui sont difficilement détectables. Autrement dit, pour insérer un message, il est d'abord nécessaire de sélectionner les zones de l'image sûres, celles qui après insertion introduisent une modification de la statistique difficilement détectable. Pour ce faire, la plupart des méthodes adaptatives actuelles se basent sur la minimisation d'une mesure de distorsion modélisant l'impact sur la sécurité dû à l'insertion. L'avantage de ces méthodes adaptatives par minimisation d'impact d'insertion, par rapport aux mé-

thodes par préservation de modèles, est qu'elles sont plus généralistes et même plus sûres, et ce malgré leur nature heuristique.

Les travaux présentés dans ce manuscrit se focalisent sur les méthodes de stéganographie par modification du médium de couverture. Plus particulièrement sur les méthodes d'insertion adaptatives par minimisation d'impact d'insertion. Dans ce qui suit, nous présentons les différents concepts de ce principe ainsi que les différentes méthodes de l'état de l'art.

2.6.1 l'impact d'insertion (la distorsion)

L'impact d'insertion, ou plutôt la distorsion, est une mesure spécifiquement consacrée aux schémas stéganographiques adaptatifs, utilisant le principe de minimisation d'impact d'insertion. Elle est modélisée par une fonction mathématique, $D(\mathbf{x}, \mathbf{y}) : \mathcal{C} \times \mathcal{S} \rightarrow \mathbb{R}$, qui reflète l'impact causé par le processus d'insertion du message secret, sur la statistique originale du médium de couverture. En d'autres termes, il s'agit d'une approximation de la détectabilité statistique causée par l'insertion du message secret.

Soit $\mathbf{x} = (x_1, \dots, x_n)$ une image de couverture composée de n éléments, et $\mathbf{y} = (y_1, \dots, y_n)$ l'image stégo qui lui est associée. D'une manière générale, la distorsion introduite par le processus d'insertion du message secret peut alors être définie formellement par [Ker *et al.*, 2013] :

$$D(\mathbf{x}, \mathbf{y}) = \|f(\mathbf{x}) - f(\mathbf{y})\|, \quad (2.9)$$

avec f est la fonction qui retourne un vecteur caractéristique de l'image passée en paramètre.

Cette formulation mathématique de la fonction de distorsion (Eq. 2.9) pose un gros problème pour le stéganographe, car elle est non-additive et non-locale, et ceci pour tous les espaces caractéristiques utilisés en stéganalyse (histogrammes de premier ordre, co-occurrences d'ordre supérieur ...). Afin de simplifier le problème, deux approches ont été proposées :

La première approche consiste à approximer la distorsion en une version additive. Dans cet esprit, [Filler *et al.*, 2010, Filler *et al.*, 2011] proposent une formulation additive de la distorsion (Eq. 2.10) qui utilise une carte de détectabilité $\rho \in \mathbb{R}_+^n$. Le principe consiste à attribuer, pour chaque élément de couverture x_i , un coût de détectabilité $\rho_i \in \mathbb{R}_+$, qui modélise l'impact de la modification du $i^{\text{ème}}$ élément sur la sécurité. L'additivité de la fonction de distorsion suppose que les modifications effectuées en chaque pixel sont indépendantes les unes des autres. Autrement dit, la modification d'un élément de couverture n'affecte pas la détectabilité des sites voisins.

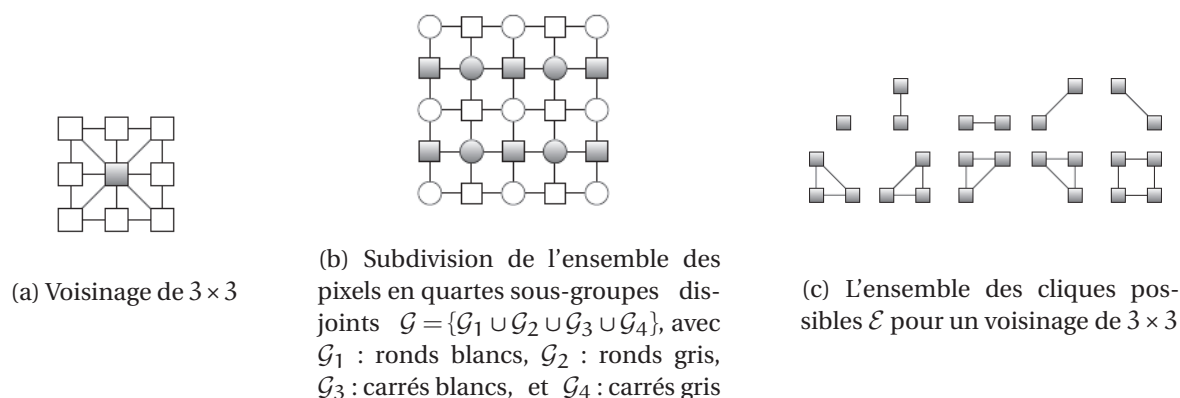


FIGURE 2.9 – Exemple illustrant 1) un partitionnement de l'image, 2) le système de cliques.

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \rho_i |x_i - y_i|, \quad (2.10)$$

tel que $0 \leq \rho_i \leq \infty$ est le coût de modification du $i^{\text{ème}}$ élément de couverture.

À ce stade, nous pouvons aisément conclure que la difficulté majeur de cette approche est le calcul de la carte de détectabilité $\rho \in \mathbb{R}_+^n$, qui doit être corrélée à la détectabilité statistique du support. Le calcul des poids ρ_i reflétant l'impact du processus d'insertion sur la sécurité est encore un problème ouvert, qui a commencé à être étudié à partir de la fin 2010, lors de la compétition BOSS [Bas *et al.*, 2011] à travers l'algorithme HUGO [Pevný *et al.*, 2010]. La section 2.6.3 présente les différentes propositions actuelles de l'état de l'art, pour le calcul de la carte de détectabilité dans le contexte d'une distorsion additive.

La deuxième approche consiste à diviser le problème global (Eq. 2.9), en sous problèmes locaux simples que l'on sait résoudre. Dans cet esprit, [Filler et Fridrich, 2010] proposent une nouvelle fonction de distorsion non-additive (Eq. 2.11) pour les images digitales dans le domaine spatial. Le principe de leur approche consiste à diviser l'ensemble des pixels de l'image de couverture, noté ici \mathcal{G} , en sous-groupes de pixels indépendants les uns des autres $\mathcal{G} = \{\mathcal{G}_1 \cup \dots \cup \mathcal{G}_g\}$. Le message est coupé, par la suite, en petits morceaux égales au nombre des sous-ensembles. Chaque partie du message est insérée dans un sous-groupe, en utilisant une mesure de distorsion qui prend en considération la dépendance des pixels voisins, selon un certain degré de voisinage. La fonction de distorsion utilisée (Eq. 2.11) est une somme de distorsions locales, calculées à partir d'un système particulier de voisinage, nommé *cliques* et noté \mathcal{E} (voir l'exemple illustré sur la Figure 2.9). Elle est non-additive, dont le sens où elle fait appel à la représentation caractéristique de l'image de couverture et des images stégo obtenues après la modification (voir Eq. 2.12).

Pour un système de voisinage donné (voir la Figure 2.9-(a)), nous noterons par \mathcal{E} l'ensemble des cliques possibles, tel que $e \in \mathcal{E}$ est une clique *ssi* chaque paire de pixels de e sont des voisins. La distorsion, selon [Filler et Fridrich, 2010], peut alors être donnée par :

$$D(\mathbf{x}, \mathbf{y}) = \sum_{e \in \mathcal{E}} V_e(\mathbf{x}, \mathbf{y}), \quad (2.11)$$

tel que pour une clique $e \in \mathcal{E}$:

$$V_e(\mathbf{x}, \mathbf{y}) = \|f_e(\mathbf{x}) - f_e(\mathbf{y})\| = \sum_{k=1}^d w_k |f_e^{(k)}(\mathbf{x}) - f_e^{(k)}(\mathbf{y})|, \quad (2.12)$$

avec $f_e(\mathbf{x})$ (resp. $f_e(\mathbf{y})$) la fonction qui retourne le vecteur caractéristique de l'image de couverture \mathbf{x} (resp. de l'image stégo \mathbf{y}) associé à la clique e , d la taille des vecteurs caractéristiques résultant, w_k le poids correspondant à la $k^{\text{ème}}$ dimension du vecteur caractéristique, et $f_e^{(k)}(\mathbf{x})$ la $k^{\text{ème}}$ composante du vecteur caractéristique de l'image \mathbf{x} .

Les travaux de recherches présentés dans ce manuscrit se focalisent sur les méthodes de dissimulation utilisant une approximation additive de la distorsion.

2.6.2 Le problème de minimisation d'impact d'insertion

Tout algorithme pratique de stéganographie par minimisation d'impact d'insertion a pour objectif d'insérer un message $\mathbf{m} = \{0, 1\}^m$ donné dans un support hôte \mathbf{x} , en essayant de réduire au minimum l'impact causé par l'insertion [Filler *et al.*, 2010]. Afin d'atteindre cet objectif, il est important d'établir d'abord une mesure de distorsion $D : \mathcal{C} \times \mathcal{S} \rightarrow \mathbb{R}$ capable de modéliser au mieux la détectabilité statistique due à l'insertion (voir la section 2.6.1). Une fois établie, la fonction de distorsion peut alors être réduite sous la contrainte d'un *payload* fixe.

Dans notre cas, pour une fonction de distorsion additive D , telle que celle donnée par [Filler *et al.*, 2010, Filler *et al.*, 2011] (voir Eq. 2.10), et une insertion binaire ($|x_i - y_i| \leq 1$). La solution au problème de minimisation d'impact d'insertion sous la contrainte d'un *payload* fixe est alors sous la forme suivante [Fridrich et Filler, 2007] :

$$\min D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n p_i \rho_i, \quad (2.13)$$

avec p_i la probabilité de modification du $i^{\text{ème}}$ pixel définie par [Fridrich et Filler, 2007] :

$$p_i = \frac{e^{-\lambda\rho_i}}{1 + e^{-\lambda\rho_i}}. \quad (2.14)$$

Le paramètre λ est obtenu en résolvant l'équation suivante :

$$-\sum_{i=1}^n (p_i \log_2 p_i + (1 - p_i) \log_2 (1 - p_i)) = m. \quad (2.15)$$

Cette formalisation de la stéganographie adaptative par minimisation d'impact d'insertion permet de découper le processus d'insertion en deux étapes distinctes et successives : 1) le calcul d'une carte de détectabilité (calcul des coûts $\{\rho_i\}_{i=1}^n$), et 2) l'insertion par un algorithme adaptatif pratique (en théorie modification des pixels avec les probabilités $\{p_i\}_{i=1}^n$). L'avantage majeur qui découle de cette séparation est que l'évaluation de la sécurité d'une carte de détectabilité particulière ne nécessite plus d'utiliser un algorithme de l'état de l'art pour l'insertion. En pratique si l'on souhaite insérer un message en minimisant l'impact d'insertion (la carte de détectabilité $\rho \in \mathbb{R}_+^n$ est donc connue) avec la contrainte d'un *payload* fixe ($= m$), il est possible de simuler l'insertion optimale en cherchant le paramètre λ (résolution de l'équation 2.15), puis en modifiant chaque pixel x_i selon la probabilité p_i définie dans l'équation 2.14.

2.6.3 La carte de détectabilité

Comme mentionné précédemment, le but de la stéganographie par minimisation d'impact d'insertion est d'apporter le minimum de distorsion au support hôte $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{C}$, afin de produire le *stégo* objet $\mathbf{y} = (y_1, \dots, y_n) \in \mathcal{S}$ qui communique le message secret $\mathbf{m} = (m_1, \dots, m_m)$. Pour ce faire, une fonction de distorsion $D(\mathbf{x}, \mathbf{y})$, souvent additive, est établie afin d'être minimisée sous la contrainte d'un *payload* fixe (voir les sections 2.6.1 et 2.6.2). Cette fonction de distorsion se base généralement sur l'utilisation d'une carte de détectabilité $\rho \in \mathbb{R}_+^n$ qui attribue à chaque élément de couverture x_i avec $i \in \{1, \dots, n\}$, un coût de détectabilité $\rho_i \in \mathbb{R}_+$ modélisant l'impact sur la sécurité dû à la modification de cet élément. La carte de détectabilité calculée est utilisée, par la suite, pour insérer le message secret dans les régions sûres du support hôte. De ce fait, il est important de s'assurer que la carte calculée $\rho = \{\rho_i \in \mathbb{R}_+\}_{i=1}^n$ reflète au mieux le niveau de sécurité de chaque élément de couverture. En pratique, lorsqu'il s'agit d'images numériques naturelles, le calcul de la carte de détectabilité constitue un véritable challenge pour le stéganographe, compte-tenu de la complexité statistique des images et l'absence de modèles statistiques complets. Dans ce contexte, plusieurs travaux ont été menés pour répondre à cette problématique.

Dans cette sous-section nous discutons les différentes propositions de l'état de l'art, pour le calcul de la carte de détectabilité. Nous visitons l'histoire avec la vision "carte de

détectabilité" et "minimisation d'impact d'insertion", inexistantes à l'époque de la création de certains algorithmes.

L'algorithme F5

L'algorithme F5 pour les images JPEG, proposé par [Westfeld, 2001], est un algorithme par correspondance (LSB *Matching* (± 1)) qui insère les bits de message secret dans les coefficients AC DCT non-nuls (voir l'exemple illustré sur la Figure 2.8). Pour l'insertion, il utilise la technique de codage matriciel (*Matrix Embedding*), afin de réduire le nombre total de modifications à effectuer sur le support hôte. À travers cette démarche, l'auteur suppose que tous les éléments de couverture ont le même risque de détectabilité lors de la modification du médium de couverture. Autrement dit, il considère un coût de détectabilité identique pour tous les éléments de couverture ($\rho_i = 1$).

L'algorithme nsF5

Pour corriger les faiblesses de l'algorithme F5, [Fridrich *et al.*, 2005] proposent une nouvelle version améliorée de l'algorithme initial. Le principe de leur algorithme, nommé nsF5 (*non-shrinkage F5*), consiste à utiliser la technique de papier mouillé, avec des coûts de détectabilité à deux valeurs $\rho_i \in \{1, \infty\}$. Pour figer les sites sensibles à l'insertion, les zones ne devant pas être modifiées auront donc un coût de détectabilité $\rho_i = \infty$, et les autres zones un $\rho_i = 1$.

L'algorithme HUGO

L'algorithme HUGO¹⁰, proposé par [Pevný *et al.*, 2010] lors de la compétition BOSS¹¹, a joué un rôle prépondérant dans les algorithmes de dissimulation adaptatifs actuels. Pour l'insertion du message secret, il utilise une carte de détectabilité variable, qui à chaque pixel de l'image de couverture attribue un niveau de détectabilité $\rho_i \in \mathbb{R}_+$, comme suggéré dans [Filler *et al.*, 2010, Filler *et al.*, 2011]. Le calcul de coût de détectabilité repose sur l'utilisation de caractéristiques de haute dimension calculées à partir de l'image de couverture. Ces caractéristiques correspondent aux probabilités conditionnelles en chaque pixel de l'image filtrée.

10. HUGO : Highly Undetectable steGO.

11. Break Our Steganography System [Bas *et al.*, 2011] : premier challenge en stéganalyse, visant à évaluer la sécurité de l'algorithme de HUGO. Une base de 10000 images était mise à disposition de la communauté pour être analysée. L'objectif était de distinguer les images *stégo* des images *cover*. <http://www.agents.cz/boss/BOSSFinal/>

L'algorithme MOD

L'algorithme MOD ¹² proposé par [Filler et Fridrich, 2011], étend la proposition HUGO en définissant un coût de détectabilité $\rho_i \in [0, \infty]$ paramétré par un nombre élevé de paramètres. La recherche des paramètres menant au meilleur niveau de sécurité est effectuée itérativement en répétant insertion puis modification des paramètres, grâce à l'algorithme d'optimisation *downhill simplex*. Le niveau de sécurité est évalué à chaque itération en utilisant comme critère la taille de la marge d'un SVM ¹³.

WOW & UNIWARD

L'algorithme WOW proposé par [Holub et Fridrich, 2012], ainsi que l'algorithme UNIWARD proposé par [Holub et Fridrich, 2013], passent par le domaine d'ondelettes, et utilisent des filtres directionnels pour le calcul de la carte de détectabilité. L'algorithme UNIWARD attribut à chaque élément de couverture un coût de détectabilité, $\rho_i \in [0, \infty[$, calculé à partir de la variation relative entre les coefficients d'ondelettes résiduels de l'image de couverture, et les coefficients d'ondelette résiduels de l'image stégo obtenue après modification de l'élément en question. L'algorithme WOW, qui est très similaire à UNIWARD, calcule d'abord la différence pondérée entre les coefficients d'ondelettes résiduels de l'image de couverture, et les coefficients d'ondelettes résiduels de l'image stégo puis agrège le résultat obtenu pour construire une carte de détectabilité $\rho = \{\rho_i \in [0, \infty]\}_{i=1}^n$. Le but étant d'éviter l'insertion dans les endroits prévisibles de l'images de couverture (tel que les bordures), et de favoriser l'insertion dans les endroits imprévisibles (tel que les zones texturées ou bruitées).

12. MOD : Model Optimized Distortion [Kodovský *et al.*, 2011].

13. SVM : Support Vecteur Machine.

2.7 Synthèse

Au cours du présent chapitre, nous avons présenté les différentes notions de stéganographie, ainsi que les principales méthodes de dissimulation d'information cachées. Parmi les méthodes de dissimulation présentées, nous nous sommes intéressés, plus particulièrement, aux méthodes de stéganographie adaptatives, celles qui reposent sur le principe de minimisation d'impact d'insertion. Pour l'insertion du message secret, nous avons vu que la plupart des méthodes adaptatives actuelles utilisent une carte de détectabilité, ρ , qui attribue à chaque élément de couverture un coût de détectabilité, ρ_i , reflétant son niveau de sécurité lors de la modification. Cette carte de détectabilité étant très importante pour l'insertion, doit refléter au mieux la détectabilité statistique, ce qui représente un véritable verrou scientifique en stéganographie en raison du manque de modèles statistiques complets représentant les images naturelles de nature complexe et variée. Pour répondre à cette problématique, nous présentons dans le chapitre 5 un nouveau schéma d'insertion de données secrètes, basé sur l'utilisation d'oracle pour le calcul de la carte de détectabilité.

Les codes correcteurs d'erreurs en stéganographie

N'admettez rien a priori si vous pouvez le vérifier.

RUDYARD KIPLING - *Écrivain*

Préambule

L'introduction des codes correcteurs d'erreurs en stéganographie fut une grande avancée en terme d'efficacité d'insertion des algorithmes stéganographiques. Dans ce chapitre, nous nous intéressons de près à l'utilisation de ce paradigme mathématique dans le contexte de stéganographie. Nous discutons, dans un premier temps, l'utilité des codes correcteurs d'erreurs de manière générale (section 3.1). Ensuite, nous définissons le principe général des codes linéaires (section 3.2). Enfin, nous nous intéressons plus particulièrement au rôle ainsi qu'à l'apport des codes correcteurs d'erreurs en stéganographie (section 3.3).

Sommaire

3.1	Utilité des codes correcteurs d'erreurs	38
3.2	Les codes correcteurs linéaires	39
3.3	Stéganographie et codes correcteurs	41
3.4	Synthèse	48

3.1 Utilité des codes correcteurs d'erreurs

Les codes correcteurs ont été proposés, pour la première fois, pour corriger les erreurs qui peuvent se produire au cours de la transmission de données sur un canal de communication particulier, ou les erreurs survenant au cours de la lecture/écriture sur un support physique (bande, CD, DVD,...etc), ou même sur des supports de stockage, lorsque les données subissent une altération donnée.

Prenons l'exemple d'une communication numérisée par Internet. Lors de la transmission d'un signal numérique donnée (une image, un son, un fichier text...), celui-ci est généralement ramené à une séquence de bits $e_1 e_2 \dots$. À cause du bruit présent et des différentes manipulations qui peuvent avoir lieu sur le canal de communication, la séquence envoyée peut subir des dégradations au cours de la transmission. De ce fait, il est important de trouver le moyen de fiabiliser la transmission de ces données. Pour ce faire, des méthodes de détection et correction d'erreurs ont été introduites afin de 1) vérifier l'intégrité des données interceptées, et 2) corriger, dans la mesure du possible, les erreurs produites lors de la transmission.

Une des méthodes de codage les plus simples, illustrant ce principe, est de répéter chaque bit de la séquence envoyée. La séquence $e_1 e_2 \dots$, que l'on veut envoyer, sera ainsi transmise sous la nouvelle forme $e_1 e_1 e_2 e_2 \dots$. Lors de la réception du message, pour détecter les éventuelles erreurs, le décodeur utilisé peut simplement comparer les couples de bits reçus. Si les bits composant un couple sont différents alors il y a présence d'une erreur de transmission. Cette démarche simple permet, en doublant la longueur du message, de détecter la présence éventuelles d'erreurs. Toutefois, elle ne permet pas de corriger l'erreur trouvée.

Un autre exemple simple, permettant cette fois ci de corriger les erreurs détectées, est de tripler les bits envoyés au lieu de les doubler. Dans la mesure où l'on considère qu'il y a au maximum une erreur pour chaque 3 bits de la séquence envoyée, alors il est possible dans ce cas de figure de corriger les erreurs détectées. Lors de la réception, le décodeur n'a qu'à choisir le symbole qui apparaît deux fois dans chaque triplet de la séquence reçue. Cette méthode n'est efficace que dans le cas où il y a au plus une erreur pour chaque 3 bits envoyés. De nombreux codes plus efficaces ont été proposés dans la littérature [Peterson et Weldon, 1972] [Abbrugiati, 2006] [Roth, 2006].

3.2 Les codes correcteurs linéaires

3.2.1 Définition d'un code linéaire

Un code correcteur d'erreur est dit linéaire si il est structuré comme un sous-espace vectoriel d'un espace vectoriel de dimension finie sur un corps fini. Dans le cas binaire, le corps employé est celui du corps de Galois $\mathbb{F}_2 = \{0, 1\}$, et le terme correcte est alors celui de code linéaire binaire.

Un code linéaire est décrit par trois paramètres : $[n, k, \delta]$, avec $n > k$, et tel que n représente la longueur du code, k définit la taille des mots une fois décodés (appelé également la dimension du code), et δ représente la distance minimale entre chaque mot du code. Dire que φ est un code linéaire, est exactement équivalent à dire que φ est une application injective : $\varphi : \{0, 1\}^k \rightarrow \{0, 1\}^n$. Les codes linéaires constituent l'essentiel des codes correcteurs employés par l'industrie et les scientifiques pour la détection et la correction des erreurs sur les différents canaux de transmission (physique/numérique).

3.2.2 Distance minimale d'un code linéaire

Une des propriétés qui rendent les codes linéaires très intéressants est qu'il est peu complexe de calculer la distance minimale. Pour un code linéaire, la distance minimale δ est égale au plus petit poids non nul de l'ensemble des mots de code. En d'autres termes, c'est la plus petite distance entre chaque couple de mots de code. La distance utilisée pour un couple de mots de code $(c_1, c_2) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ est celle de *Hamming*, définie par $d_H(c_1, c_2) = \sum_{i=1}^n c_{1_i} \oplus c_{2_i}$. Le *poids de Hamming* d'un code c , quand à lui, définit le nombre de 1 dans le vecteur, et est donné par la distance $d_H(c, \vec{0})$.

3.2.3 Matrice génératrice d'un code linéaire

Dans la théorie des codes détecteurs et correcteurs d'erreurs, la transmission d'un message $m \in \mathbb{F}_2^k$, au travers d'un canal bruité, s'effectue en transformant celui-ci en un mot de code $c \in \mathbb{F}_2^n$, par le biais d'une matrice dite *génératrice*, notée $G \in \mathcal{M}_{n,k}(\mathbb{F}_2)$ (c'est-à-dire avec n lignes, k colonnes, à coefficients dans $\{0, 1\}$). De manière plus formelle, l'opération d'encodage du message m s'effectue par l'équation suivante :

$$c = Gm. \quad (3.1)$$

La matrice génératrice \mathbf{G} est souvent représentée sous une forme particulière, dite *systematique*, tel que :

$$\mathbf{G} = \begin{pmatrix} \mathbf{I}_k \\ \mathbf{G}' \end{pmatrix}, \quad (3.2)$$

où \mathbf{I}_k est la matrice d'identité d'ordre k , et \mathbf{G}' est une matrice de redondance. La matrice \mathbf{G}' permet la détection des erreurs éventuelles.

3.2.4 Matrice de contrôle d'un code linéaire

À la réception, le mot reçu est contrôlé dans le but de vérifier s'il comporte ou non des erreurs éventuelles. Pour ce faire, le récepteur, qui reçoit un mot $\mathbf{c}' \in \mathbb{F}_2^n$, utilise une matrice dite de *contrôle* (ou de *parité*) $\mathbf{H} \in \mathcal{M}_{n-k,n}(\mathbb{F}_2)$, pour calculer le *syndrome* $\mathbf{s} \in \mathbb{F}_2^{n-k}$:

$$\mathbf{s} = \mathbf{H}\mathbf{c}'. \quad (3.3)$$

Dans le cas d'un code systematique :

$$\text{Si } \mathbf{G} = \begin{pmatrix} \mathbf{I}_k \\ \mathbf{G}' \end{pmatrix} \text{ alors } \mathbf{H} = \begin{pmatrix} \mathbf{G}' & \mathbf{I}_{n-k} \end{pmatrix}. \quad (3.4)$$

Une fois calculé, si le syndrome $\mathbf{s} = \{0, \dots, 0\}$, alors le récepteur déduit que le mot $\mathbf{c}' = \mathbf{c}$, et que le message reçu ne comporte pas d'erreurs. Dans le cas inverse, si $\mathbf{s} \neq \{0, \dots, 0\}$, alors le récepteur déduit que le message intercepté comporte une ou plusieurs erreurs de transmission.

Remarquons au passage, qu'il est possible d'avoir plusieurs codes qui partagent le même syndrome \mathbf{s} . Dans le jargon des codes correcteurs, l'ensemble des codes ayant le même syndrome \mathbf{s} est appelé la classe (ou *coset*) de \mathbf{s} , et est défini par : $\mathcal{C}(\mathbf{s}) = \{\mathbf{c} \in \mathbb{F}_2^n : \mathbf{H}\mathbf{c} = \mathbf{s}\}$. L'élément de la classe $\mathcal{C}(\mathbf{s})$ qui possède le plus faible poids de Hamming est appelé le chef de classe (en anglais *coset leader*).

3.3 Stéganographie et codes correcteurs

3.3.1 Technique de matrix embedding

Comme mentionné précédemment (chapitre 2), l'objectif principal de la stéganographie est de transmettre de manière furtive l'information secrète au sein d'un médium hôte. Lors de la dissimulation, l'information introduite va forcément altérer le support hôte. De ce fait, il est important de trouver un moyen intelligent d'introduire l'information secrète, de telle sorte que la distorsion apportée sur le support de couverture soit la plus petite possible (voir la section 2.6.2). Ces premières approches en stéganographie associaient la distorsion au nombre de modification apportées sur le support. Pour minimiser le nombre de modifications apportées tout en essayant de communiquer le maximum d'information possible (ce qui revient à augmenter l'efficacité d'insertion, voir la section 2.4.4), ces techniques utilisaient le concept *de matrice de codage* (en anglais *matrix embedding*), qui a été introduit par [Crandall, 1998], puis implémenté pour la première fois par [Westfeld, 2001] avec l'algorithme F5 utilisant les codes de Hamming. L'idée derrière cette méthode de codage par syndrome consiste à détourner l'utilisation classique des codes correcteurs d'erreurs, pour pouvoir représenter le message à transmettre sous forme de syndrome. Toute l'astuce de cette approche réside du côté de l'émetteur (le stéganographe), qui doit trouver le moyen de modifier l'image de couverture, de sorte que : 1) le syndrome calculé au niveau du récepteur correspondant au message désiré, et 2) l'image soit la moins modifiée.

De manière plus formelle, le but recherché de la technique de *matrix embedding* est de communiquer un message $\mathbf{m} \in \mathbb{F}_q^{n-k}$ (dans le cas binaire $q = 2$) au travers d'un support $\mathbf{x} \in \mathbb{F}_q^n$, ceci en le modifiant le moins possible. Le principe est de modifier le support de couverture \mathbf{x} en \mathbf{y} , de telle sorte que :

$$\mathbf{H}\mathbf{y} = \mathbf{m}, \quad (3.5)$$

avec $\mathbf{H} \in \mathcal{M}_{n-k,n}$ la matrice de parité du code. La transformation du vecteur $\mathbf{x} \in \mathbb{F}_q^n$ en $\mathbf{y} \in \mathbb{F}_q^n$ s'effectue alors en recherchant le vecteur de modification $\mathbf{e} \in \mathbb{F}_q^n$:

$$\mathbf{y} = \mathbf{x} + \mathbf{e}. \quad (3.6)$$

En injectant, l'équation Eq. 3.5 dans Eq. 3.6, nous obtenons alors la formule suivante :

$$\mathbf{H}(\mathbf{x} + \mathbf{e}) = \mathbf{m} \Leftrightarrow \mathbf{H}\mathbf{e} = \mathbf{m} - \mathbf{H}\mathbf{x}. \quad (3.7)$$

Le problème consiste donc à trouver le vecteur \mathbf{e} , qui est un mot de code ayant comme syndrome $\mathbf{m} - \mathbf{H}\mathbf{x}$. Puisque le but recherché est d'effectuer le minimum possible de modifications sur le support \mathbf{x} , le code optimal recherché devrait être le chef de la classe $\mathcal{C}(\mathbf{m} - \mathbf{H}\mathbf{x})$. Autrement dit, le vecteur qui possède le plus faible poids de Hamming.

De manière générale, la résolution de l'Eq. 3.7 est très compliquée (en temps et en espace de calcul), car elle nécessite l'utilisation du pivot de Gauss, de complexité cubique sur le nombre de lignes de la matrice de parité \mathbf{H} (dans notre cas le nombre de ligne est de $n - k$). Pour simplifier le problème, il est fréquent d'utiliser des codes linéaires de forme simple. Dans la plupart du temps, les codes utilisés sont des codes linéaires ayant un rayon de couverture faible.

Exemple de matrix embedding : utilisation des codes de Hamming

Les codes de Hamming sont les premiers codes correcteurs à avoir été utilisés pour le concept de *matrix embedding*, matérialisé pour la première fois par l'algorithme F5. Grâce à leur propriétés bien particulières, cette famille de codes linéaires permet, à la fois, la détection et la correction d'erreur, pour une erreur au maximum. Parmi les propriétés qui rendent ce code très intéressant est que la distance minimale δ est égale à 3, et un rayon de couverture faible égale à 1. Par ailleurs, les codes de Hamming sont des codes paramétrés par un entier $p \in \mathbb{N}^+$, tel que, $n = 2^p - 1$ et $k = 2^p - 1 - p$. Pour une utilisation des codes de Hamming en stéganographie, le syndrome $\mathbf{m} - \mathbf{H}\mathbf{x} = \mathbf{H}\mathbf{e}$, représentant le message secret, est donc de longueur $n - k = p$. La matrice de parité du code, quand à elle, est une matrice $\mathbf{H} \in \mathcal{M}_{p, 2^p - 1}$ formée de colonnes en binaire représentant les premiers entiers (1 à $2^p - 1$).

Afin d'illustrer le principe de *matrix embedding* en stéganographie, nous présentons ici un exemple simple, basé sur les codes de Hamming.

Pour $p = 3$, et $\mathbf{m} = (1, 0, 1)$ le message que nous souhaitons insérer dans le support hôte $\mathbf{x} = (0, 1, 1, 1, 0, 0, 1)$, la matrice de parité \mathbf{H} est donc sous la forme suivante :

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad (3.8)$$

Comme expliqué auparavant (section 3.3.1), le principe de la technique de *matrice embedding* consiste à chercher le vecteur de modification $\mathbf{e} = (e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ tel que $\mathbf{H}(\mathbf{x} + \mathbf{e}) = \mathbf{m}$. Pour ce faire, nous calculons simplement le syndrome $\mathbf{m} - \mathbf{H}\mathbf{x}$:

$$\begin{aligned} \mathbf{m} - \mathbf{H}\mathbf{x} &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \end{aligned} \quad (3.9)$$

Le résultat obtenu est alors $\mathbf{H}\mathbf{e} = (1, 1, 1)$, ce qui correspond à la 7^{ème} colonne de la matrice de parité \mathbf{H} . Le vecteur de modification est alors sous la forme : $\mathbf{e} = (0, 0, 0, 0, 0, 0, 1)$.

Pour communiquer le message $\mathbf{m} = (1, 0, 1)$, le support $\mathbf{x} = (0, 1, 1, 1, 0, 0, 1)$ est alors transformé en $\mathbf{y} = \mathbf{x} + \mathbf{e} = (0, 1, 1, 1, 0, 0, 0)$. Ainsi, lors du décodage, le récepteur, qui a reçu le support stéganographié \mathbf{y} , n'aura alors qu'à calculer le syndrome $\mathbf{H}\mathbf{y}$ pour extraire le message secret :

$$\begin{aligned}
 \mathbf{Hy} &= \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
 &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \\
 &= \mathbf{m}
 \end{aligned} \tag{3.10}$$

Nous venons d'illustrer la technique de *matrice embedding*, appliquée aux codes de Hamming, pour insérer p bits, dans un support x de $2^p - 1$ bits, ceci en apportant au maximum une seule modification sur celui-ci (soit une modification de 0 ou de 1 dans le support). Notons au passage, que cette technique n'est efficace que sur une petite portion de l'image. En effet, si on désire appliquer la technique précédente sur une image en niveau de gris de 512×512 (soit 262144 pixels (ou LSB) potentiels pour la dissimulation), et n'effectuer qu'une seule modification dans l'image, le message inséré doit être constitué au plus de 18^1 bits, ce qui est très peu pour la communication. Pour envoyer un message plus grand, tout en effectuant qu'une seule modification, nous avons besoin d'un support de couverture beaucoup plus grand. En pratique, pour palier ce problème et augmenter la capacité d'insertion, le message à dissimuler est découpé en plusieurs morceaux. Chaque morceau du message est ensuite inséré, par la technique précédente, dans une partie de l'image.

3.3.2 Les codes à papier mouillé

La technique de *matrix embedding* peut également être appliquée à d'autres codes linéaires. Dans cette partie, nous présentons le principe des codes à papier mouillé (en anglais *wet paper codes*) [Fridrich *et al.*, 2005].

Posons d'abord le problème, avant d'expliquer le principe de cette technique. En stéganographie, lors de l'insertion, l'émetteur peut décider de choisir "les endroits" du support

1. Pour dissimuler un message de 18 bits, avec la technique de *matrix embedding*, le support hôte utilisé doit être de taille $2^{18} - 1 = 262143 < 262144$, ce qui est équivalent à une image de 512×512 .

où effectuer les modifications (sélection du canal), et ce pour des raisons de sécurité. Le problème est que lors de l'extraction, le récepteur ne connaît pas les pixels utilisés par l'émetteur pour la dissimulation du message secret, et ne peut donc pas déterminer le canal de sélection, puisqu'il n'a pas accès au support original. Ce problème est appelé *écriture sur papier mouillé*, car il est souvent modélisé par la métaphore suivante : un émetteur souhaite cacher un message secret sur un papier, dont certaines parties sont mouillées. Lors de l'écriture du message, l'émetteur ne peut pas écrire au endroits mouillés, et peut modifier uniquement les endroits secs. Lors de la réception, le papier stéganographié ayant séché au cours de la transmission, le destinataire ne peut déterminer les endroits où le message est écrit. On voit donc apparaître le concept de *sélection non partagée*.

La technique de codes à papier mouillé, proposée par [Fridrich *et al.*, 2005] avec l'algorithme nsF5, est une solution efficace pour permettre aux deux parties (l'émetteur et le récepteur) de communiquer des informations secrètes sous le scénario précédent. Pour expliquer le principe de fonctionnement de ces codes, nous présentons dans ce qui suit un exemple illustrant la technique de *matrix embedding*.

Soit $\mathbf{m} = (1, 0, 1)$ le message à dissimuler dans le support hôte $\mathbf{x} = (1, 0, 0, 0, 0, 0, 1)$, et $\mathcal{Q} = \{1, 2, 5, 6, 7\}$ l'ensemble des bits qui peuvent être modifiés, c-à-d que les bits 3 et 4 ne peuvent pas être modifiés. Comme expliqué précédemment (section 3.3.1), pour trouver le vecteur de modification \mathbf{e} qui transforme le support \mathbf{x} en \mathbf{y} , on utilise la technique de *matrix embedding* en calculant le syndrome $\mathbf{m} - \mathbf{H}\mathbf{x}$:

$$\begin{aligned}
 \mathbf{H}\mathbf{e} = \mathbf{m} - \mathbf{H}\mathbf{x} &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
 &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \\
 &= \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}
 \end{aligned} \tag{3.11}$$

Si on applique le même raisonnement de la section précédente, le solution recherchée à notre problème est $\mathbf{e} = (0, 0, 1, 0, 0, 0, 0)$, mais puisque on ne peut modifier ni le bit n°3, ni le bit n°4, on a donc $e_3 = e_4 = 0$. Les colonnes 3 et 4 sont enlevées de la matrice de parité \mathbf{H} , et le système de résolution devient alors :

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_5 \\ e_6 \\ e_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad (3.12)$$

Pour ce genre de système à 5 inconnues, la solution recherchée, qui est de $\mathbf{e} = (1, 1, 0, 0, 0)$, est très facilement calculable en utilisant les techniques de combinaisons et de substitutions. De manière générale, la résolution de ce genre de systèmes est très complexe en temps et en espace mémoire; elle est cubique sur le nombre d'équation du système. En pratique, pour des systèmes plus importants, on utilise des matrices aléatoire de forme particulière dite "creuses"² afin de réduire la complexité de calcul du système.

3.3.3 Les codes STC

L'approche treillis (ou en anglais STC : *Syndrome Trellis Codes*), proposée par [Filler *et al.*, 2011], est une approche pratique reposant sur le principe de décodage par treillis (algorithme de Viterbi). Comme toutes les méthodes de codage par syndrome, le problème est de trouver le vecteur \mathbf{y} , représentant l'image stéganographiée, tel que lors de la réception le syndrome $\mathbf{H}\mathbf{y} = \mathbf{m}$. Pour ce faire, l'approche treillis utilise une matrice de parité \mathbf{H} de forme particulière; constituée de zéros et d'une sous-matrice binaire $\hat{\mathbf{H}}$ de taille $h \times w$ partagée entre l'émetteur et le récepteur. La matrice de parité \mathbf{H} est obtenue en plaçant m (qui est aussi la taille du message secret à insérer) copies de la sous-matrice $\hat{\mathbf{H}}$ une à coté de l'autre et en décalant à chaque fois d'une ligne en bas. Le reste de la matrice est mis à 0 (voir l'exemple sur la Figure 3.1). Pour trouver le vecteur \mathbf{y} recherché, tout en essayant de minimiser la distorsion apportée sur le support, l'approche STC fait appel à un graphe d'états et de branches étiquetées (appelé treillis), composé de m blocs correspondants aux différentes copies de $\hat{\mathbf{H}}$, et organisé sous forme d'une grille de $2h$ lignes et de $(w + 1)$ colonnes. Les états se trouvant entre chaque deux colonnes adjacentes sont organisés de façon à former un graphe biparti. Pour la résolution du problème posé, le stéganographe (l'émetteur) fournit d'abord à l'entrée du treillis une carte de détectabilité, qui associe à chaque pixel un coût reflétant l'impact de sa modification sur la sécurité. Cette

2. En mathématique une matrice creuse est une matrice contenant beaucoup de zéros.

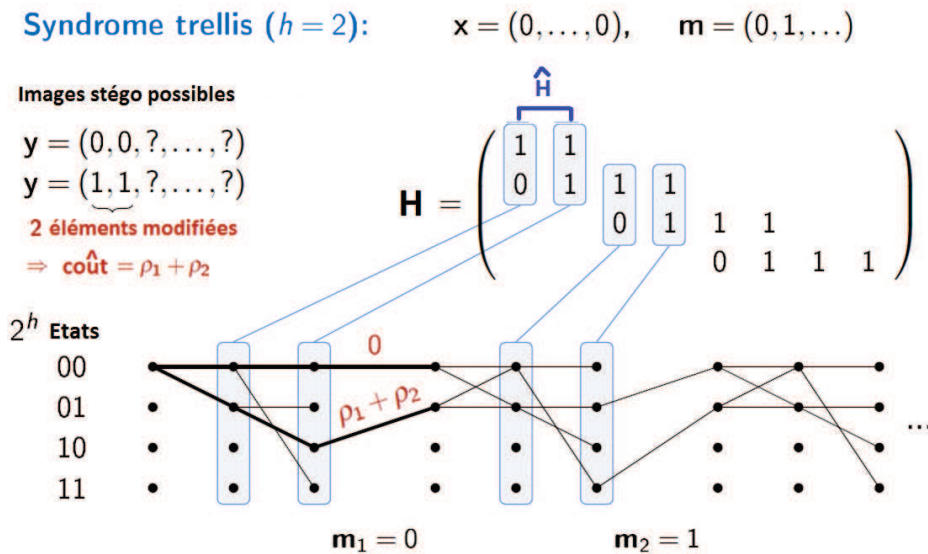


FIGURE 3.1 – Illustration du principe de fonctionnement de l’approche treillis (STC : Syndrome Trellis Codes).

carte de détectabilité est utilisée pour pondérer les branches du treillis, et tout chemin du treillis représente une solution à l’équation $\mathbf{H}\mathbf{y} = \mathbf{m}$. Au finale, La solution optimale est représentée par le chemin du plus faible coût (le chemin le plus court) qui est identifié par l’algorithme de Viterbi.

L’exemple sur la Figure 3.1 est une illustration simplifiée du principe de fonctionnement de l’approche treillis (STC).

Actuellement l’approche treillis est l’approche pratique la plus efficace en terme d’efficacité d’insertion ; c’est la méthode qui se rapproche le plus de la borne théorique.

3.4 Synthèse

Au cours de ce chapitre, nous nous sommes intéressés à la notion de codes correcteurs d'erreurs en stéganographie. Tout d'abord, nous avons présenté de façon générale l'utilité des codes correcteurs ainsi que leur principe de fonctionnement. Ensuite, nous avons discuté le rôle et l'apport des codes linéaires en stéganographie. En particulier, nous avons vu les techniques de codage par syndrome. Nous avons défini le concept de *Matrix embedding*, puis présenté quelques exemples de codes correcteurs utilisés en stéganographie actuelle. Nous avons également vu que grâce à ce puissant paradigme mathématique, les algorithmes de dissimulation de données actuels parviennent à augmenter considérablement leur efficacité d'insertion.

Dans le chapitre (chapitre 4) suivant nous nous intéresserons de près à la discipline duale de la stéganographie qui est *la stéganalyse*.

La Stéganalyse

Connais l'adversaire et surtout
connais toi toi-même et tu seras
invincible.

SUN TZU - *Philosophe chinois*

Préambule

La stéganalyse a pour objectif de détecter la présence de données dissimulées à l'aide d'un algorithme stéganographique. Elle est la discipline duale de la stéganographie. Dans ce chapitre, nous introduisons cette discipline, ainsi que les différentes méthodes de stéganalyse de la littérature. Dans un premier temps, nous présentons le principe général de la stéganalyse (section 4.1), ainsi que les différents scénarios possibles (section 4.2). Ensuite, nous étudions les différentes approches de stéganalyse : les méthodes d'analyse ciblées (section 4.3), les méthodes d'analyse aveugles (section 4.4), ainsi que les nouvelles méthodes de stéganalyse basées sur la théorie de décision et sur la théorie des jeux (section 4.5).

Sommaire

4.1	Attaque d'un schéma de stéganographie	50
4.2	Les principaux scénarios de la stéganalyse	51
4.3	Analyse ciblée d'un schéma stéganographique	54
4.4	Analyse aveugle d'un schéma stéganographique	59
4.5	La stéganalyse sous d'autres angles	70
4.6	Synthèse	72

4.1 Attaque d'un schéma de stéganographie

La stéganalyse (ou analyse stéganographique) est à la stéganographie, ce que représente la cryptanalyse à la cryptographie. Contrairement à la cryptographie qui a comme but de récupérer le message, ayant été préalablement crypté, sans connaissance de la clé, la stéganalyse n'a pas comme objectif initial d'extraire les données dissimulées à l'aide d'un algorithme stéganographique. Elle consiste uniquement en la détection de la présence des données cachées. En d'autres termes, la stéganalyse a pour but principal de détecter si un médium donné sert ou non de conteneur stéganographique. Dans le cas de la stéganographie par modification d'un médium dit empirique¹ (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 $\mathbf{x} = (x_1, \dots, x_n)$, le problème de détection de message secret peut-être représenté comme un test entre deux d'hypothèses :

$$\left\{ \begin{array}{l} H_0 : \mathbf{x} \sim P_C \quad \text{le support } \mathbf{x} \text{ ne contient pas de message caché (cover)} \\ H_1 : \mathbf{x} \sim P_S \quad \text{le support } \mathbf{x} \text{ contient un message caché (stego)} \end{array} \right. \quad (4.1)$$

Le stéganalyste, représenté par la gardienne Eve dans *le problème des prisonniers* [Simmons, 1983] (voir la section 2.2), doit donc décider entre ces deux hypothèses pour juger si oui ou non le médium est stéganographié.

Comme indiqué précédemment, 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. 4.1) ;
- *la stéganalyse à gardien actif* : peut également faire le test d'hypothèse (Eq. 4.1), 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,

1. Rappelons qu'un médium de couverture est dit empirique si son modèle statistique est partiellement ou totalement inconnu [Filler, 2011].

le stéganalyste peut contourner le message secret pour ses propres fins. Il peut même réintroduire un autre message falsifié. Pour ce type de stéganalyse, [Craver, 1998] a proposé un protocole de stéganographie à clé publique luttant contre la modification et la falsification du message secret.

Pour rappel, les travaux présentés dans ce manuscrit considèrent uniquement le cas de la stéganalyse à gardien passif, dans le cadre de l'utilisation de la stéganographie par modification appliquée aux images numériques naturelles (voir la section 2.3.3). Dans ce qui suit, nous présentons d'abord différents scénarios de la stéganalyse actuelle (section 4.2). Ensuite, nous passons en revue différentes approches d'analyse d'un schéma de stéganographie : les approches d'analyse ciblée (section 4.3), les approches d'analyse aveugles (section 4.4), et des nouvelles approches abordant la stéganalyse sous différents angles (section 4.5).

4.2 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.

4.2.1 Stéganalyse à clairvoyance

La stéganalyse à clairvoyance se place dans le contexte des principes annoncés par A. Kerckhoffs [Kerckhoffs, 1883], 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, pour ce scénario 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 H_0 et H_1 dans l'Eq. 4.1).

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 [Kodovský *et al.*, 2011] [Holub et Fridrich, 2012] ou en stéganalyse [Fridrich *et al.*, 2011a] [Fridrich et Kodovský, 2012] [Kodovský *et al.*, 2012].

Les travaux présentés dans ce manuscrit considèrent également ce premier scénario de stéganalyse à clairvoyance.

4.2.2 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* α 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 α_0 , pour juger de la présence ou non d'un message caché :

$$\begin{cases} H_0 & : \alpha \approx 0 \\ H_1 & : \alpha \geq \alpha_0 \end{cases}, \quad (4.2)$$

où α_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 [Fridrich *et al.*, 2003] [Miche *et al.*, 2010] [Guan *et al.*, 2011] [Pevný *et al.*, 2012], 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 [Kodovský et Fridrich, 2012a].

4.2.3 Stéganalyse universelle

Dans le cas de la stéganalyse universelle, la communication du message secret est également réalisée à travers une seule image. Pour ce scénario, 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 P_C ; ce qui revient en pratique à choisir entre les deux hypothèses suivantes :

$$\begin{cases} H_0 : \mathbf{x} \sim P_C & \text{la distribution de l'image } \mathbf{x} \text{ est proche d'une distribution } cover \\ H_1 : \mathbf{x} \not\sim P_C & \text{la distribution de l'image } \mathbf{x} \text{ est différente d'une distribution } cover \end{cases} \quad (4.3)$$

Dans ces conditions, il est très difficile pour le stéganalyste de monter une attaque efficace contre le schéma de dissimulation. Actuellement dans la littérature, il n'y a pas encore de solution satisfaisante pour ce scénario [Pevný et Fridrich, 2008]. Les pistes les plus pertinentes pour ce scénario sont probablement l'utilisation des vecteurs caractéristiques de haute dimension [Fridrich *et al.*, 2011b] [Fridrich et Kodovský, 2012], et l'emploi de mécanismes puissants de stéganalyse pour la gestion de nouveauté, tel que les SVM mono-classe [Pevný et Fridrich, 2008], les classifieurs multi-classes, ou les ensembles classifieurs pour la classification par fusion de votes [Kodovský *et al.*, 2012].

4.2.4 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 stéganalyste 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, [Fridrich *et al.*, 2011a] évoquent la possibilité de réaliser l'apprentissage sur une base contaminée, mais cela ne semble pas donner de bons résultats. [Gul et Kurugollu, 2011], 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*. [Lubenko et Ker, 2012a] [Lubenko et Ker, 2012b] 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. [Pasquet *et al.*, 2013] 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 [Kodovský et Fridrich, 2011]. Leur approche ne nécessite pas l'emploi d'une très grande base d'images.

4.2.5 Stéganalyse par mise en commun

La stéganalyse par mise en commun, ou *Pooled Steganalysis* en anglais [Ker, 2006], 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 stéganalyste 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 acteur² 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) quand 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 (*stego*). Dans le cadre de la stéganalyse par mise en commun à plusieurs acteurs, récemment [Ker et Pevný, 2011] [Ker et Pevný, 2012b] [Ker et Pevný, 2012a] 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. Le problème de stéganalyse par mise en commun est encore ouvert, les avancées dans les scénarios 4.2.3 et 4.2.4 pourraient également donner des éléments supplémentaires à la compréhension de ce scénario.

4.3 Analyse ciblée d'un schéma stéganographique

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 caractérisants 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. Dans ce qui suit, nous nous intéressons de près à ce genre

2. Le terme acteur, pour le scénario de stéganalyse par mise en commun, désigne tout entité physique ou morale qui communique et échange des données numériques sur le canal de communication surveillé.

d'attaques. Nous exposons d'abord les caractéristiques utilisées lors d'une attaque ciblée (section 4.3.1), puis nous présentons quelques méthodes classiques de stéganalyse ciblée de la littérature (section 4.3.2).

4.3.1 Caractéristiques utilisées

Comme mentionné précédemment, l'objectif d'une attaque ciblée est d'identifier les points faibles d'un algorithme stéganographique particulier, pour pouvoir par la suite détecter les images stéganographiées par cet algorithme. Pour ce faire, les méthodes ciblées visent d'abord un algorithme stéganographique donné, ensuite étudient : les étapes de son implémentation, la particularité du type et du format des images utilisées, la différence entre la statistique des images de couvertures et la statistique des images modifiées par cet algorithme. Le but de cette démarche est de trouver un ensemble de caractéristiques spécifiques à cet algorithme, permettant de discriminer les images de couverture des images stéganographiées. De manière générale, ces caractéristiques sont construites en suivant une de ces stratégies :

- Analyser le processus de dissimulation utilisé, pour essayer de trouver une faille particulière dans l'implémentation de l'algorithme stéganographique.
- Identifier les caractéristiques qui ont un changement prévisibles, lors de la modification du support de couverture hôte.
- Trouver le moyen d'estimer certaines caractéristiques statistiques, permettant de modéliser l'image de couverture originale, à partir de l'image stéganographiée interceptée.
- chercher une/des caractéristiques, spécifiques au format d'images, qui ont une valeur connue dans les images de couverture.

Dans ce qui suit nous présentons quelques exemples d'attaques pour illustrer le principe des méthodes d'analyse ciblée.

4.3.2 Quelques méthodes d'analyse ciblée

Afin d'illustrer le principe de fonctionnement d'une attaque ciblée, nous présentons dans cette section quelques exemples de méthodes de stéganalyse simples et classiques de la littérature.

L'analyse du χ^2

La méthode de stéganalyse χ^2 , développée par [Westfeld et Pfitzmann, 1999], est une attaque ciblée générale, qui peut être employée pour la détection de tout algorithme d'insertion utilisant le principe de stéganographie par substitution (*LSB Replacement*, voir sec-

tion 2.5.1). Cette attaque, pionnière dans le domaine, repose sur le test statistique du χ^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 surécriture des LSB réduit l'écart de fréquence entre des nuances de gris adjacentes au sens LSB.

En effet, la modification du LSB de chaque pixel va entraîner une modification de ± 1 sur celui-ci. Ce changement de ± 1 sur chaque pixel a pour conséquence de transformer un pixel de valeur $2i$ en un pixel de valeur $2i + 1$. De même un pixel de valeur $2i + 1$ deviendra un pixel de valeur $2i$. Ainsi, les fréquences de chaque élément composant une paire $(2i, 2i + 1)$ ont tendance à s'égaliser après l'insertion LSB (voir la figure 4.1).

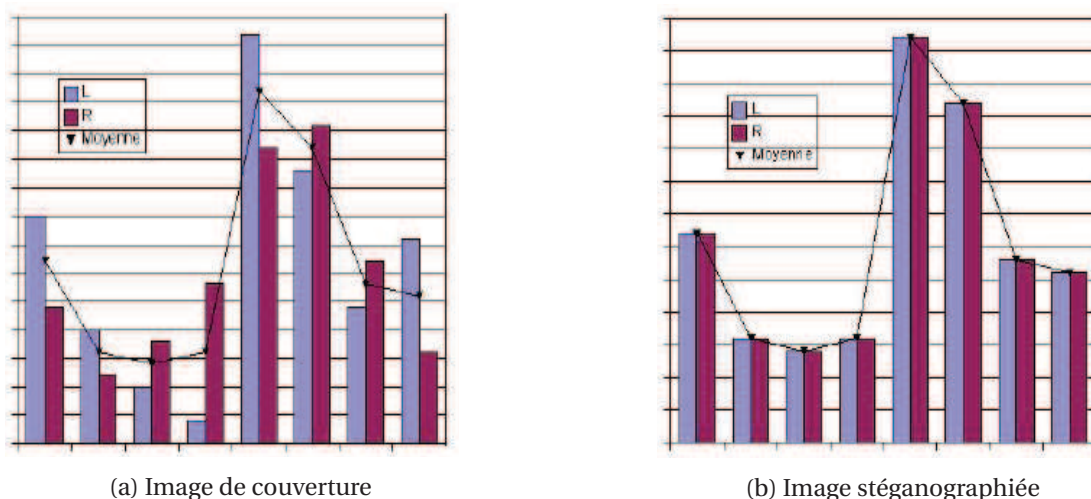


FIGURE 4.1 – Exemple d'un histogramme d'une image avant et après insertion LSB.

En pratique, si on prend l'exemple d'un histogramme présentant 20 pixels d'intensité 50, et 400 pixels d'intensité 51, alors, comme chaque pixel a une chance sur deux d'être modifié, environ 10 pixels d'intensité 50 deviendront d'intensité 51, et environ 200 pixels d'intensité 51 deviendront d'intensité 50. Ainsi au final sur l'histogramme de l'image modifiée, nous obtiendrons environ le même nombre de pixels pour les deux intensités 50 et 51 (soit à peu près 210 pixels).

Avec cet exemple, on comprend aisément que la fréquence théorique attendue, pour une paire de valeurs i , après l'insertion LSB d'un message, peut être calculée à partir de la moyenne des deux valeurs composant ce PoV :

$$n_i^* = \frac{n_{2i} + n_{2i+1}}{2}. \quad (4.4)$$

Pour juger de la présence d'un contenu caché dans une image donnée, les auteurs analysent la différence de distribution entre les valeurs calculées théoriquement et celle étant réellement présentes dans l'image. Pour déterminer la similarité entre les deux distributions, ils utilisent le test de χ^2 qui est défini comme suit :

$$\chi_{v-1}^2 = \sum_{i=1}^v \frac{(n_{2i} - n_i^*)^2}{n_i^*}, \quad (4.5)$$

avec n_{2i} la fréquence mesurée sur l'image analysée, n_i^* la fréquence théorique attendue après insertion du message (Eq. 4.4), et $v-1$ le degré de liberté qui correspond au nombre de paires de valeurs (PoV) définies précédemment.

La probabilité que les deux distributions soient identiques est donnée par la formule suivante :

$$p = 1 - \frac{1}{2^{\frac{v-1}{2}} \Gamma(\frac{v-1}{2})} \int_0^{\chi_{v-1}^2} e^{-\frac{x}{2}} x^{\frac{v-1}{2}-1} dx, \quad (4.6)$$

tel que Γ est la fonction gamma d'Euler.

La méthode χ^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 [Westfeld et Pfitzmann, 1999]). Plus tard dans la littérature, [Provos et Honeyman, 2001] ont adapté cette méthode aux algorithmes stéganographiques par substitution opérant dans le domaine transformé sur des images JPEG (tel que l'algorithme Jsteg [Upham, 1997]). Pour appliquer l'analyse χ^2 sur les images JPEG, les auteurs utilisent le même concept décrit ci-dessus, mais cette fois ci sur les coefficients DCT.

L'analyse χ^2 n'est efficace que quand il s'agit de détection de données cachées de manière 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 χ^2 ne donne pas de bons résultats. Pour résoudre le problème de l'insertion aléatoire, [Provos et Honeyman, 2001] proposent également, dans leur publication, d'adapter cette méthode pour la détection de contenu caché manière aléatoire. Pour cela, ils appliquent le test de χ^2 sur une fenêtre de taille plus petite que l'image, qui se déplace au fur et à mesure.

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 [Fridrich *et al.*, 2001], repose sur la classification des pixels, selon leur variation, en groupes distincts.

Pour une image interceptée $\mathbf{x} = (x_1, \dots, x_n) \triangleq \mathcal{I}^n$, en niveau de gris avec $\mathcal{I} = \{0, \dots, 255\}$, la première étape de la méthode de stéganalyse RS consiste à scinder l'image \mathbf{x} en groupes disjoints de n' pixels, $\mathcal{G} = \{x_1, \dots, x_{n'}\}$ avec $n' < n$. Une fois effectuée les auteurs utilisent une fonction de discrimination f (telle que celle définie dans l'Eq. 4.7) qui prend en compte la variation des pixels dans \mathcal{G} . Cette fonction de discrimination attribue un nombre réel à chacun des groupes. Plus le groupe des pixels est bruité, plus la valeur de la fonction f est grande.

$$f: \begin{array}{ccc} \mathcal{G} & \longmapsto & \mathbb{R} \\ x_1, \dots, x_{n'} & \longrightarrow & \sum_{i=1}^{n'-1} |x_{i+1} - x_i| \end{array} \quad (4.7)$$

Par la suite, des fonctions réversibles de permutations, $\{F_1, F_{-1}, F_0\}$, sont également définies par les auteurs sur les groupes de pixels. Ces fonctions consistent essentiellement en des permutations des nuances de gris. Ils simulent l'ajout de bruits à l'image source :

$$\begin{aligned} F_1 &: 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 254 \leftrightarrow 255 \\ F_{-1} &: -1 \leftrightarrow 0, 1 \leftrightarrow 2, \dots, 255 \leftrightarrow 256 \\ F_0 &: 1 \leftrightarrow 1, 2 \leftrightarrow 2, \dots, 255 \leftrightarrow 255 \end{aligned} \quad (4.8)$$

Pour appliquer ces fonctions de permutation aux différents groupes de pixels définis en Eq. 4.8, les auteurs utilisent un masque \mathbf{M} de taille $1 \times n'$, avec des valeurs comprises dans $\{-1, 0, 1\}$. Ceci étant défini, les auteurs classifient chaque groupe de pixels dans un des trois ensemble suivant :

$$\begin{aligned} \text{Groupe Régulier} &: \mathcal{G} \in \mathcal{R}_{\mathbf{M}} \Leftrightarrow f(F(\mathcal{G})) > f(\mathcal{G}) \\ \text{Groupe Singulier} &: \mathcal{G} \in \mathcal{S}_{\mathbf{M}} \Leftrightarrow f(F(\mathcal{G})) < f(\mathcal{G}) \\ \text{Groupe Inutilisable} &: \mathcal{G} \in \mathcal{U}_{\mathbf{M}} \Leftrightarrow f(F(\mathcal{G})) = f(\mathcal{G}) \end{aligned} \quad (4.9)$$

La méthode de stéganalyse RS se base sur le fait que pour une image de couverture (naturelle sans aucune modification) $\mathcal{R}_{\mathbf{M}} = \mathcal{R}_{-\mathbf{M}}$ et $\mathcal{S}_{\mathbf{M}} = \mathcal{S}_{-\mathbf{M}}$, alors que pour une image modifiée avec une insertion LSB, la différence $\mathcal{R}_{\mathbf{M}} - \mathcal{S}_{\mathbf{M}}$ tend à être réduite à mesure que la taille du message dissimulé augmente. Pour un message dissimulé avec 50% des LSB

permutés on a $\mathcal{R}_M \cong \mathcal{S}_M$. La stéganographie par substitution des LSB a l'effet inverse sur \mathcal{R}_{-M} et \mathcal{S}_{-M} . En effet, plus la taille du message à insérer est grande plus la différence entre les deux valeurs \mathcal{R}_{-M} et \mathcal{S}_{-M} augmente. Les auteurs calculent également l'intersection des courbes \mathcal{R}_M et \mathcal{S}_M pour estimer la taille du message inséré.

L'analyse RS s'appuie sur l'ajout de bruit, de ce fait elle 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 χ^2 . Cette combinaison de méthodes est efficace pour une large palette d'algorithmes stéganographique procédant par substitution des LSB.

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 [Westfeld, 2001] utilisant le principe de stéganographie par correspondance (LSB Matching (± 1)) sur les images JPEG (voir la section 2.5.2). Le principe de cette méthode d'analyse, proposé par [Fridrich *et al.*, 2002b], 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 [Fridrich *et al.*, 2002a] [Pevný et Fridrich, 2007a][Kodovský et Fridrich, 2009].

4.4 Analyse aveugle d'un schéma stéganographique

La stéganalyse aveugle est plus généraliste que la stéganalyse ciblée. Elle s'appuie principalement sur l'utilisation de mécanismes d'apprentissage et de classification supervisés. Ces mécanismes, n'étant pas spécifique à un algorithme particulier, peuvent être employés pour tout scénario en stéganalyse (à clairvoyance pour un algorithme préalablement connu, ou bien pour tout autre scénario où l'algorithme stéganographique employé n'est pas connu d'avance). L'objectif de ce genre d'approche est d'identifier ce qui caractérise une image de couverture d'une autre image stéganographiée, pour pouvoir discri-

miner 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, ensuite par l'emploi d'un classifieur particulier pour les différentes phases d'apprentissage et de classification. Notons au passage, que les méthodes de stéganalyse aveugle, exploitant les mécanismes d'apprentissage automatique, fonctionnent en trois temps différents [Fridrich, 2009] [Schaathun, 2012] :

- *Une phase d'apprentissage* : 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 *stégo*). Les images utilisées doivent être de la même dimension afin de respecter la loi de la racine carré (voir section 2.4.3). Lors de cette phase, le stéganalyste procède d'abord à l'extraction des caractéristiques de chacun des média composant la base d'images (voir la section 4.4.1). 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.
- *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. Généralement, lors des premiers tests, le stéganalyste connaît la classe des images qui ont été fournies, et peut ainsi comparer les résultats obtenus de la classification avec les résultats correctes, pour juger la performance du détecteur.
- *Une phase de mise en service* : dans laquelle le détecteur construit est mis en situation réelle. Lors de cette phase, le détecteur automatique, dans sa version finale, est placé sur un canal de communication donné, afin d'empêcher toute communication secrète illicite.

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.

Les travaux présentées dans ce manuscrit s'intéressent particulièrement aux méthodes de stéganalyse aveugle par apprentissage. Dans cette section, nous décrivons d'abord la méthode de calcul des caractéristiques utilisées en stéganalyse aveugle actuelle (section 4.4.1), ensuite nous présentons quelques outils d'apprentissage et de classification (section 4.4.2).

4.4.1 Caractéristiques utilisées

En stéganographie aveugle, l'extraction des caractéristiques, jugées pertinentes pour la discrimination des classes, est généralement l'étape la plus cruciale. Elle permet non seulement de réduire l'espace de recherche du classifieur choisit par le stéganalyste, mais également une bonne séparation des classes recherchées (dans notre cas la classe *cover* et la classe *stego*). De ce fait, il est important de choisir des caractéristiques qui discriminent au mieux les images de couvertures des images stéganographiées. Les caractéristiques extraites doivent être sensibles au changement stéganographique, et insensibles au contenu de l'image. Dans ce contexte, plusieurs travaux ont été menés que ce soit dans le domaine spatial : SPAM-[Pevný *et al.*, 2010], SRM-[Fridrich et Kodovský, 2012], PSRM-[Holub *et al.*, 2013], ou dans le domaine transformé JPEG : CCPEV-[Pevný et Fridrich, 2007b], CCJRM-[Kodovský et Fridrich, 2012b].

Dans ce manuscrit, nous nous focalisons principalement sur l'extraction de caractéristiques dans le domaine spatial. Nous présentons dans ce qui suit une méthodologie générique pour l'extraction de vecteurs caractéristiques, inspirée du *Spatial Rich Model* [Fridrich et Kodovský, 2012] :

1. Calcul des résidus

Les techniques de dissimulation adaptatives introduisent peu de modifications sur le support hôte (revoir la section 2.6). Elles insèrent le message secret dans les régions qui sont difficiles à détecter (telles que les régions bruitées et texturées). Une fois inséré, le message peut être considéré comme un bruit ajouté à l'image de couverture. De ce fait, il est plus intéressant de modéliser le bruit de l'image de couverture plutôt que son contenu [Katzenbeisser et Petitcolas, 2000]. Les méthodes de stéganalyse actuelles utilisent ce principe pour déceler la présence d'un message caché. Afin d'extraire et modéliser le bruit résiduel, ils utilisent différents filtres passe-haut (linéaires, et non-linéaires) HOLMES-[Fridrich *et al.*, 2011b] SRM-[Fridrich et Kodovský, 2012] PSRM-[Holub *et al.*, 2013].

Pour une image de couverture $\mathbf{x} = (x_1, \dots, x_n)$, notée également $\mathbf{X} = (X_{ij}) \in \mathbb{R}^{n_1 \times n_2}$, le bruit résiduel $\mathbf{R} = (R_{ij}) \in \mathbb{R}^{n_1 \times n_2}$ est calculé par plusieurs filtres passe-haut différents linéaires et non-linéaires qui peuvent être formulés comme suit :

$$R_{ij} = \hat{X}_{ij}(\mathcal{N}_{ij}) - cX_{ij}, \quad (4.10)$$

avec \mathcal{N}_{ij} le voisinage du pixel X_{ij} , ($X_{ij} \notin \mathcal{N}_{ij}$), $c \in \mathbb{N}$ l'ordre du bruit résiduel, et \hat{X}_{ij} une fonction de prédiction qui estime la valeur du pixel de couverture X_{ij} à partir de son voisinage \mathcal{N}_{ij} . L'ensemble des pixels utilisés pour le calcul du résidu est appelé *clique* et leur cardinalité (nommée *span* en anglais) est notée s .

Dans la littérature, il existe différents types de caractéristiques basées sur le calcul de résidus. Par exemple, les caractéristiques SPAM, proposées par [Pevný *et al.*, 2010], utilisent pour le calcul des résidus de premier ordre le filtre passe-haut $R_{ij} = X_{i,j+1} - X_{ij}$ (avec $s = 2$), et pour les résidus de second ordre le filtre $R_{ij} = X_{i,j-1} + X_{i,j+1} - 2X_{ij}$ (avec $s = 3$). Pour les caractéristiques MINMAX, proposées par [Fridrich *et al.*, 2011a], les auteurs utilisent différents filtres linéaires produisant ainsi plusieurs matrices résiduelles. Pour introduire une certaine non-linéarité dans leur calcul de résidus, ils utilisent les opérateurs '*min*' et '*max*' sur les matrices résiduelles obtenues par les différents filtres pour construire le bruit résiduel final.

2. Troncature et Quantification

Une fois calculé, le bruit résiduel est ensuite quantifié puis tronqué :

$$R_{ij} \leftarrow \text{trunc}_T(\text{round}(R_{ij}/q)), \quad (4.11)$$

tel que $q \in \mathbb{R}_0^+$ est le pas de quantification, et $\text{trunc}_T(x)$ une fonction de troncature définie par :

$$\text{trunc}_T(x) = \begin{cases} x & \text{when } x \in [-T, T], \\ T \text{ sign}(x) & \text{otherwise.} \end{cases} \quad (4.12)$$

La quantification est une étape importante, car elle permet d'analyser différentes gammes d'amplitude de bruit (faible amplitude : aplat, moyenne et haute amplitude : bordures et zones texturées). La troncature quand à elle permet de réduire la plage des valeurs du bruit résiduel (en prenant un petit T). L'objectif est d'obtenir des matrices de co-occurrence bien "peuplées". Autrement dit, des matrices de co-occurrence avec un nombre suffisant d'échantillons (voir la section suivante sur les matrices de co-occurrence).

3. Les matrices de co-occurrence

La troisième étape pour le calcul des caractéristiques est celle du calcul des matrices de co-occurrence sur un voisinage. Lors de cette étape, le stéganalyste calcule (par exemple dans les directions : horizontale, verticale, ainsi que les deux diagonales principales majeure et mineure), les probabilités d'apparition des m -uplets dans les matrices résidus issues de la deuxième étape (après quantification et troncature).

Soit C^h la matrice de co-occurrence horizontale, de dimension $(2T + 1)^m$, donnée par :

$$C_{d_1 \dots d_m}^h = \Pr(R_{ij} = d_1 \wedge \dots \wedge R_{i,j+m-1} = d_m), \quad d_1 \dots d_m \in \{-T, \dots, T\}, \quad (4.13)$$

avec m l'ordre de co-occurrence. Respectivement, les matrices de co-occurrence C^v, C^d, C^m , pour verticale, diagonale majeure, et diagonale mineure, sont également définies de la même manière. Notons au passage qu'il est également possible de calculer d'autres types de matrices de co-occurrence. Par exemple, au lieu de calculer les co-occurrence de résidus voisins sur la même ligne, on peut calculer les co-occurrences de m -uplets de forme carrée. Les articles suivants [Fridrich *et al.*, 2011a] [Fridrich *et al.*, 2011b] [Fridrich et Kodovský, 2012] présentent cela plus en détails.

4. La fusion des caractéristiques

Les méthodes de stéganalyse récentes utilisent des vecteurs caractéristiques de plus en plus grands. Pour construire le vecteur caractéristique final, $f \in \mathbb{R}^d$, de haute dimension, d , ces méthodes fusionnent les différentes caractéristiques issues des différentes matrices de co-occurrence et des différents filtres. Une fois construit, le vecteur caractéristique final, f , peut alors être utilisé pour caractériser une image lors du mécanisme d'apprentissage puis de classification. Dans la littérature actuelle il existe plusieurs vecteurs caractéristiques de haute dimension. Parmi ces vecteurs caractéristiques on retrouve le *Spatial domain Rich Model* (SRM) de dimension $d = 34671$ proposé par [Fridrich et Kodovský, 2012].

4.4.2 Quelques outils pour la classification

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

Séparateurs à vaste marge (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 supervisé, introduit par V. Vapnik en 1995 [Vapnik, 1995] [Cortes et Vapnik, 1995]. 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.

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*.

Pour un ensemble de données linéairement séparables (l observations (\mathbf{x}_i, f_i, c_i) avec $i = 1, \dots, l$, $f_i \in \mathbb{R}^d$ le vecteur caractéristique associé à l'image $\mathbf{x}_i \in \mathbb{R}^n$, et $c_i \in \{+1, -1\}$ la classe correspondante), doté d'une fonction de discrimination de forme suivante :

$$\phi(\mathbf{f}) = \text{signe}((\mathbf{f} \cdot \mathbf{w}) + b), \quad (4.14)$$

avec \mathbf{w} un vecteur de dimensions m , et b un scalaire, le problème de classification SVM consiste à trouver l'hyperplan séparateur optimale. Autrement dit, trouver l'hyperplan séparateur qui maximise la marge SVM, tout en classifiant correctement les données. De manière formelle, cela se traduit par la résolution du problème d'optimisation suivant :

$$\begin{cases} [\mathbf{f}, b] = \arg \max_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}\|^2, \\ \text{sous la contrainte} \\ c_i((\mathbf{f} \cdot \mathbf{w}) + b) \geq 1 \quad \forall i \in \{1, \dots, l\} \end{cases} \quad (4.15)$$

La Figure 4.2 illustre un exemple de discrimination linéairement séparable pour un classifieur à marge maximale. Sur la Figure, l'hyperplan séparateur optimal est représenté par la droite d'équation $\mathbf{w}\mathbf{f} + b = 0$.

En pratique, la plupart des données réelles ne sont pas linéairement séparables, à cause de la présence du bruit dans les observations. Dans un tel cas, l'utilisation du classifieur à marge maximale, sous sa forme précédente (voir les équations 4.14 et 4.15), ne peut être envisagée et doit être donc adaptée. Afin de surmonter ce problème, 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 4.3). 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 [Hasan et Boris, 2006] [Frezza-Buet, 2012]. Cette transformation

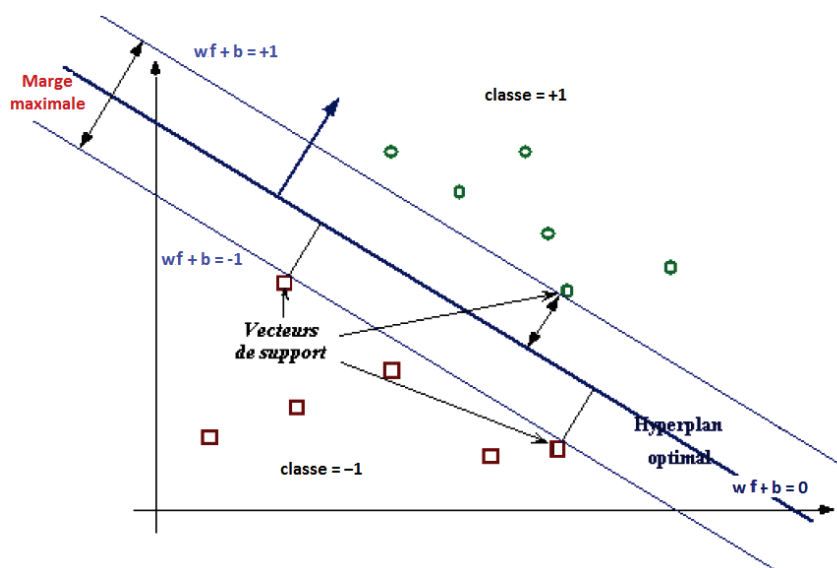


FIGURE 4.2 – Illustration du principe de fonctionnement d'un SVM binaire dans le cas d'un problème linéairement séparable.

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 :

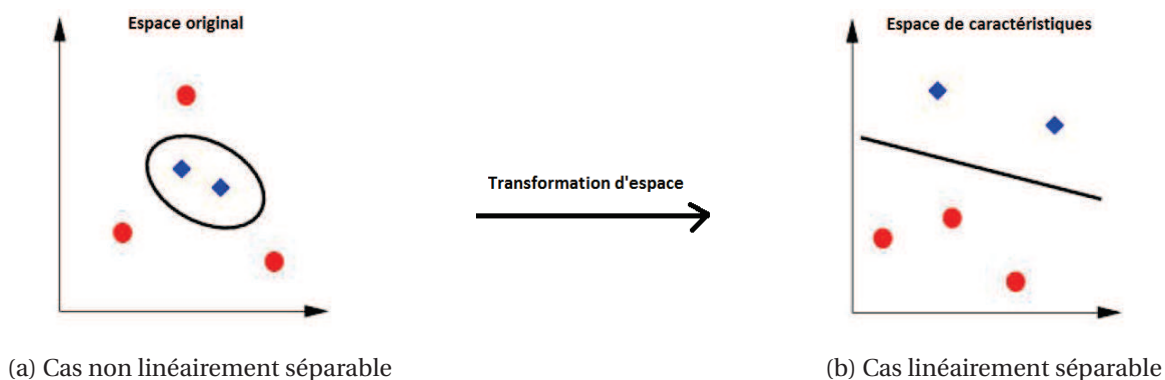


FIGURE 4.3 – Exemple illustrant le principe de résolution, pour un SVM, dans le cas où les données sont non-linéairement séparables.

$$\hat{\alpha} = \arg \max_{\alpha} \left(\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j c_i c_j K(\mathbf{f}_i, \mathbf{f}_j) \right), \quad (4.16)$$

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

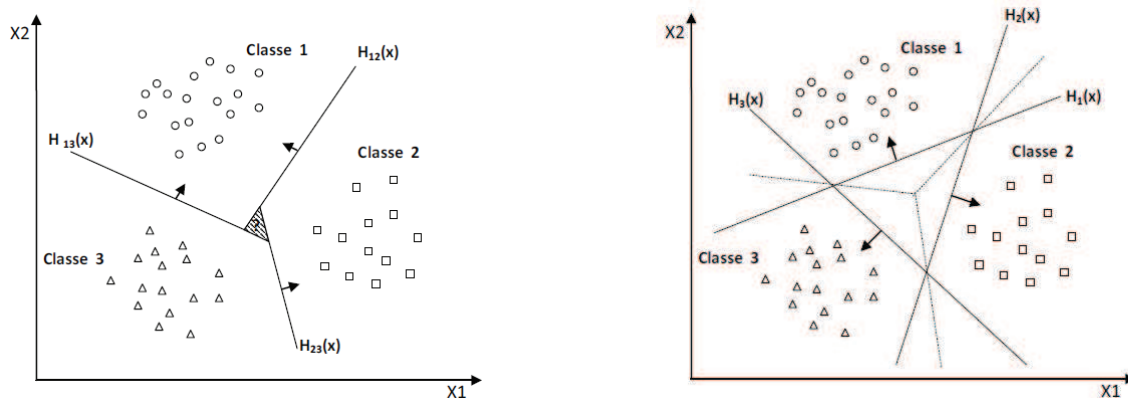
$$K : \begin{array}{ll} \mathbb{R}^d \times \mathbb{R}^d & \rightarrow \mathbb{R} \\ \mathbf{f}_i, \mathbf{f}_j & \rightarrow k(\mathbf{f}_i, \mathbf{f}_j) \end{array} \quad (4.17)$$

La fonction de décision quand à elle devient :

$$\phi(\mathbf{f}) = \sum_{i=1}^l \alpha_i c_i k(\mathbf{f}_i, \mathbf{f}) + b. \quad (4.18)$$

b) SVM multiclass

Dans le cas de classification multiclass, 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 multiclass consiste à réduire le



(a) Approche une classe contre une autre (1vs1)

(b) Approche une classe contre le reste (1vsR)

FIGURE 4.4 – Exemple illustrant quelques approches de décomposition pour un classifieur SVM multiclass [Djeffal, 2012].

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 4.4). Le lecteur intéressé par ces approches pourra consulter les références suivantes [Guermeur, 2007] [Djeffal, 2012].

Notons au passage qu'il est envisageable d'utiliser les SVMs multiclassés pour le scénario de stéganalyse universelle avec ou sans *cover-source mismatch*. Parmi les travaux qui ont exploré cette piste, on retrouve [Pevný et Fridrich, 2006, Pevný et Fridrich, 2007a] [Dong *et al.*, 2009].

c) SVM monoclasse (OC-SVM)

Dans les machines à vecteur support binaires ainsi que les SVMs multiclassés 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.

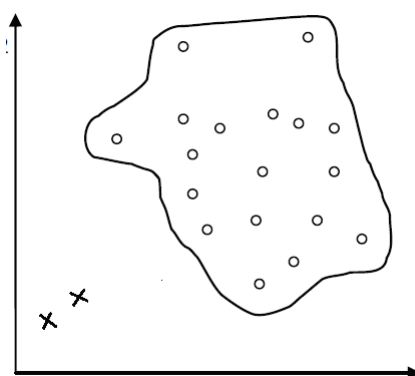


FIGURE 4.5 – Illustration du principe de fonctionnement d'un classifieur SVM monoclasse.

Pour la classification SVM monoclasse, 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.

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 [B. Roue, 2005] [Pevný, 2008] [Pevný *et al.*, 2010]. De nos jours, avec l'augmentation de la dimension des espaces caractéristiques en stéganalyse, les machines à vecteurs supports (SVMs) ne sont plus adaptés. Pour cela, de nouveaux outils pour l'apprentissage et la classification ont été proposés comme alternative. Parmi ces outils, nous retrouvons l'ensemble de classifieurs FLD [Kodovský et Fridrich, 2011] [Kodovský *et al.*, 2012], que nous présentons maintenant.

Les schémas stéganographiques modernes tel que HUGO [Pevný *et al.*, 2010] ont tendance à utiliser des espaces de caractéristiques de plus en plus grand (voir section 2.6.3), ce qui constitue un vrai problème pour la stéganalyse. Afin de résoudre ce problème, [Kodovský et Fridrich, 2011] 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 $\mathcal{F} = \{F_1, \dots, F_L\}$ de classifieurs FLD (*Fisher Linear Discriminant*) binaires.

Soit $\mathcal{A} = \{\mathbf{f}_i, c_i\}_{i=1}^N$ une base d'apprentissage composée de N observations appartenant aux deux classes d'images *cover* et *stégo*, avec $\mathbf{f}_i \in \mathbb{R}^d$ un vecteur caractéristique de grande dimension d , caractérisant la $i^{\text{ème}}$ image, et $c_i \in \{-1, +1\}$ la classe qui lui est associée (-1 pour une image *cover*, et +1 pour une image *stégo*).

Lors de *la phase d'apprentissage*, chaque classifieur FLD apprend à associer correctement à chaque observation \mathbf{f}_i le numéro de la classe c_i à laquelle elle appartient :

$$\begin{aligned} F_l: \mathbb{R}^d &\rightarrow \{-1, +1\} \\ \mathbf{f}_i &\rightarrow F_l(\mathbf{f}_i) \end{aligned} \tag{4.19}$$

Pour cela, chaque classifieur FLD utilise la base d'apprentissage \mathcal{A} , afin de calculer le vecteur \mathbf{w}_l orthogonal à l'hyper-plan séparant les observations de la classe *cover* de ceux de la classe *stégo*.

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 d_{red} avec ($d_{red} \ll d$). En pratique, avant la phase d'apprentissage chaque classifieur FLD choisit pseudo-aléatoirement un sous-ensemble de caractéristiques (d_{red} caractéristiques) à partir de chaque vecteur caractéristique $f_i \in \mathbb{R}^d$.

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, à travers la formule suivante [Kodovský et Fridrich, 2011] :

$$\begin{aligned} R: \quad \mathbb{R}^d &\rightarrow \{-1, +1\} \\ f &\rightarrow R(f), \end{aligned} \quad (4.20)$$

tel que :

$$R(f) = \begin{cases} +1 & \text{si } \sum_{l=1}^L F_l(f) > 0, \\ -1 & \text{sinon.} \end{cases} \quad (4.21)$$

Lors de l'apprentissage, le seuil de décision de chaque classifieur FLD est ajusté, afin de minimiser l'erreur totale de détection (P_E) sur les données d'apprentissage. Dans cet esprit, [Kodovský et al., 2012] proposent de définir la probabilité d'erreur de détection P_E comme suit :

$$P_E = \min_{P_{FP}} \frac{1}{2} (P_{FP} + P_{FN}(P_{FP})), \quad (4.22)$$

avec P_{FP} la probabilité de faux positif et P_{FN} la probabilité de faux négatif. Plus la probabilité d'erreur (P_E) est petite, plus la classification en deux classes est meilleure. Dans ce manuscrit, nous utiliserons également cette mesure pour évaluer l'efficacité de détection de l'ensemble de classifieurs FLD.

Notons au passage qu'il est possible d'améliorer encore plus les performances de l'ensemble de classifieurs FLD. Dans [Chaumont et Kouider, 2012], nous avons proposé quelques améliorations sur cet ensemble de classifieurs, permettant d'augmenter son efficacité de détection tout en gardant la même complexité de calcul.

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 [Lubenko et Ker, 2012a] [Lubenko et Ker, 2012b], ils ont proposé un nouvel classifieur qui est *l'ensemble average perceptron*. Un peu plus

tard, pour ce même scénario, les auteurs de [Pasquet *et al.*, 2013] ont montré qu'une adaptation de l'ensemble de classifieurs FLD de Kodovský permettrait d'obtenir de meilleurs résultats.

4.5 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 :

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

Comme mentionné précédemment (section 4.1), 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 $\mathbf{x} = \{x_1, \dots, x_n\}$ est distribuée selon une loi P_C définissant l'hypothèse nulle H_0 ou bien selon une loi P_S définissant l'hypothèse alternative H_1 (voir l'Eq. 4.1). Le problème principal est alors de formuler une définition statistique précise de l'hypothèse H_0 (l'image \mathbf{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 final du détecteur statistique est obtenue par le biais d'une fonction π qui associe à l'image analysée \mathbf{x} la classe à laquelle elle appartient (-1 pour une image de couverture et $+1$ pour une image stéganographiée, voir Définitions 2 et 3).

Définition 2 (Test statistique [Zitzmann, 2013]). *On appelle test statistique (ou bien règle de décision) entre deux hypothèses H_0 et H_1 toute application surjective et mesurable $\delta : \mathbb{R}^n \rightarrow \{H_0, H_1\}$. Autrement dit, définir un test statistique binaire δ à partitionner l'espace des observations \mathbb{R}^n en deux ensembles disjoints (également appelé régions d'acceptation) : Ω_0 et Ω_1 . Nos avons alors $\mathbf{x} \in \Omega_0$ si $\delta(\mathbf{x}) = H_0$, et $\mathbf{x} \in \Omega_1$ si $\delta(\mathbf{x}) = H_1$.*

Définition 3 (Fonction de décision [Zitzmann, 2013]). *Une fonction de décision π associée au test δ est une application $\pi : \mathbb{R}^n \rightarrow \{-1, +1\}$ définie par :*

$$\pi = \begin{cases} -1 & \text{si } \mathbf{x} \in \Omega_0 \\ +1 & \text{si } \mathbf{x} \in \Omega_1 \end{cases}$$

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 $\beta(\delta)$.

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 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 [Fillatre, 2011] [Cogranne, 2011] [Zitzmann, 2013].

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 [van Damme, 1991]. Pour un contexte de stéganographie/stéganalyse, [Ettinger, 1998] 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, [Schöttle et Böhme, 2012] ont développé la première méthode pratique, basée sur la théorie des jeux, pour l'insertion adaptative d'un message secret. Pour plus de détails sur ce sujet, le lecteur est invité à consulter les différentes références bibliographiques citées.

4.6 Synthèse

Dans ce chapitre, nous avons présenté un état de l'art des principaux concepts de la stéganalyse. Tout d'abord, nous avons défini le problème de détection d'un message caché, ainsi que différents scénarios possibles. Ensuite, nous avons passé en revue différentes méthodes d'attaques en stéganalyse. En particulier, nous avons vu les méthodes de stéganalyse ciblée, les méthodes de stéganalyse aveugles ainsi que les méthodes de stéganalyses basées sur la théorie de la décision, et les méthodes basées sur la théorie des jeux. Lors de la description des méthodes aveugles, 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 [Kodovský et Fridrich, 2011]. Dans le chapitre suivant (chapitre 5), nous utilisons ce nouveau concept non pas pour la détection mais pour la dissimulation. Nous proposons un nouveau schéma de dissimulation stéganographique, qui exploite les informations de l'ensemble de classifieurs FLD comme oracle lors de l'insertion du message. Dans le chapitre 6, l'ensemble de classifieurs FLD est également exploité mais cette fois-ci pour la conception d'une nouvelle mesure de sécurité pour la sélection des images stéganographiées lors de la transmission.

Deuxième partie

Contributions

La Stéganographie Adaptative par Oracle (ASO)

Découvrir c'est bien souvent dévoiler quelque chose qui a toujours été là, mais que l'habitude cachait à nos regards.

ARTHUR KOESTLER - *Écrivain*

Préambule

La fin de l'année 2010 marqua le début d'une nouvelle aire en stéganographie. L'apparition d'algorithmes stéganographiques adaptatifs basés sur le principe de minimisation d'impact d'insertions par l'utilisation d'une carte de détectabilité a constitué une grande révolution. Dans la continuité de ces travaux, nous proposons au cours de ce chapitre un nouvel algorithme de stéganographie adaptative dans le domaine spatial (nommé ASO : Adaptive Steganography by Oracle) basé sur l'utilisation d'un oracle pour le calcul de la carte de détectabilité. La section 5.1 expose les différentes raisons qui nous ont poussé à développer cette nouvelle méthode de dissimulation. La section 5.2 est consacrée à la description générale de l'algorithme ASO. La méthode de calcul de la carte de détectabilité ainsi que la méthode d'insertion du message secret sont décrites dans les sections 5.3 et 5.4. La section 5.5 présente les détails techniques de l'implémentation ASO. Enfin, la section 5.6 est dédiée à la présentation et à la discussion des résultats obtenus.

Sommaire

5.1	Motivation	76
5.2	Vue d'ensemble de la méthode ASO	78
5.3	Calcul de la carte de détectabilité	79
5.4	Insertion du message secret	84
5.5	Détails techniques de l'implémentation de ASO	85
5.6	Tests et résultats	86
5.7	Conclusion	96

5.1 Motivation

Comme mentionné précédemment (section 2.6), l'objectif des algorithmes de stéganographie par minimisation d'impact d'insertion est d'apporter le minimum d'altérations au support hôte $\mathbf{x} = (x_1, \dots, x_n)$, afin de produire le *stégo* $\mathbf{y} = (y_1, \dots, y_n)$ objet qui communiquera le message secret $\mathbf{m} = (m_1, \dots, m_n)$. Pour ce faire, les méthodes d'insertion adaptatives récentes utilisent une carte de détectabilité $\rho \in \mathbb{R}_+^n$, qui attribue à chaque élément de couverture x_i , avec $i \in \{1, \dots, n\}$, un coût de détectabilité $\rho_i \in \mathbb{R}_+$ modélisant l'impact sur la sécurité dû à la modification de cet élément. Cette carte de détectabilité ρ est généralement associée à une fonction de distorsion $D(\mathbf{x}, \mathbf{y})$ qui est minimisée sous la contrainte d'un *payload* fixe. Le problème majeur de ce genre d'approches est donc de trouver le moyen le plus efficace permettant de calculer les coûts ρ_i qui reflètent au mieux la détectabilité statistique lors de la modification.

Pour résoudre ce problème, la plupart des méthodes de stéganographie adaptatives actuelles de l'état de l'art, lors de l'insertion, ne prennent en compte que les informations de l'image de couverture courante qui sera utilisée pour la dissimulation. Les informations de l'ensemble de la base d'images utilisée par l'émetteur ne sont pas exploitées. Pour rappel, l'algorithme F5 [Westfeld, 2001] considère un coût de détectabilité identique pour tous les éléments de couverture ($\rho_i = 1$). L'algorithme nsF5 [Fridrich *et al.*, 2005] fait appel à la technique de papier mouillé, avec $\rho_i \in \{1, \infty\}$, pour empêcher la modification des zones sensibles à l'insertion. L'algorithme HUGO [Pevný *et al.*, 2010] utilise une carte de détectabilité variable $\rho_i \in \mathbb{R}_+$ calculée à partir de caractéristiques de haute dimension tirées de l'image de couverture. Les caractéristiques utilisées correspondent aux probabilités conditionnelles en chaque pixel de l'image filtrée. L'algorithme MOD [Kodovský *et al.*, 2011] utilise pour l'insertion du message secret une carte de détectabilité paramétrique $\rho_i \in [0, \infty]$. Les paramètres de la carte menant au meilleur niveau de sécurité sont calculés grâce à l'algorithme d'optimisation *downhill simplex*. Le critère utilisé pour juger à chaque itération du niveau de sécurité atteint est la taille de la marge SVM utilisé pour la vérification. Les algorithmes WOW [Holub et Fridrich, 2012] et UNIWARD [Holub et Fridrich, 2013] exploitent les informations du domaine d'ondelettes de l'image de couverture, et utilisent des filtres directionnels pour le calcul de leur carte de détectabilité. L'algorithme UNIWARD [Holub et Fridrich, 2013] utilise une carte $\rho = \{\rho_i \in [0, \infty]_{i=1}^n\}$, calculée à partir de la variation relative entre les coefficients d'ondelettes résiduels de l'image de couverture et celle de l'image stéganographiée. L'algorithme WOW [Holub et Fridrich, 2012], quand à lui, utilise une carte $\rho = \{\rho_i \in [0, \infty]_{i=1}^n\}$, calculée à partir de l'agrégation de la différence pondérée entre les coefficients d'ondelettes résiduels de l'image de couverture et de l'image stéganographiée. En attribuant un $\rho_i = \infty$ aux endroits prévisibles à l'insertion, la méthode empêche la modification de ces sites lors de la dissimulation. Le problème de calcul de la carte de détectabilité, modélisant au mieux l'impact de modification sur la sécurité statistique du support hôte, est encore un véritable challenge.

Dans ce chapitre, nous proposons un nouveau schéma stéganographique adaptatif dans le domaine spatial, basé sur l'utilisation d'un oracle pour le calcul de la carte de détectabilité. L'originalité de notre approche (ASO : *Adaptive Steganography by Oracle*) est que lors du calcul de la carte de détectabilité, l'approche proposée ne prend pas en compte uniquement le modèle de distribution de l'image de couverture courante que l'on souhaite stéganographier, mais également le modèle de toute la base de données utilisée par l'émetteur. Ceci permet de préserver les deux distributions à la fois, introduisant ainsi une nouvelle philosophie de dissimulation. Contrairement à l'approche MOD proposée par [Filler et Fridrich, 2011], qui utilise une méthode paramétrique pour réduire la marge SVM séparant la classe *cover* de la classe *stégo*, nous proposons une méthode de calcul de carte de détectabilité non paramétrique, qui utilise l'ensemble de classifieurs FLD proposé par [Kodovský et al., 2012] comme oracle. Nous exploitons ainsi à la fois les informations de l'image de couverture courante et les informations tirées de la base entière des images de couvertures de l'émetteur. Par ailleurs, là où MOD échoue, l'approche ASO parvient à résoudre le problème de complexité face à l'utilisation de vecteurs caractéristiques de grande dimension. En effet, contrairement à MOD [Filler et Fridrich, 2011], ASO garde une bonne stabilité numérique, et ce même avec l'augmentation des espaces caractéristiques.

Soulignons au passage, que le schéma ASO ne peut en aucun cas être confondu avec l'approche FCM¹ proposée par [Kodovský et al., 2008]. En effet, les deux algorithmes de dissimulation ASO et FCM sont très différents. L'approche FCM a pour principe de préserver le modèle de distribution de l'image de couverture originale, grâce à la restauration des caractéristiques du modèle utilisé. Lors de l'insertion, les coefficients DCT de l'image JPEG de couverture sont répartis en deux ensembles disjoints. Le premier ensemble de coefficients est utilisé pour la dissimulation du message secret, tandis que le second ensemble est utilisé pour la restauration et la modification du vecteur de caractéristiques. Contrairement à FCM, l'approche ASO a pour objectif de rapprocher la distribution de l'image stéganographiée vers une distribution *cover*. Pour cela, ASO ne procède pas à une restauration des caractéristiques comme FCM. Pour ASO, les coûts de détectabilité, de la carte utilisée pour l'insertion, sont calculés de manière à favoriser la modification des pixels impliquant un déplacement du vecteur caractéristique de l'image stéganographiée vers la distribution *cover*. À travers notre approche ASO, nous présentons une méta-méthode générique de stéganographie basée oracle, qui peut être adaptée à n'importe quel vecteur caractéristique. La complétude² de ASO peut être facilement améliorée et étendue, ceci en utilisant des caractéristiques mieux choisies et plus couvrantes.

1. FCM : *Feature Correction Method*.

2. Un modèle d'images est dit complet si l'ensemble de ses caractéristiques décrivent (couvrent) complètement la nature et les variations des images naturelles. Lorsqu'il s'agit d'un schéma stéganographique, on dit qu'il est complet si son processus d'insertion pervertit la globalité du vecteur caractéristique utilisé, ce qui en pratique a pour conséquence de rendre le schéma indétectable face à toute attaque par d'autres caractéristiques. Pour plus de détails, la notion de complétude est très bien expliquée dans [Kodovský et al., 2008].

Dans ce qui suit, nous présentons d'abord le schéma général du fonctionnement de l'approche ASO (section 5.2). Puis, nous décrivons en détails les différentes étapes de dissimulation du processus ASO : le calcul de la carte de détectabilité (section 5.3), l'insertion du message secret (en section 5.4), et les détails techniques pour l'implémentation (section 5.5). Ensuite, nous présentons et analysons les différents résultats obtenus (section 5.6). Enfin, nous nous exposons nos conclusions et clôturons ce chapitre en section 5.7.

5.2 Vue d'ensemble de la méthode ASO

L'algorithme de stéganographie adaptative par oracle (ASO) que nous proposons est une nouvelle philosophie de dissimulation de données, qui repose sur l'utilisation d'un oracle pour l'insertion du message. Cet oracle fait appel à l'ensemble de classifieurs FLD (voir la section 4.4.2) pour calculer la carte de détectabilité. L'objectif est d'acquérir le plus d'informations possibles sur le processus d'apprentissage sur la base de données utilisée par l'émetteur afin d'augmenter la sécurité du processus d'insertion. Le schéma proposé est un schéma itératif qui préserve à la fois la distribution de l'image de couverture courante et la distribution de la base d'images utilisateur.

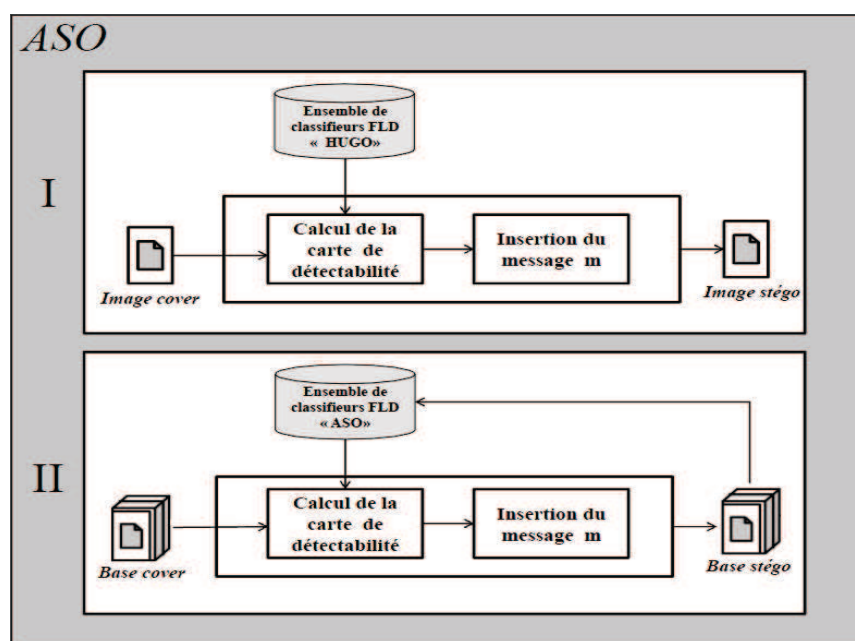


FIGURE 5.1 – Schéma de Stéganographie Adaptative par Oracle (ASO).

Comme l'illustre la Figure 5.1 représentant le mode de fonctionnement de ASO, le schéma de stéganographie adaptative par oracle se décompose en deux phases :

La première phase (notée I sur la Figure 5.1), utilise pour le calcul de la carte de détectabilité, un ensemble de classifieurs FLD entraîné à distinguer les images *cover* des images *stégo* de l'algorithme HUGO. Une fois calculée, la carte de détectabilité est utilisée pour l'insertion du message. Cette opération est effectuée comme dans HUGO, en simulant l'insertion parfaite via l'algorithme optimale (section 2.6.2 : équations Eq. 2.14 et Eq. 2.15). À la sortie de cette première phase nous obtenons une image *stégo* ASO.

La seconde phase (notée II sur la Figure 5.1) est une phase itérative, qui vise à augmenter l'indétectabilité du message (augmentation de la sécurité). Elle utilise, pour le calcul de la carte de détectabilité, à chaque itération, un ensemble de classifieurs FLD qui a appris à faire la différence entre les images de couverture et les images stéganographiées ASO obtenues lors de l'itération précédente. Le processus itératif est répété jusqu'à obtention de la probabilité d'erreur de classification désirée. À la fin de cette seconde phase, nous obtenons une base d'images *stégo* ASO.

Pour toutes les itération, le processus de dissimulation ASO est effectué sur la même base d'images de couverture.

À la la sortie du système, le schéma ASO permet d'obtenir toute une base d'images stéganographiées au lieu d'une seule image. Ceci permet donc au stéganographe, lors de la phase de transmission, de choisir l'image *stégo* la plus sûre (ou les images *stégo* les plus sûres, selon le scénario envisagé) pour la communication de son message secret.

5.3 Calcul de la carte de détectabilité

Notre stratégie de dissimulation de données s'appuie sur le principe de minimisation d'impact d'insertion (section 2.6.2). Comme indiqué précédemment, ce principe permet de séparer le processus de création d'un algorithme stéganographique en deux étapes distinctes : le calcul de la carte de détectabilité, et l'algorithme d'insertion du message secret (les codes correcteurs utilisés). Dans ce qui suit, nous proposons une nouvelle méthode de calcul de la carte de détectabilité basée sur l'utilisation d'un oracle.

5.3.1 Aspect théorique

Le schéma de stéganographie adaptative par oracle (ASO) repose sur l'adaptativité de l'insertion à travers l'utilisation d'une carte de détectabilité $\rho = \{\rho_i \in [0, \infty[]_{i=1}^n$ calculée par un oracle. Les fonctionnalités de l'ensemble de classifieurs FLD de Kodovský (voir la section 4.4.2), ainsi que les informations acquises lors de l'apprentissage sont exploitées lors du calcul de la carte de détectabilité. L'objectif est de tirer profit des informations de la base de données utilisée par l'émetteur afin d'augmenter la sécurité du processus d'insertion.

Considérons une image de couverture en niveau de gris $\mathbf{x} = (x_1, \dots, x_n)$ composée de n pixels, un vecteur \mathbf{f}_x caractérisant cette image, une fonction de distorsion D additive telle que (Eq. 2.10), et une insertion par correspondance des LSB (*LSB Matching*, voir la section 2.5.1).

Notre objectif est de calculer la carte de détectabilité $\rho \in \mathbb{R}^n$ indépendamment en chaque pixel x_i . Pour ce faire, nous définissons en premier lieu la détectabilité ρ_i du pixel x_i de la même façon que dans HUGO [Pevný *et al.*, 2010] :

$$\rho_i = \min(\rho_i^{(+)}, \rho_i^{(-)}), \quad (5.1)$$

avec $\rho_i^{(+)}$ (respectivement $\rho_i^{(-)}$) la détectabilité après modification $+1$ (respectivement -1) du pixel x_i .

Pour construire cette carte de détectabilité, nous proposons de calculer la détectabilité $\rho_i^{(+)}$ (resp. $\rho_i^{(-)}$) grâce à un oracle formé de L classifieurs FLD [Kodovský et Fridrich, 2011]. Pour cela, nous définissons la détectabilité $\rho_i^{(+)}$ (resp. $\rho_i^{(-)}$) comme étant la somme sans pondération de la détectabilité $\rho_i^{(l)}$ de chaque classifieur F_l , avec $l \in \{1.., L\}$:

$$\rho_i^{(+)} = \sum_{l=1}^L \rho_i^{(l)(+)}, \quad \text{et} \quad \rho_i^{(-)} = \sum_{l=1}^L \rho_i^{(l)(-)}, \quad (5.2)$$

avec $\rho_i^{(l)(+)}$ (resp. $\rho_i^{(l)(-)}$) la détectabilité fournie par le $l^{\text{ème}}$ classifieur.

Pour un classifieur F_l , $l \in \{1.., L\}$, nous définissons la détectabilité $\rho_i^{(l)(+)}$, $l \in \{1.., L\}$, par :

$$\begin{aligned} \rho_i^{(l)(+)} &= \frac{\mathbf{w}^{(l)} \cdot \mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{w}^{(l)} \cdot \mathbf{f}_x^{(l)}}{s^{(l)}} \\ &= \frac{\mathbf{w}^{(l)} \cdot (\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)})}{s^{(l)}}. \end{aligned} \quad (5.3)$$

De même, la détectabilité $\rho_i^{(l)(-)}$ est définie par :

$$\rho_i^{(l)(-)} = \frac{\mathbf{w}^{(l)} \cdot (\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)})}{s^{(l)}}, \quad (5.4)$$

avec $s^{(l)} \in \mathbb{R}_+$ le facteur de normalisation (*le scaling*) du $l^{\text{ème}}$ classifieur F_l , $\mathbf{w}^{(l)}$ le vecteur orthogonal à l'hyper-plan séparateur des deux classes *cover* et *stégo* du classifieur F_l , $\mathbf{f}_x^{(l)}$

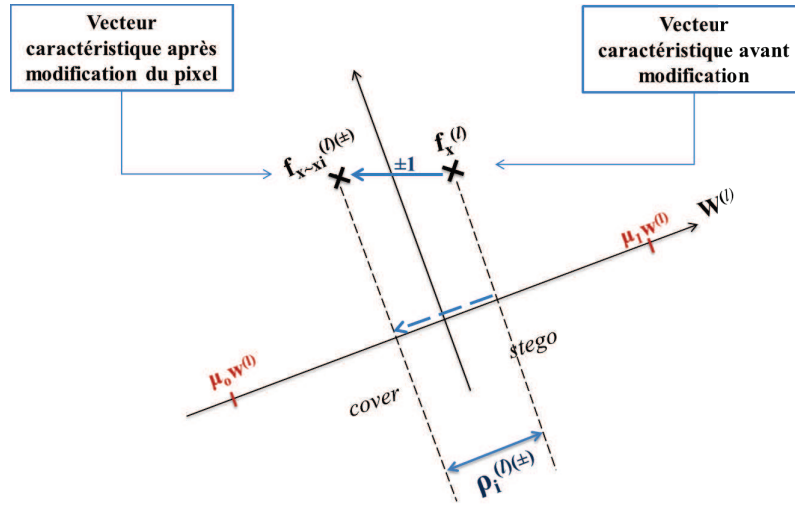


FIGURE 5.2 – Exemple illustrant le principe du calcul des coûts de détectabilité pour un classifieur FLD.

le vecteur caractéristique à classer par le classifieur F_l , et $\mathbf{f}_{x \sim x_i}^{(l)(+)}$ (resp. $\mathbf{f}_{x \sim x_i}^{(l)(-)}$) le vecteur caractéristique après modification $+1$ (resp. -1) du pixel x_i .

Notre objectif est d'obtenir une valeur $\rho_i^{(l)(+)}$ (resp. $\rho_i^{(l)(-)}$) faible, lorsque la modification $+1$ (resp. -1) entraîne un déplacement $(\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)})$ (resp. $(\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)})$) vers la classe *cover* (voir l'exemple sur la Figure 5.2). L'hypothèse forte de notre schéma ASO est que la modification d'un pixel doit avoir tendance à rapprocher l'image *stégo* d'une image *cover*. Par construction, le vecteur $\mathbf{w}^{(l)}$ est toujours orienté dans le sens *cover* vers *stégo*. Ainsi, en calculant $\rho_i^{(l)(+)}$ et $\rho_i^{(l)(-)}$ par les équations (Eq. 5.3) et (Eq. 5.4), nous obtenons exactement ce comportement, puisque les détectabilités $\rho_i^{(l)(+)}$ et $\rho_i^{(l)(-)}$ sont minimales lorsque le vecteur $\mathbf{w}^{(l)}$ et le vecteur $(\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)})$ (resp. $(\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)})$) sont colinéaires et de sens opposé, autrement dit, lorsque :

$$\mathbf{w}^{(l)} \cdot (\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)}) < 0 \quad \text{ou lorsque} \quad \mathbf{w}^{(l)} \cdot (\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)}) < 0. \quad (5.5)$$

Géométriquement (comme l'illustre la Figure 5.2), la classe des images de couverture (*cover*) et la classe des images stéganographiées (*stégo*) sont délimitées par un hyper-plan séparateur, dont le vecteur $\mathbf{w}^{(l)}$ est orthogonal. Une petite valeur de $\rho_i^{(l)(+)}$ ou $\rho_i^{(l)(-)}$ correspond :

- soit à se diriger vers la région de la classe *cover* si le vecteur caractéristique \mathbf{f}_x de l'image x est dans la région *stégo*.

- soit à s'enfoncer encore plus dans la région *cover* si le vecteur caractéristique \mathbf{f}_x de l'image x est déjà dans la région *cover*.

L'exemple sur la Figure 5.2 illustre le cas où la modification du pixel x_i de l'image x par ± 1 entraîne un déplacement $(\mathbf{f}_{x \rightarrow x_i}^{(l)(\pm)} - \mathbf{f}_x^{(l)})$ du vecteur caractéristique \mathbf{f}_x vers la distribution des images de couverture. Sur la Figure, après la modification du pixel, le vecteur \mathbf{f}_x devient $\mathbf{f}_{x \rightarrow x_i}^{(l)(\pm)}$ et se déplace donc de la classe *stégo* à la classe *cover*. Ce déplacement résulte un coût de détectabilité $\rho_i^{(l)(\pm)}$ (Eq. 5.3 et Eq. 5.4) négatif, ce qui indique que ce pixel est hypothétiquement sûr pour de la dissimulation du message secret.

Par ailleurs, on comprend aisément la nécessité d'avoir un facteur de mise à l'échelle $s^{(l)}$ propre à chaque classifieur. En effet, cela permet d'avoir une mesure de détectabilité $\rho_i^{(l)(+)}$ et $\rho_i^{(l)(-)}$ d'un même ordre de grandeur entre les classifieurs. Puisque le calcul $\rho_i^{(l)(+)}$ et $\rho_i^{(l)(-)}$ s'obtient par une somme des $\rho_i^{(l)(+)}$ et $\rho_i^{(l)(-)}$, alors chaque classifieur doit donner un coût de détectabilité $\rho_i^{(l)(+)}$ ou $\rho_i^{(l)(-)}$ avec $l \in \{1 \dots, L\}$, d'un même ordre de grandeur.

Dans un classifieur FLD, avec des distributions gaussiennes, 95% [Bajorski, 2011] des vecteurs de caractéristiques $\mathbf{f}_x^{(l)}$, après projection tombent dans l'intervalle :

$$[\mu_0^{(l)} \mathbf{w}^{(l)} - 2\sqrt{\mathbf{w}^{(l)\top} \Sigma_0^{(l)} \mathbf{w}^{(l)}}, \mu_1^{(l)} \mathbf{w}^{(l)} + 2\sqrt{\mathbf{w}^{(l)\top} \Sigma_1^{(l)} \mathbf{w}^{(l)}}], \quad (5.6)$$

avec $\mu_0^{(l)}$ (resp. $\mu_1^{(l)}$) le vecteur moyenne de la classe *cover* (resp. de la classe *stégo*), et $\Sigma_0^{(l)}$ (resp. $\Sigma_1^{(l)}$) la matrice de co-variance de la classe *cover* (resp. de la classe *stégo*).

Soit la fonction de normalisation $g^{(l)} : \mathbb{R}^{\text{dred}} \rightarrow [0, 1]$ définie par :

$$g^{(l)}(\mathbf{f}_x^{(l)}) = \frac{\mathbf{w}^{(l)} \cdot \mathbf{f}_x^{(l)}}{s^{(l)}} + b^{(l)}, \quad (5.7)$$

avec $b^{(l)} \in \mathbb{R}$ une valeur de translation facilement calculable mais inutile par la suite, et $s^{(l)}$ le facteur de normalisation définie par :

$$s^{(l)} = (\mu_1^{(l)} - \mu_0^{(l)}) \mathbf{w}^{(l)} + 2(\sqrt{\mathbf{w}^{(l)\top} \Sigma_0^{(l)} \mathbf{w}^{(l)}} + \sqrt{\mathbf{w}^{(l)\top} \Sigma_1^{(l)} \mathbf{w}^{(l)}}). \quad (5.8)$$

Si l'on applique $g^{(l)}$ sur un vecteur caractéristique $\mathbf{f}_x^{(l)}$ quelconque de la base d'apprentissage il y a environ 95% de chance d'obtenir une valeur dans l'intervalle $[0, 1]$. En pratique, notons que notre objectif n'est pas d'avoir une valeur entre $[0, 1]$, mais d'avoir un bon étalement des valeurs ainsi qu'un ordre de grandeur similaire entre les classifieurs.

Ainsi, la détectabilité $\rho_i^{(l)(+)}$ du classifieur F_l est définie par (le terme $b^{(l)}$ de l'équation disparaît) :

$$\rho_i^{(l)(+)} = \frac{\mathbf{w}^{(l)} \cdot (\mathbf{f}_{x_{\sim x_i}}^{(l)(+)} - \mathbf{f}_x^{(l)})}{(\mu_1^{(l)} - \mu_0^{(l)})\mathbf{w}^{(l)} + 2(\sqrt{\mathbf{w}^{(l)\top}\Sigma_0^{(l)}\mathbf{w}^{(l)}} + \sqrt{\mathbf{w}^{(l)\top}\Sigma_1^{(l)}\mathbf{w}^{(l)}})}. \quad (5.9)$$

De même, la détectabilité $\rho_i^{(l)(-)}$ est :

$$\rho_i^{(l)(-)} = \frac{\mathbf{w}^{(l)} \cdot (\mathbf{f}_{x_{\sim x_i}}^{(l)(-)} - \mathbf{f}_x^{(l)})}{(\mu_1^{(l)} - \mu_0^{(l)})\mathbf{w}^{(l)} + 2(\sqrt{\mathbf{w}^{(l)\top}\Sigma_0^{(l)}\mathbf{w}^{(l)}} + \sqrt{\mathbf{w}^{(l)\top}\Sigma_1^{(l)}\mathbf{w}^{(l)}})}. \quad (5.10)$$

À la fin du calcul, nous obtenons une carte de détectabilité $\rho \in \mathbb{R}$ composée de valeurs positives et négatives. Pour obtenir une carte de détectabilité positive $\rho = \{\rho_i \in [0, \infty[]_{i=1}^n$, nous translatons l'ensemble des valeurs de notre carte par la valeur $\rho_{\min} = \min(\rho)$ où ρ_{\min} est la plus petite détectabilité de la carte ρ . Une fois calculée, cette carte de détectabilité est alors utilisée pour l'insertion du message secret.

5.3.2 Aspect pratique

Le calcul d'un vecteur caractéristique $\mathbf{f}_x \in \mathbb{R}^d$, de grande dimension d , est une opération coûteuse en espace mémoire et en temps de calcul. Dans notre cas, le vecteur \mathbf{f}_x est obtenu en appliquant plusieurs filtres passe-haut et en calculant les occurrences des différents résidus (voir la section 4.4.1 sur l'extraction des caractéristiques). D'autre part, la construction de la carte de détectabilité de l'algorithme ASO passe par le calcul des valeurs $\rho_i^{(l)(+)}$ et $\rho_i^{(l)(-)}$ pour chaque pixel x_i , ce qui nécessite de calculer deux nouveaux vecteurs $\mathbf{f}_{x_{\sim x_i}}^{(l)(+)}$ et $\mathbf{f}_{x_{\sim x_i}}^{(l)(-)}$ à chaque modification d'un pixel par ± 1 et ce pour chaque classifieur F_l (voir Eq. 5.3 et Eq. 5.4). Du fait que pour un classifieur FLD le vecteur $\mathbf{w}^{(l)}$ ainsi que le facteur de normalisation $s^{(l)}$ sont calculés lors de la phase d'apprentissage, ce sont donc des valeurs pré-calculées qui ne sont pas à recalculées lors du calcul $\rho^{(l)(+)}$ et $\rho^{(l)(-)}$. Autrement dit, le problème de complexité de la carte de détectabilité ρ provient principalement du calcul des vecteurs $\mathbf{f}_{x_{\sim x_i}}^{(l)(+)}$ et $\mathbf{f}_{x_{\sim x_i}}^{(l)(-)}$. Afin de réduire la complexité, au lieu de calculer séparément $\mathbf{f}_{x_{\sim x_i}}^{(l)(+)}$ puis $\mathbf{f}_{x_{\sim x_i}}^{(l)(-)}$, nous proposons de calculer directement sur une zone réduite de l'image, la variation $(\mathbf{f}_{x_{\sim x_i}}^{(l)(+)} - \mathbf{f}_x^{(l)})$ et $(\mathbf{f}_{x_{\sim x_i}}^{(l)(-)} - \mathbf{f}_x^{(l)})$ impliquée par la modification $+1$ ou -1 sur le pixel x_i .

Pour résoudre le problème de complexité, nous définissons pour chaque pixel x_i une zone locale carrée, de taille $r \times r$, centrée autour du pixel x_i . Cette zone locale définit l'ensemble des pixels responsables du changement entre les vecteurs $\mathbf{f}_x^{(l)}$ et $\mathbf{f}_{x_{\sim x_i}}^{(l)(+)}$ (resp. entre $\mathbf{f}_{x_{\sim x_i}}^{(l)(-)}$ et $\mathbf{f}_x^{(l)}$). Ainsi, nous considérons uniquement cet ensemble de pixels lors

du calcul des variations des vecteurs caractéristiques. Le paramètre r définissant la taille de la zone carré dépend du nombre de pixels nécessaire pour le calcul de résidus s (la taille du filtre), et de l'ordre la matrice de co-occurrence m utilisé pour le calcul du vecteur caractéristique (section 4.4.1). La zone définie pour le calcul de la variation $(\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)})$ et $(\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)})$ doit être suffisamment large pour couvrir toutes les modifications possibles dues à la modification du pixel x_i . Sachant que la modification d'un pixel x_i par ± 1 peut affecter (dans les cas non pathologiques) m -uplets, $(x_{i+a}, x_{i+(a+1)}, \dots, x_{i+(a+m)})$ avec $a \in \{-\lfloor \frac{r}{2} \rfloor, \dots, \lfloor \frac{r}{2} \rfloor - m\}$, et ce pour toutes les directions, nous avons choisi de mettre $r = s + 2(m - 1)$ pour couvrir toutes les variations possibles des vecteurs $(\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)})$ et $(\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)})$.

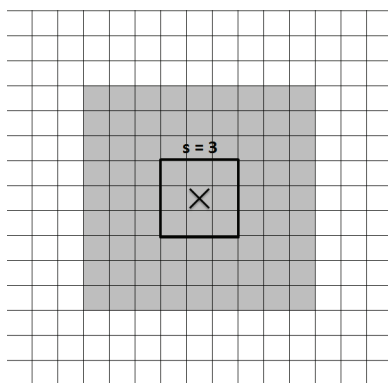


FIGURE 5.3 – Exemple illustrant le principe de calcul des variations des vecteurs caractéristiques sur une zone réduite de taille $r = 9$. Sur l'exemple le filtre 1-D résiduel utilisé pour l'extraction des caractéristiques est de taille $s = 3$.

De façon plus simpliste, prenons l'exemple illustré sur la Figure 5.3. Pour un filtre résiduel 1-D de taille $s = 3$ et $m = 4$, les variations $(\mathbf{f}_{x \sim x_i}^{(l)(+)} - \mathbf{f}_x^{(l)})$ et $(\mathbf{f}_{x \sim x_i}^{(l)(-)} - \mathbf{f}_x^{(l)})$, obtenues après modification, sont calculées à partir d'une zone réduite de taille $r = 9$.

5.4 Insertion du message secret

La dernière étape, après le calcul de la carte de détectabilité, est l'insertion du message secret dans l'image de couverture qui lui est associée. Lors de cette étape, le stéganographe fait appel aux codes correcteurs d'erreurs, pour insérer son message de manière efficace. Autrement dit, insérer le message secret, tout en apportant le minimum de modifications possibles (en terme de nombre) sur le support hôte (voir la partie sur l'efficacité d'insertion en section 2.4.2). Dans la littérature, il existe plusieurs propositions de codes correcteurs d'erreurs qui sont adaptés à la stéganographie. Pour les méthodes adaptatives par mini-

misation d'impact d'insertion utilisant une carte de détectabilité (tel que ASO), l'approche treillis (STC : *Syndrome Trellis Codes*) est par construction le choix le plus adapté (voir chapitre 3.3.3). C'est d'ailleurs, l'approche la plus performante actuellement en terme d'efficacité d'insertion [Filler *et al.*, 2011]. Le principe de cette méthode de codage par syndrome consiste à trouver le vecteur \mathbf{y} qui représente l'image *stégo* pour la résolution de l'équation $\mathbf{H}\mathbf{y} = \mathbf{m}$ (\mathbf{H} étant la matrice de parité et \mathbf{m} le message à transmettre), tout en essayant de minimiser l'impact d'insertion dû à la modification de l'image de couverture. Pour ce faire, le codeur STC prend en entrée la carte de détectabilité précédemment calculée, puis lance l'algorithme de Viterbi sur le treillis constitué d'états et de branches pondérées par les poids de la carte. L'objectif est de trouver le chemin du plus faible coût, c'est-à-dire celui qui correspond à la solution recherchée. Une fois que les endroits de modifications sont identifiés, l'insertion du message secret peut être concrètement réalisée.

Dans notre cas, l'insertion du message secret est effectuée en simulant l'insertion parfaite via l'algorithme optimal, c'est-à-dire en cherchant le paramètre λ de l'Eq. 2.15, puis en modifiant chaque pixel x_i selon la probabilité p_i définie dans l'Eq. 2.14 (voir la section 2.6.2).

5.5 Détails techniques de l'implémentation de ASO

Pour l'implémentation de notre algorithme de dissimulation ASO, nous avons utilisé notre propre implémentation C++ de l'ensemble de classifieurs FLD (section 4.4.2). Nous avons fixé le nombre de classifieurs FLD à $L = 30$, et la dimension des sous-ensembles utilisés par les classifieurs à $d_{\text{red}} = 250$. Pour le processus d'apprentissage de l'oracle ASO nous avons utilisé la base BossBase v1.00 database³ composée de 10 000 images en niveau de gris de taille 512×512 au format pgm ; 5000 de couverture et 5000 images stéganographiées par l'algorithme HUGO [Pevný *et al.*, 2010] sont utilisées lors de l'apprentissage. Le *payload* est fixé par l'utilisateur. Pour maintenir un certain équilibre entre l'optimalité et la performance, nous avons choisi de représenter chaque image par un vecteur caractéristique MINMAX [Fridrich *et al.*, 2011a] de taille $d = 5330$. Les paramètres des caractéristiques MINMAX utilisées, ainsi que leur dimension sont présentés dans le tableau 5.1.

Pour l'implémentation de ASO nous avons fait appel au centre de calcul HPC@LR⁴ de Montpellier 2, qui est un centre dédié au calcul de haute performance. Le schéma de sté-

3. BOSSBase v1.00 : base d'images disponible sur le site <http://agents.cz/boss/BOSSFinal/>

4. HPC@LR : Centre de Compétences en calcul haute performance de la région Languedoc Roussillon - Université de Montpellier 2. HPC@LR est une plateforme matérielle hybride de 15-Teraflop (double précision), composée de 80 nœuds de calcul IBM dx360 M3 disposant chacun de deux processeurs SIX CORE INTEL, 2 nœuds à Large mémoire (SMP) de $\times 80$ coeurs Intel(R) Xeon(R) CPU E7- 8860, 6 CPU/GPU de 2 processeurs QUAD CORE INTEL WESTMERE + 2 cartes NVIDIA TESLA M2050, 4 lames CELL de deux processeurs PowerXCell 8i à 4GHz, une double lame PS702 configurée avec 16 cœurs Power7, un Réseau INFINIBAND QDR IBM 12800-180, et une mémoire de stockage externe IBM DCS9900 avec 150 disques SATA

s	q	m	T	d
3	2	3	3	686
3	2	4	2	1250
3	2	3	4	1458
4	2	3	3	686
4	2	4	2	1250

TABLE 5.1 – Paramètres des caractéristiques MINMAX [Fridrich *et al.*, 2011a] utilisées. s : le nombre des pixels utilisés pour le calcul du résidu, q : le pas de quantification, m : l'ordre des matrices de co-occurrence, T : le seuil de la troncature, et d : la dimension résultante (voir la section 4.4.1).

ganographie ASO a été parallélisé sur un serveur d'architecture SMP composé de 80 coeurs Intel(R) Xeon(R) CPU E7- 8860 cadencés à 2.27GH. Pour $N = 10000$ images, $d = 5330$ caractéristiques, $d_{red} = 250$ et $L = 30$ classifieurs, l'exécution de l'algorithme ASO avec seulement une itération (étape I sur la Figure 5.1) prend moins de 14 heures. Comme mentionné précédemment, l'étape la plus couteuse en temps de calcul, pour l'exécution du schéma ASO, est celle du calcul des vecteurs caractéristiques lors de la construction de la carte de détectabilité.

5.6 Tests et résultats

Dans cette section, nous nous intéressons aux différents résultats que nous avons obtenu lors de nos tests sur l'algorithme proposé ASO. Du point vue stéganalyse, nous nous posons dans le cadre de la stéganalyse à clairvoyance (section 4.2.1). Autrement dit, dans ce qui suit nous considérons que la gardienne Eve connaît : le processus de dissimulation ASO, le processus de dissimulation HUGO, la distribution originale des images de couverture utilisées pour l'insertion, et la quantité de bits (*le payload*) insérés. Eve est donc capable de régénérer de son coté des images stéganographiées avec l'algorithme ASO pour apprendre à distinguer entre les images de couverture et les images stéganographiées contenant un message secret. Nous rappelons également que le scénario de stéganographie étudié dans cette section se limite uniquement à l'envoi d'une seule image lors de la communication du message secret. En d'autres termes nous nous intéressant pas à la stéganographie par lot (*Batch steganography*).

5.6.1 Évaluation de la sécurité de ASO

Pour évaluer l'efficacité de notre schéma de stéganographie ASO, nous avons d'abord testé l'efficacité du processus de dissimulation ASO avec uniquement une seule itération.

de lTo. Pour plus de détails sur l'architecture du centre HPC@LR, le lecteur est prié de se référer au site <https://www.hpc-lr.univ-montp2.fr/>.

Autrement dit, nous avons évalué uniquement la première phase de ASO (notée I sur la Figure 5.1), L'oracle utilisé est donc un oracle qui a appris à faire la distinction entre les images de couverture et les images stéganographiées avec l'algorithme HUGO.

Pour rappel (voir la section 5.5) :

- L'implémentation de l'oracle ASO est réalisée en utilisant le vecteur caractéristique MINMAX de dimension $d = 5330$, avec notre propre version C++ de l'ensemble de classifieurs FLD. Concernant les paramètres de l'ensemble de classifieurs, nous avons fixé le nombre de classifieurs FLD à $L = 30$ et la dimension des sous-ensembles de caractéristiques utilisés par les classifieurs à $d_{red} = 250$ (chaque classifieur FLD utilise une graine, différente de celle des autres classifieurs, pour choisir de façon aléatoire son sous-ensemble de caractéristiques) ;
- La phase d'apprentissage de l'oracle ASO est réalisée sur 5000 images *cover* (de la base BossBase-v1) et 5000 images *stégo* HUGO qui leur sont associées.
- L'algorithme de dissimulation ASO prend en entrée 10000 images de couverture (BossBase-v1), et génère en sortie 10000 images stéganographiées.

Protocole expérimental de la stéganalyse

Afin de tester la performance de ASO (avec une seule itération) en situation réelle, nous avons comparé la sécurité de l'algorithme ASO avec celle de l'algorithme HUGO, et ce pour cinq *payloads* différents : de 0.1 *bpp* à 0.5 *bpp*. Pour ce faire :

- Nos expérimentations ont été réalisées pour deux types différents de caractéristiques : Le vecteur caractéristique *Spatial domain Rich Model* SRMQ1 de dimension $d = 12753$, et le vecteur caractéristique complet *Spatial domain Rich Model* SRM de dimension $d = 34671$, proposés par [Fridrich et Kodovský, 2012]. Le vecteur caractéristique SRMQ1 utilise un pas de quantification fixe lors du calcul des résidus, alors que le vecteur caractéristique SRM utilise plusieurs pas de quantification. Les deux vecteurs SRMQ1, et SRM sont la fusion de différents sous-modèles (caractéristiques) SPAM, MINMAX, SQUARE, et EDGE (voir la section 4.4.1) ;
- Pour la stéganalyse nous avons utilisé la version originale de l'ensemble de classifieurs FLD de Kodovský⁵, celle avec recherche automatique des paramètres d_{red} et L [Kodovský *et al.*, 2012] (Autrement dit, $L \neq 30$, $d_{red} \neq 250$, et des sous-espaces aléatoires différents générés par différentes graines) ;
- Lors de la stéganalyse avec le détecteur de Kodovský, nous avons utilisé la base BossBase-v1.00 composée de 10000 images *cover* et de 10000 *stégo* qui leur sont associées.

5. La source MATLAB originale de l'ensemble de classifieurs FLD de Kodovský est disponible sur le site : <http://dde.binghamton.edu/download/ensemble/>.

- Pour chaque *payload*, la base d'images pour la stéganalyse a été coupée de façon aléatoire en deux parties égales : une partie pour la phase d'apprentissage, et une autre pour la phase test. L'opération de division de la base est répétée cinq fois, avec une graine différente à chaque fois. La performance présentée au final, pour chaque *payload*, est la moyenne de la stéganalyse des cinq divisions.

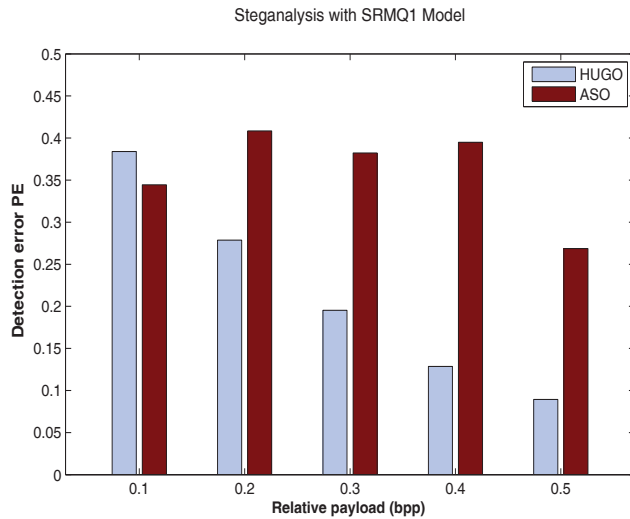
Pour résumer, l'oracle ASO et le détecteur (l'ensemble de classifieurs FLD de Kodovský) utilisé pour la stéganalyse sont totalement différents. Les caractéristiques ainsi que les sous-espaces des classifieurs sont également différents.

Analyse des résultats

Les résultats de la stéganalyse de l'algorithme ASO et de l'algorithme HUGO sont illustrés sur les Figures 5.4 et 5.5. La Figure 5.4 présente les résultats obtenus pour le vecteur caractéristique SRMQ1, et la Figure 5.5 présente les résultats de la stéganalyse pour le vecteur caractéristique SRM. Comme illustré sur ces deux Figures, pour les deux expérimentations ASO et HUGO conservent à peu près le même comportement. Comme nous pouvons le constater, la performance de l'algorithme ASO en terme de sécurité est supérieure à celle de HUGO pour les *payloads* de 0.2 *bpp* à 0.5 *bpp*, et ce pour les deux vecteurs SRMQ1 et SRM. À titre d'exemple, pour un *payload* de 0.5 *bpp* la probabilité d'erreur de détection, P_E (voir l'Eq. 4.22), de l'algorithme ASO en utilisant le vecteur caractéristique SRMQ1 est de 26.87%, contre seulement 8.94% pour l'algorithme HUGO. De même, en étendant la taille du vecteur caractéristique, on obtient également le même comportement. À 0.5 *bpp*, avec le vecteur SRM, la probabilité d'erreur de détection, P_E , de ASO est supérieure à celle de HUGO. Elle est de 26.74% contre 8.35% pour HUGO. Nous pouvons donc dire que pour les *payloads* élevées, la sécurité de l'algorithme ASO est meilleure que celle de l'algorithme HUGO.

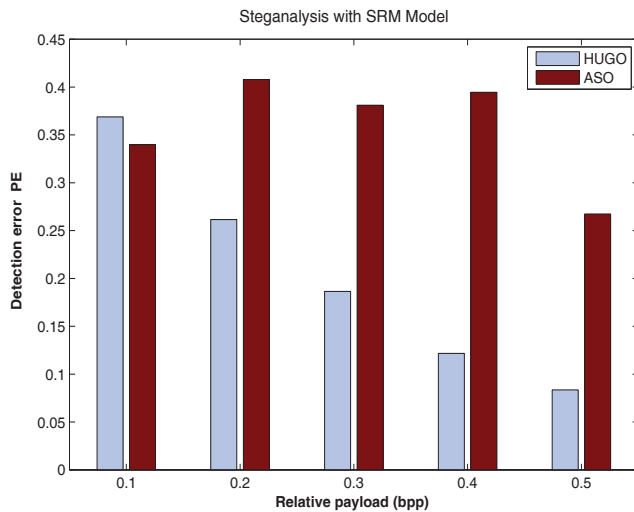
Nous constatons également que le niveau de sécurité de ASO, pour les petits taux d'insertion, est plus faible que celui de HUGO. Pour un *payload* de 0.1 *bpp*, en utilisant le vecteur caractéristique SRM, la probabilité d'erreur de détection, P_E , de ASO est de 33.99% contre 36.89% pour HUGO. Respectivement, pour le même *payload* avec comme vecteur caractéristique le SRMQ1, l'erreur de détection pour ASO est de 34.45% comparée à 38.40% pour HUGO. Cela peut s'expliquer par le fait que pour un tel *payload*, l'oracle utilisé pour le calcul de la carte de détectabilité (section 5.3) a du mal à distinguer les images de couverture des images stéganographiées HUGO, ce qui influence automatiquement l'insertion avec ASO. Autrement dit, l'algorithme ASO n'arrive pas à distinguer les régions sûres de celles qui ne le sont pas.

Par ailleurs, on note que pour les deux modèles SRMQ1 et SRM, de 0.1 *bpp* à 0.5 *bpp*, l'algorithme ASO a plutôt un comportement non-monotone en comparaison de celui de HUGO. Par exemple, à 0.4 *bpp* pour le vecteur caractéristique SRM, la probabilité d'erreur de détection P_E de ASO est supérieure à celle à 0.3 *bpp*. Elle est de 39.45% pour 0.4 *bpp*,



Algorithm	Payload (bpp)				
	0.1	0.2	0.3	0.4	0.5
HUGO	0.3840	0.2786	0.1953	0.1285	0.0894
ASO	0.3445	0.4084	0.3822	0.3949	0.2687

FIGURE 5.4 – Niveau de sécurité de ASO vs HUGO, pour le vecteur caractéristique SRMQ1 de dimension 12753. La performance finale, présentée pour chaque *payload*, est la moyenne obtenue de la stéganalyse de cinq bases différentes (voir le protocole expérimental en section 5.6.1).



Algorithm	Payload (bpp)				
	0.1	0.2	0.3	0.4	0.5
HUGO	0.3689	0.2616	0.1865	0.1217	0.0835
ASO	0.3399	0.4080	0.3811	0.3945	0.2674

FIGURE 5.5 – Niveau de sécurité de ASO vs HUGO, pour le vecteur caractéristique complet SRM de dimension 34671. La performance finale, présentée pour chaque *payload*, est la moyenne obtenue de la stéganalyse de cinq bases différentes (voir le protocole expérimental en section 5.6.1).

contre 38.11% pour le *payload* de 0.3 *bpp*, ce qui est contre intuitif et inattendu. La probabilité d'erreur P_E devrait plutôt décroître de façon monotone, et non accroître. La première explication qui peut expliquer ce comportement non-monotone de ASO est sans doute le fait que nous avons fixé les paramètres de l'oracle ASO à $d_{red} = 250$ et $L = 30$ lors de notre propre implémentation C++. Ces paramètres devraient probablement être différents pour chaque *payload*, ce qui indique que notre approche pourrait être encore améliorée par la recherche automatique des paramètres optimaux d_{red} et L pour chaque taux d'insertion (*payload*). La deuxième explication qui pourrait également expliquer ce comportement est que nous avons deux effets contradictoires pour ASO. D'une part, plus le taux d'insertion est faible plus le processus de dissimulation est difficilement détectable (à cause du nombre peu élevé de modifications apportées sur le support hôte), et d'autre part, moins le taux d'insertion est important, moins est la fiabilité et la précision de l'oracle ASO pour le calcul de la carte de détectabilité (ce qui rend le processus de dissimulation ASO plus détectable).

De manière globale, les résultats obtenus démontrent que la sécurité de notre schéma ASO est meilleure que celle de l'algorithme HUGO. Face aux deux modèles de stéganalyse SRMQ1 (de dimension 12753) et SRM (de dimension 34671), l'algorithme ASO avec uniquement une seule itération présente de bonnes performances en terme de sécurité. Cela confirme que les images stéganographiées par ASO sont construites de manière à garder leur distribution le plus proche possible de la classe des images de couverture. Autrement dit, le processus de dissimulation ASO est construit de telle façon à garder la frontière qui sépare les deux classes *cover/stégo* très fine, ce qui rend difficile la détection du message caché. Il est vrai que la complétude [Kodovský *et al.*, 2008] de l'algorithme ASO ne peut être garantie face à toutes les attaques possibles (avec d'autres vecteurs caractéristiques). Toutefois, nous tenons à souligner, que l'approche que nous présentons représente une nouvelle philosophie de dissimulation générale, qui peut être appliquée à n'importe quel algorithme de stéganographie (autre que HUGO), et qui peut encore être améliorée en utilisant un modèle de couverture plus complet.

5.6.2 Évaluation du processus itératif de ASO

Comme expliqué précédemment en section 5.2, le schéma de dissimulation proposé ASO est constitué de deux phases successives :

La première phase (notée I sur la Figure 5.1) utilise pour la dissimulation de données un oracle qui a appris à distinguer entre les images de couverture et les images stéganographiées avec HUGO. À la fin de cette phase, le processus de dissimulation ASO effectue une première itération et fournit en sortie les premières images stéganographiées avec ASO.

La seconde phase de ASO (notée II sur la 5.1) est une phase itérative qui utilise pour chaque itération un oracle qui a appris à distinguer les images de couverture des images

stéganographiées avec ASO de l'itération précédente. Les mêmes images de couvertures sont utilisées à chaque itération, et le processus itératif est répété jusqu'à obtention de la performance souhaitée. L'objectif de cette seconde phase est d'améliorer la sécurité du schéma ASO.

Afin d'évaluer la nécessité du processus itératif de ASO (phase II sur la Figure 5.1), et pour trouver le nombre d'itérations nécessaires pour l'amélioration de la sécurité, nous avons testé l'algorithme ASO avec 2 et 3 itérations. Pour chaque itération, nous avons testé la sécurité du schéma ASO pour cinq *payloads* différents : de 0.1 *bpp* à 0.5 *bpp*. Pour chaque *payload*, la base d'images BossBase-v1.00 est coupée de manière aléatoire en deux parties égales apprentissage/test. La performance de ASO (P_E) présentée au final, pour chacun des *payloads*, est la moyenne de la stéganalyse des cinq bases générées par la division. L'ensemble de ces tests ont été réalisés avec comme vecteur caractéristique le *Spatial Rich Model* (SRM) de dimension 34671 [Fridrich et Kodovský, 2012], et en utilisant la version originale de l'ensemble de classifieurs FLD de [Kodovský et Fridrich, 2011] (celle faisant appel à la recherche automatique des paramètres).

ASO		Payload (bpp)				
		0.1	0.2	0.3	0.4	0.5
P_E	ASO-1 1 iteration (Phase I)	0.3399	0.4080	0.3811	0.3945	0.2674
	ASO-2 2 iterations (Phase II)	0.4522	0.4478	0.4683	0.4665	0.4663
	ASO-3 3 iterations (Phase II)	0.4536	0.3925	0.3540	0.4215	0.3397

TABLE 5.2 – Comparaison des performances des 3 itérations de l'algorithme ASO, face au vecteur caractéristique SRM de dimension 34671. La stéganalyse de ASO pour chaque itération est effectuée sur cinq *payloads* différents. La performance présentée est la moyenne sur cinq bases différentes.

Le Tableau 5.2 présente les résultats de la stéganalyse de ASO, pour les trois itérations. Sur le tableau, les trois algorithmes notés ASO-1, ASO-2 et ASO-3 dénotent respectivement le schéma ASO avec uniquement une itération, deux itérations, puis trois itérations. Comme nous pouvons le constater, la meilleure performance en terme de sécurité est obtenue avec ASO-2. Avec deux itérations, la probabilité d'erreur de détection (P_E) de tous les *payloads* est au alentour de 45-46%, ce qui indique qu'à ce stade du processus de dissimulation, le détecteur est presque incapable de distinguer entre les images de couvertures et les images stéganographiées avec ASO. La seconde itération augmente donc considéra-

blement la performance de l'algorithme ASO. Par exemple, pour le *payload* de 0.5 *bpp*, la probabilité d'erreur de détection P_E de ASO-1 est de 26.74%, contre 46.63% pour ASO-2, soit une augmentation de 19.89%. De même pour les autres *payloads*, la sécurité de ASO avec deux itérations est meilleure que celle avec une seule itération.

Pour l'algorithme ASO-3 (ASO avec 3 itérations), de 0.2 *bpp* à 0.5 *bpp*, nous constatons que la probabilité d'erreur de détection P_E commence à décroître. Lors de cette troisième itération, l'oracle utilisé pour le calcul de la carte de détectabilité apprend à distinguer entre les images de couverture et les images stéganographiées avec ASO-2; la diminution de la sécurité de ASO indique clairement que cet oracle a beaucoup de difficulté à séparer les deux classes. La fiabilité de la carte de détectabilité générée par l'oracle est donc remise en cause, ce qui a pour effet d'affecter considérablement la sécurité du processus de dissimulation de ASO lors de cette itération.

En résumé, les résultats illustrés sur le tableau 5.2 confirment que la seconde phase itérative (phase II sur la Figure 5.1) améliore considérablement la sécurité de l'algorithme stéganographique ASO. À travers nos expérimentations, nous avons montré qu'il ne fallait pas plus que deux itérations pour augmenter la sécurité de ASO.

5.6.3 Analyse de la carte de détectabilité ASO

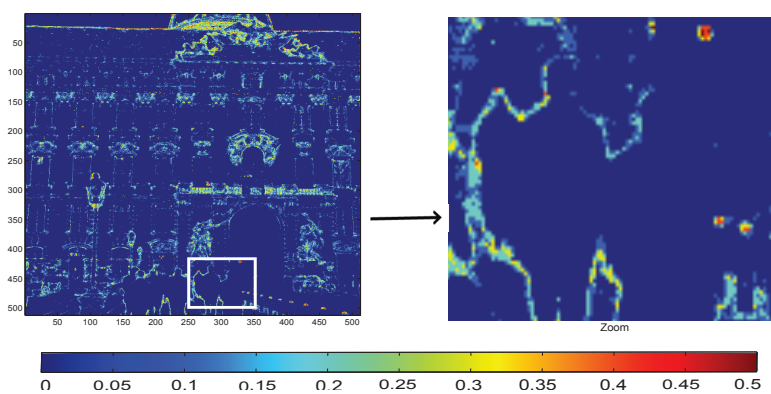
À ce stade, une des questions qui peut être posée est "Où est-ce que ASO cache le message secret". Pour répondre à cette question, nous avons étudié la probabilité d'insertion de ASO-1 (ASO avec uniquement une seule itération) et la probabilité d'insertion de l'algorithme HUGO, à différents *payloads* (de 0.1 *bpp* à 0.5 *bpp*), et ce pour 100 images de la base BossBase-v1.00. Étonnamment, pour toutes les images et tous les *payloads* étudiés, nous avons constaté que l'algorithme ASO-1 et l'algorithme HUGO ont deux stratégies d'insertion totalement différentes. Contrairement à HUGO, qui insère avec une grande probabilité dans les régions texturées et les bordures, l'algorithme stéganographique ASO-1 semble à première vue avoir un comportement totalement imprévisible lors de l'insertion.

La Figure 5.6 représente l'image des probabilités d'insertion de l'algorithme HUGO et de l'algorithme ASO-1, pour l'image n° 13 de la base BossBase à 0.2 *bpp*. Pour le calcul des probabilités d'insertion, nous avons simulé l'insertion de 100 messages différents (voir les équations Eq. 2.14 et Eq. 2.15, en section 2.6.2). Comme l'illustre la figure, les pixels rouges correspondent aux régions qui ont une probabilité d'insertion élevée, tandis que les pixels en bleu marine représentent les régions avec une probabilité d'insertion faible. Autrement dit, la probabilité d'insertion dans les pixels en rouge est beaucoup plus élevée que celle dans les pixels en bleu.

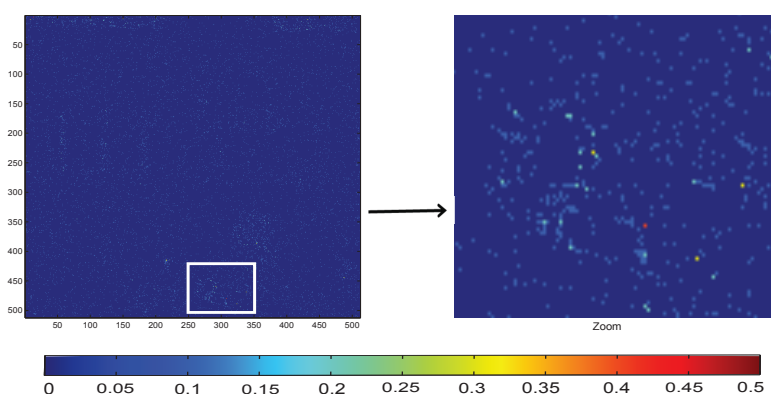
Afin de comprendre ce qui se passe réellement avec ASO-1, nous avons comparé la distribution de la carte de détectabilité de HUGO avec celle de l'algorithme ASO. Pour les



(a) Image originale en niveau de gris



(b) HUGO à 0.2 bpp



(c) ASO à 0.2 bpp

FIGURE 5.6 – La probabilité d'insertion à 0.2 *bpp* de (b) l'algorithme HUGO et (c) de l'algorithme ASO-1, pour (a) l'image n° 13 de la base BossBase-v1 (image en niveau de gris de taille 512 × 512).

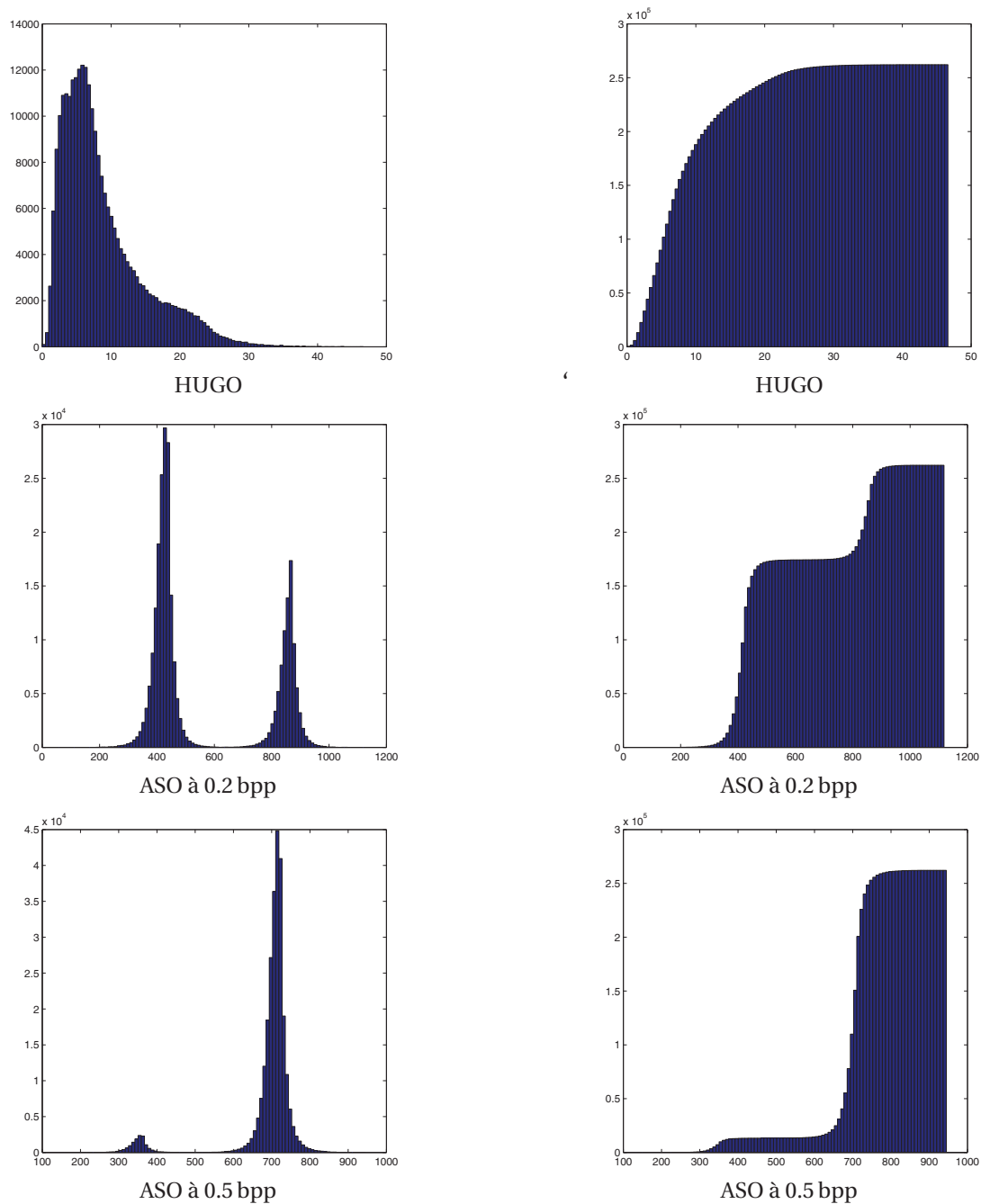


FIGURE 5.7 – Analyse des cartes de détectabilité de HUGO et ASO-1 (ASO avec une seule itération) pour l'image en niveau de gris n° 13 de la base BossBase-v1 (image (a) sur la Figure 5.6).

À Gauche : l'histogramme de la carte de détectabilité des algorithmes HUGO et ASO-1.

À Droite : les histogrammes cumulés correspondants.

mêmes 100 images testées précédemment, avec les cinq différents *payloads* (de 0.1 *bpp* à 0.5 *bpp*), nous avons constaté que l'algorithme ASO-1 n'est pas uniformément réparti, ce qui indique que le comportement de ASO-1 n'est pas aléatoire contrairement à ce que nous pouvions penser. Par ailleurs, nous avons noté également, qu'à la différence de l'algorithme HUGO qui utilise une seule carte de détectabilité identique pour tous les *payloads*, le schéma stéganographique ASO calcule une nouvelle carte de détectabilité pour chaque *payload*. Autrement dit, le schéma de stéganographie ASO utilise pour l'insertion des données secrètes une carte de détectabilité différente d'un *payload* à un autre.

La Figure 5.7 présente les histogrammes des cartes de détectabilité de l'algorithme ASO-1 (à 0.2 *bpp* et 0.5 *bpp*) et de l'algorithme HUGO, pour l'image n° 13 de la base BossBase-v1.00. Les histogrammes cumulés correspondants à ces algorithmes sont également présentés sur cette figure. Comme illustré, la distribution des cartes de détectabilité de l'algorithme ASO et celle de l'algorithme HUGO sont différentes. La carte de détectabilité de l'algorithme ASO, et la carte de détectabilité de l'algorithme HUGO, possèdent deux distributions très distinctes, ce qui explique leur stratégies différentes pour la sélection des pixels à modifier. La distribution de la carte de détectabilité ASO est bien particulière ; elle est différente d'un *payload* à un autre, mais tout en gardant le même comportement. La distribution de la carte de détectabilité ASO est caractérisée, à chaque fois, par deux modes, de forme gaussienne, bien distincts.

En résumé, nos analyses confirment que le calcul de la carte de détectabilité est une étape cruciale pour l'insertion du message secret. Lors de l'insertion des données, la carte de détectabilité permet de faire la distinction entre les zones sûres à modifier (les pixels avec un faible coût de détectabilité, qui correspondent aux pixels en rouge sur la Figure 5.6), et les zones non-sûres à éviter (les pixels avec un coût de détectabilité élevé, qui correspondent aux pixels en bleu sur la Figure 5.6). En outre, les résultats de notre analyse indiquent que le comportement du schéma ASO, lors de la sélection des pixels à modifier, n'est pas aléatoire. À travers son comportement, l'algorithme proposé ASO tente non seulement de préserver le modèle de distribution de l'image de couverture, mais également celui de la base d'images utilisée par l'émetteur, et ce sans modifier nécessairement des zones spécifiques (telles que les régions texturées et les bordures).

5.7 Conclusion

Au cours de ce chapitre, nous avons proposé une nouvelle méthode de stéganographie adaptative, dédiée aux images numériques naturelles dans le domaine spatial. La méthode de Stéganographie Adaptatif par Oracle (ASO) proposée est une nouvelle méthodologie de dissimulation de données, qui s'appuie sur l'utilisation d'un oracle pour le calcul de la carte de détectabilité. La méthode ASO exploite les informations de l'ensemble de classificateurs FLD de [Kodovský et Fridrich, 2011] pour partager l'espace des caractéristiques en deux régions distinctes (*cover* et *stégo*), et utilise ensuite cette séparation comme oracle pour le calcul des coûts de détectabilité. La carte de détectabilité établie est construite de telle façon à favoriser la modification des pixels, qui ont tendance à rapprocher la distribution de l'image *stégo* vers la distribution des images *cover*. Grâce à cette méthodologie d'insertion, le schéma ASO permet non seulement de préserver le modèle de distribution de l'image de couverture courante mais également le modèle de distribution de la base d'images utilisée par l'émetteur.

L'approche ASO proposée est une approche itérative, qui a pour but d'augmenter la sécurité du processus d'insertion du message à cacher. Les études expérimentales que nous avons mises en place ont montré que notre schéma de dissimulation ASO présente de bonnes performances en termes de sécurité. Avec uniquement une seule itération, l'algorithme ASO offre à l'émetteur la possibilité d'envoyer de longs messages secrets avec une plus grande sécurité que celle de l'algorithme HUGO. L'amélioration qu'apporte notre approche, par rapport aux différentes techniques actuelles, est particulièrement évidente lors de la deuxième itération. Le processus itératif permet d'augmenter considérablement la performance et la sécurité de ASO.

Pour conclure, l'approche ASO est nouvelle philosophie de dissimulation. Elle peut être considérée comme une méta-méthode générique, qui peut être adaptée à n'importe quel autre algorithme de stéganographie (autre que HUGO), et à n'importe quel vecteur caractéristique. L'utilisation de vecteur de caractéristiques, mieux choisi et plus complet, aurait probablement pour effet d'augmenter la sécurité et la performance de ASO. Par ailleurs, l'approche ASO introduit un nouveau concept de stéganographie ; il s'agit du concept de stéganographie par base d'images. Ce nouveau concept, qui offre à l'émetteur la possibilité de choisir l'image stéganographiée la plus sûre (ou les images les plus sûres) lors de la transmission, sera l'objet d'une étude et d'analyse détaillée dans le chapitre suivant (chapitre 6).

Les travaux présentés au cours de ce chapitre ont donné lieu à deux publications dont une nationale [Kouider *et al.*, 2012a] et une internationale [Kouider *et al.*, 2013]. Un article revu, portant sur ces travaux, est également en cours de préparation pour une très prochaine soumission.

Le paradigme de stéganographie par base

Un problème sans solution est un problème mal posé.

ALBERT EINSTEIN - *Physicien*

Préambule

Le schéma de dissimulation ASO introduit un nouveau paradigme en stéganographie : la stéganographie par base. Cette nouvelle philosophie de dissimulation permet à l'émetteur de choisir l'image la plus sûre pour la transmission de son message secret, ce qui ajoute un niveau de sécurité supplémentaire à la communication. Dans ce chapitre, après avoir défini et présenté ce nouveau paradigme (section 6.1), nous nous intéressons plus particulièrement à la sélection d'images stéganographiées pour la transmission des données. Nous discutons, dans un premier temps, du rôle des mesures de sélection en stéganographie, et présentons un état de l'art des principales méthodes de sélection existantes dans la littérature (section 6.2). Ensuite, nous proposons une nouvelle mesure de sécurité, pour la sélection des images stéganographiées, basée sur l'exploitation de l'oracle ASO (section 6.3). Enfin, nous discutons et analysons les résultats obtenus avec cette nouvelle mesure de sélection (section 6.5).

Sommaire

6.1	Le paradigme de la stéganographie par base	98
6.2	Mesures de sélection en stéganographie	98
6.3	Mesure de sécurité basée oracle pour la sélection	101
6.4	Tests et résultats	102
6.5	Discussion et conclusion	106

6.1 Le paradigme de la stéganographie par base

À travers le schéma ASO, présenté au chapitre 5, nous introduisons un nouveau concept de dissimulation de données qui est *le paradigme de stéganographie par base*. Contrairement aux méthodes classiques de la littérature, le schéma ASO a une conception bien particulière. Pour la dissimulation du message secret, le schéma ASO nécessite l'usage d'une base complète d'images de couverture au lieu d'une seule image. À l'entrée du système, le stéganographe doit fournir cette base d'images à l'oracle ASO, pour que cet oracle puisse exploiter la distribution des images lors du calcul de la carte de détectabilité. L'objectif est d'améliorer la sécurité du processus d'insertion, ceci en préservant "au mieux" la distribution de l'image courante et celle de l'ensemble des images de la base. À la sortie du système ASO (voir la Figure 5.1), le stéganographe obtient un ensemble d'images stéganographiées au lieu d'une seule image *stégo*.

Grâce à cette fonctionnalité supplémentaire qu'offre ASO, le stéganographe, lors de la transmission du message secret, peut donc selon le scénario envisagé :

- soit scinder son message secret sur plusieurs images (scénario de stéganographie par lot (*Batch steganography*) [Ker, 2006]),
- soit insérer le même message dans plusieurs images à la fois, pour choisir au final uniquement **l'image stégo la plus sûre** pour la transmission (scénario du *one-time database* [Simmons, 1983]).

La sélection d'images sûres, pour la transmission des données secrètes, étant une étape intéressante pour l'amélioration de la sécurité de ASO, nous avons donc voulu explorer cette piste.

Dans ce qui suit, nous présentons d'abord différentes méthodes de sélections existantes dans la littérature, et discutons leur apport en stéganographie (section 6.2). Ensuite, nous proposons une nouvelle mesure pour la sélection des images stéganographiées basée sur l'utilisation des fonctionnalités de l'ensemble de classifieurs FLD de [Kodovský *et al.*, 2012] (section 6.3). Enfin, nous étudions et analysons l'efficacité de cette mesure en situation réelle (section 6.4).

6.2 Mesures de sélection en stéganographie

Comme mentionné précédemment, grâce aux fonctionnalités supplémentaires qu'offre le paradigme de stéganographie par base, le stéganographe a donc le choix entre deux cas de figure, lors de la transmission de son message secret :

Le premier cas de figure consiste à choisir un ensemble d'images de couverture, puis à diviser le message secret en plusieurs parties, pour cacher chaque partie du message dans une image de couverture différente. Pour ce scénario de stéganographie par lot [Ker, 2006],

la sélection d'images les plus sûres pour la communication des différentes parties du message est délicate. En effet, le critère de sécurité doit être à la fois un critère obtenu pour une image mais aussi pour un lot d'images. Par exemple, si on considère les images texturées comme étant celles qui sont les plus sûres, alors dans ce cas-ci, il n'est pas évident qu'il soit fiable d'utiliser uniquement des images texturées pour la transmission. En effet, si Eve (la gardienne) constate le passage de plusieurs images de nature très texturée ou très bruitée différentes de ce qu'elle voit d'habitude, elle pourrait se douter de quelque chose, et peut même stopper la communication.

Dans le deuxième cas de figure, le stéganographe a la possibilité d'insérer un même message secret dans plusieurs images de couverture différentes, pour en choisir au finale uniquement l'image la plus fiable pour la transmission. Pour ce scénario du *One-time database*, et contrairement au scénario précédent, la sélection d'image de couverture sûre pour la communication est une action très intéressante pour l'amélioration des performances en terme de sécurité. En effet, dans le cas où Alice (le stéganographe) choisie une image texturée pour embarquer son message secret, Eve la gardienne du canal, qui a l'habitude de voir passer de temps à autre des images texturées, ne pourra détecter qu'il s'agit là d'une image stéganographiée. Ainsi, la sélection d'images fiables, pour ce genre de scénario, est une piste intéressante pour l'amélioration des performances du côté du stéganographe.

Dans la littérature actuelle, il existe peu de travaux sur la sélection d'images. La plupart des travaux existants sont des méthodes de stéganographie, dites *par corrélation* ou *par sélection de médium de couverture*, qui reposent sur la notion de corrélation entre le message à cacher et l'image de couverture à sélectionner. Parmi les premières méthodes de sélection d'images de couverture potentiellement sûre pour l'usage stéganographique, nous trouvons celle de [Kermani et Jamzad, 2005]. La méthode consiste à cacher une image secrète dans une autre image, en se basant sur la similarité des textures entre les deux images. Pour ce faire, les auteurs comparent d'abord les blocs de l'image secrète avec un ensemble d'images de couverture, pour choisir l'image qui possède le plus de blocs de texture similaires à ceux de l'image secrète. Une fois que l'image de couverture, la plus similaire, est désignée (considérée comme étant la plus sûre pour porter l'image secrète), l'image secrète est alors concrètement insérée en remplaçant certain blocs de l'image de couverture par les blocs semblables de l'image secrète. Pour pouvoir extraire le message secret, la position des blocs insérés est également mémorisée dans l'image de couverture. Les auteurs de [Kharrazi et al., 2006] se sont également intéressés au problème de sélection d'image de couverture sûre pour l'insertion des données secrètes. Les auteurs ont étudié le problème sous trois scénarios différents : le stéganographe ne connaît pas, connaît partiellement, ou connaît parfaitement la méthode de stéganalyse employée, et ont suggéré alors, comme critères de sélection l'utilisation, de quelques mesures simples telles que : le facteur de qualité JPEG, le nombre de modifications dans l'image de couverture après in-

sersion du message secret, ou même le MSE¹ entre l'image de couverture et l'image stégo. Ces mesures simplistes restent comme même loin de la notion de sécurité au sens de Cachin (voir la section 2.4.1), et ne sont par conséquent pas des critères fiables pour la sélection d'images sûres. [Zheng et Cox, 2007] ont introduit la notion d'entropie conditionnelle entre le message à insérer et une image de couverture donnée. L'objectif étant de choisir, pour la dissimulation, l'image de couverture la plus corrélée au message à cacher, ce qui a pour effet de 1) réduire le nombre de bits à insérer dans l'image hôte, 2) réduire le nombre de modifications effectuées, 3) minimiser la distorsion du support, et donc 4) améliorer la sécurité du processus de dissimulation. Dans le même esprit, [Sajedi et Jamzad, 2009] utilisent un ensemble de différents classifieurs pour déterminer la capacité maximale de bits, qui peut être insérer de façon indétectable, pour chaque image de couverture. Une fois calculée, les auteurs suggèrent alors l'utilisation de cette capacité comme critère de sélection pour choisir l'image de couverture la plus sûre. Le stéganographe, lors de l'insertion ayant le choix entre plusieurs images de couverture, choisit l'image qui a une capacité égale ou supérieure à son message secret, et ce quelque soit la méthode stéganographique utilisée.

Pour rappel, les travaux présentés dans ce manuscrit sont consacrés principalement au scénario du *one-time database*. Autrement dit, nous considérons uniquement le cas où le stéganographe (l'émetteur) transmet une seule image à la fois pour communiquer son message secret. Dans ce qui suit, nous présentons une nouvelle mesure de sécurité pour la sélection des images stéganographiées, basée sur l'exploitation des fonctionnalités de l'oracle ASO (voir le chapitre 5). Contrairement aux méthodes de l'état de l'art présentées précédemment, l'approche proposée n'est pas une méthode de stéganographie par corrélation, dans le sens où elle ne cherche pas à sélectionner l'image de couverture la plus corrélée au message secret, sans prendre en considération le processus stéganographique utilisé. L'approche de sélection basée oracle proposée prend en considération le message inséré, ainsi que les modifications causées par l'algorithme stéganographique utilisé. L'approche proposée prend en entrée un ensemble d'images ayant été précédemment stéganographiées par le même algorithme, et comportant le même message secret, pour choisir en sortie une seule image *stégo*. L'objectif est de sélectionner l'image stéganographiée la plus fiable pour la transmission, celle qui après modification possède une distribution très similaire (qui pourrait être même confondue) à une distribution *cover*. Pour ce faire, nous proposons d'exploiter l'ensemble de classifieur FLD de l'oracle ASO pour calculer un score de sécurité pour chaque images stéganographiée testée. Ainsi, une fois calculé, ce score de sécurité est utilisé comme critère de sélection.

1. MSE : Mean Square Error.

6.3 Nouvelle mesure de sécurité basée oracle pour la sélection des images stégo

Afin de vérifier que la sélection des images stéganographiées améliore encore plus la sécurité de la communication, nous avons élaboré une simple mesure de sélection basée sur l'exploitation des fonctionnalités de l'oracle ASO. Lors de la transmission des informations secrètes, le stéganographe demande à l'oracle de ASO de lui fournir l'image *stégo* la plus sûre, c'est-à-dire celle qui possède une distribution proche des images de couverture. Pour ce faire, l'oracle ASO attribue à chaque image stéganographiée un score reflétant son niveau de sécurité (sa détectabilité). Une méthode, simple et efficace, consiste à compter le nombre de classifieurs FLD [Kodovský *et al.*, 2012] de l'oracle ASO qui ont classé l'image en question, comme étant une image de couverture au lieu d'une image *stégo* (voir la section 4.4.2). La Figure 6.1 illustre le principe de calcul la mesure de sécurité proposée notée S^{FLD} .

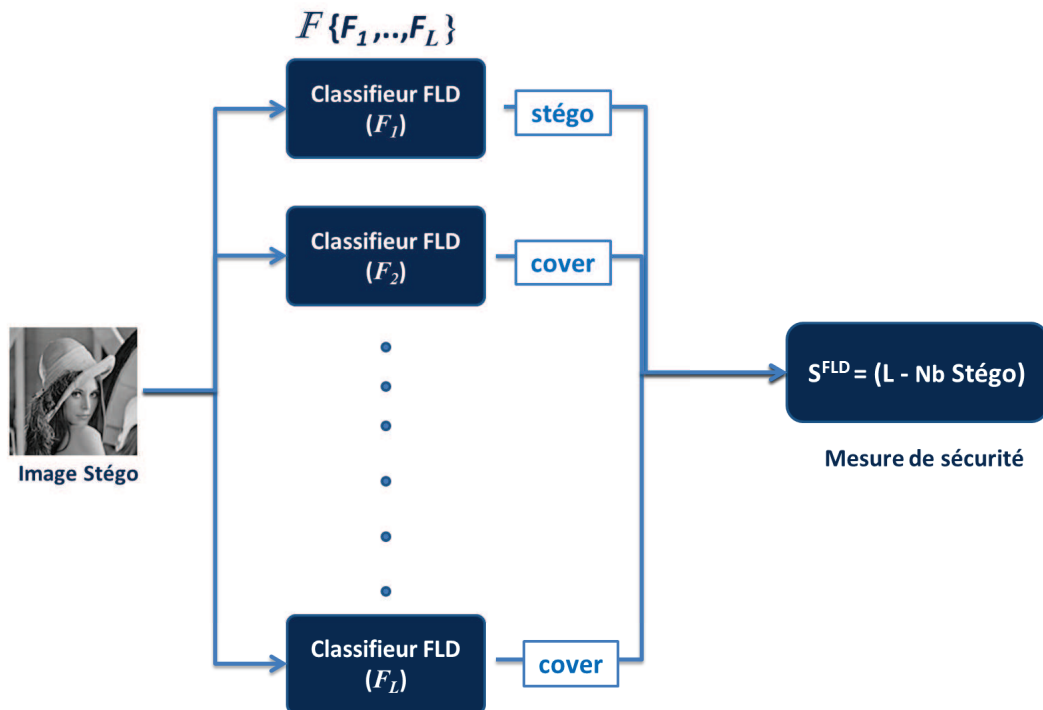


FIGURE 6.1 – Exemple illustrant le principe de la mesure de sécurité S^{FLD} pour la sélection des images *stégo*.

De manière plus formelle nous définissons le score S^{FLD} comme suit :

$$\begin{aligned} S^{\text{FLD}} : \mathcal{I}^{n_1 \times n_2} &\rightarrow \{0, \dots, L\} \\ \mathbf{y} &\rightarrow S^{\text{FLD}}(\mathbf{y}), \end{aligned}$$

tel que :

$$S^{\text{FLD}}(\mathbf{y}) = L - \sum_{l=1}^L F_l(\mathbf{f}_y), \quad (6.1)$$

avec $F_l(\mathbf{f}_y)$ la décision du classifieur F_l (pour ce test la décision est de 1 pour image stéganographiée, et 0 pour image de couverture), et \mathbf{f}_y le vecteur caractéristique représentant l'image \mathbf{y} . Plus le score $S^{\text{FLD}}(\mathbf{y})$ est élevé, plus le niveau de sécurité de l'image stéganographiée \mathbf{y} est meilleur. Notons au passage qu'en utilisant cette mesure de sécurité, nous pouvons obtenir à la fin du processus plusieurs images *stégo* avec le même score. On peut alors choisir aléatoirement l'une de ces images "les plus sûres" ou alors ajouter un critère additionnel. Par exemple, pour avoir une mesure de sélection avec une granularité plus fine, on peut envisager, pour chaque classifieur FLD, de prendre la distance entre le vecteur \mathbf{f}_y et la classe *cover*, au lieu de la décision binaire.

6.4 Tests et résultats

Le but de cette section est de vérifier l'amélioration, en terme de sécurité, qu'apporte la sélection des images stéganographiées au processus stéganographique. Afin de tester l'efficacité de notre stratégie de sélection, expliquée dans la section précédente (section 6.3), nous avons effectué d'abord une itération avec l'algorithme ASO (phase I sur la Figure 5.1), puis construit, pour chaque *payload* de 0.1 *bpp* à 0.5 *bpp*, deux bases d'images différentes constituées chacune de 500 images **stéganographiées avec ASO-1**. La première base, notée $\mathcal{B}_1^{(\alpha)}$, est constituée de 500 images stéganographiées, qui sont sélectionnées de façon aléatoire. La seconde base, notée $\mathcal{B}_2^{(\alpha)}$, est constituée de 500 images stéganographiées, sélectionnées avec la mesure de sécurité S^{FLD} (voir Eq. 6.1). Chaque image des deux bases est représentée en utilisant un vecteur de caractéristiques MINMAX de dimensions $d = 5330$ (pour les paramètres utilisés, voir le Tableau 5.1). Une fois construites, les deux bases $\mathcal{B}_1^{(\alpha)}$ et $\mathcal{B}_2^{(\alpha)}$, de chaque *payload*, sont ensuite analysées avec le SVM monoclasse (OC-SVM, voir la section 4.4.2) de la librairie LIBSVM². L'apprentissage du classifieur SVM monoclasse est effectué sur la base d'images de couverture de BossBase v1.00, en utilisant le noyau

2. LIBSVM : *A Library for Support Vector Machines*, est une librairie dédiée aux Machines à Vecteurs de Support, permettant de faire de l'apprentissage et de la classifications par plusieurs méthodes. La librairie LIBSVM est disponible à l'adresse <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

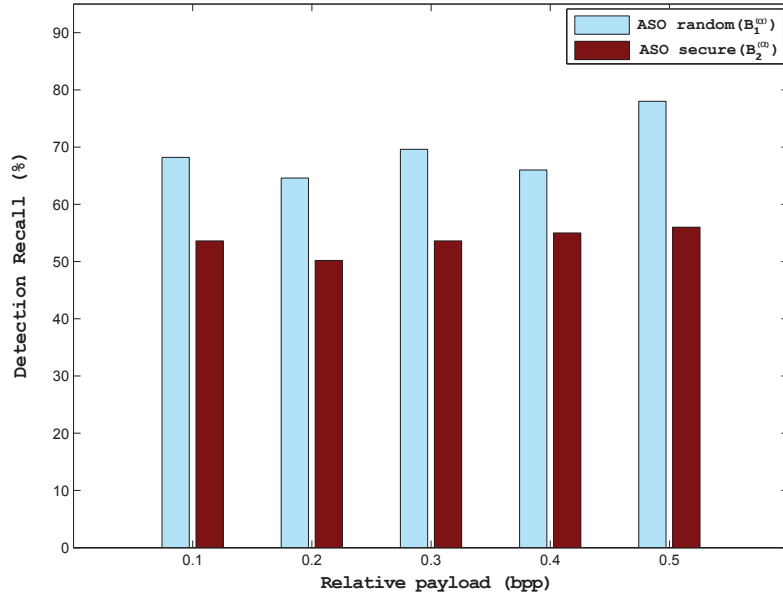


FIGURE 6.2 – Comparaison du niveau de sécurité ($\mathcal{B}_1^{(\alpha)}$ vs $\mathcal{B}_2^{(\alpha)}$) : représentation graphique du rappel de détection de $\mathcal{B}_1^{(\alpha)}$ et de $\mathcal{B}_2^{(\alpha)}$ en fonction du *payload*.

Gaussien $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$, avec $\gamma = 0.181526$, et en fixant le taux de faux positif désiré à $\nu = 0.01$. Pour l'apprentissage et la classification, les valeurs des vecteurs de caractéristiques, représentant les images ont été normalisées entre $[-1, +1]$. Les paramètres utilisés pour la normalisation sont calculées à partir des images utilisées pour l'apprentissage, c'est-à-dire, à partir des images de couverture uniquement (qui sont différentes des images *cover* utilisées pour la stéganographie et présentes dans $\mathcal{B}_1^{(\alpha)}$ et $\mathcal{B}_2^{(\alpha)}$).

En utilisant la classifieur SVM monoclasse pour la stéganalyse des deux bases ($\mathcal{B}_1^{(\alpha)}$ et $\mathcal{B}_2^{(\alpha)}$), nous cherchons à vérifier si les images stéganographiées qui ont été sélectionnées avec la mesure de sécurité S^{FLD} (Eq 6.1) sont plus sûres que celles qui ont été sélectionnées aléatoirement par le stéganographe. En d'autres termes nous cherchons à prouver l'importance de choisir l'image *stégo* la plus fiable lors de la transmission du message secret.

La Figure 6.2 illustre les résultats obtenus de la stéganalyse des deux bases $\mathcal{B}_1^{(\alpha)}$ et $\mathcal{B}_2^{(\alpha)}$, avec le classifieur SMV monoclasse. Comme indiqué sur cette figure, pour tous les *payloads* de 0.1 *bpp* à 0.5 *bpp*, la sécurité de la base d'images *stégo* $\mathcal{B}_2^{(\alpha)}$, qui est construite en utilisant le critère de sécurité S^{FLD} , est meilleure que celle de la base d'images *stégo* sélectionnées aléatoirement $\mathcal{B}_1^{(\alpha)}$. Pour les cinq taux d'insertion analysés, le rappel de détection

R^3 du classifieur SVM monoclasse, pour la base $\mathcal{B}_2^{(\alpha)}$, est plus bas que celui pour la base d'images $\mathcal{B}_1^{(\alpha)}$. Par exemple, à 0.5 *bpp*, le rappel de détection R pour la base $\mathcal{B}_1^{(\alpha)}$ est de 78%, contre seulement 56% pour la base $\mathcal{B}_2^{(\alpha)}$. De même, à 0.4 *bpp*, le rappel de détection R de la base $\mathcal{B}_2^{(\alpha)}$ est inférieur à celui de la base $\mathcal{B}_1^{(\alpha)}$; il est de 55% pour $\mathcal{B}_2^{(\alpha)}$ comparé à 66% pour $\mathcal{B}_1^{(\alpha)}$. En gros, pour tous les *payloads* de 0.1 *bpp* à 0.5 *bpp*, le rappel de détection R de la base d'images est aux alentours de 50 à 55%, ce qui indique que le stéganalysateur SVM monoclasse est entrain de classifier de façon incorrecte, une fois sur deux, une image contenant un message caché comme étant une image *cover* au lieu d'une image *stégo*. Autrement dit, face à la base d'images stéganographiées $\mathcal{B}_2^{(\alpha)}$, le comportement du classifieur SVM monoclasse est totalement aléatoire, car il n'arrive pas à faire la distinction entre les images de couverture et les images stéganographiées. Cela confirme que la base d'images *stégo* $\mathcal{B}_2^{(\alpha)}$, sélectionnée avec le critère S^{FLD} , est plus sûre que la base d'images *stégo* sélectionnées aléatoirement $\mathcal{B}_1^{(\alpha)}$.

Notons au passage, que le rappel de détection R , pour la base d'images $\mathcal{B}_2^{(\alpha)}$ à 0.1 *bpp* est plus élevé que celui à 0.2 *bpp*. Le rappel de détection R est de 53.6% à 0.1 *bpp* contre 50.2% à 0.2 *bpp*. La raison qui explique ce comportement est que l'algorithme ASO n'est pas aussi indétectable pour les faibles taux d'insertion que pour les taux d'insertion plus élevés. Comme déjà mentionné auparavant (section 5.6.1), pour un tel faible taux d'insertion, l'oracle utilisé pour le calcul de la carte de détectabilité a beaucoup de difficulté à distinguer les zones filables pour l'insertion des zones non-fiables, ce qui rend le processus de dissimulation ASO moins performant (Autrement dit plus détectable).

Pour résumer, les résultats de cette expérimentation montrent que l'ensemble des images *stégo* $\mathcal{B}_2^{(\alpha)}$, qui ont été sélectionnées par la mesure de sécurité S^{FLD} , sont plus sûres que l'ensemble d'images *stégo* $\mathcal{B}_1^{(\alpha)}$ sélectionnées aléatoirement. Face aux images sélectionnées, le stéganalysateur utilisé n'arrive pas à faire la distinction entre les images de couverture et les images stéganographiées, ce qui confirme l'importance de la phase de sélection, lors de la communication du message secret, pour le scénario du *one-time database*. Rien qu'en utilisant une simple métrique de sélection, tel que S^{FLD} , nous avons montré que nous pouvons obtenir de meilleures performances en terme de sécurité. Ainsi, nous croyons que l'utilisation d'une mesure de sécurité plus fine, et qui en prend en plus en considération la distribution du message secret, serait encore plus intéressante pour l'amélioration des performances.

3. Le rappel de détection $R = \frac{\text{nombre d'images stégo correctement classifiées}}{\text{nombre total d'images stégo}}$.

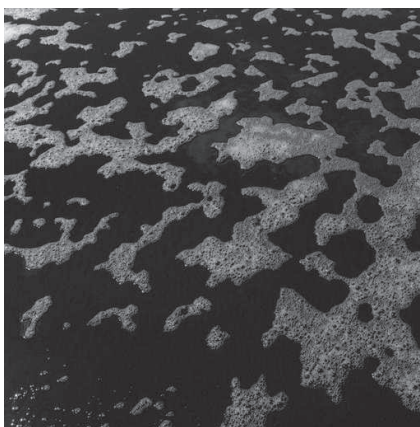
 $S^{\text{FLD}} = 30$  $S^{\text{FLD}} = 29$  $S^{\text{FLD}} = 28$  $S^{\text{FLD}} = 27$

FIGURE 6.3 – Quelques exemples d'images *stégo* sélectionnées par la mesure de sécurité S^{FLD} ($\alpha = 0.5$ et $L = 30$). Plus le score est élevé plus le niveau de sécurité est meilleur.

Notons toutefois que notre analyse a supposé un stéganalysateur n'ayant pas connaissance de notre critère de sélection (stéganalysateur à stratégie pure). Dans le cas où le stéganalysateur connaît le critère, il faut alors faire un test de type SVM binaire (à deux classes, voir section 4.4.2) avec apprentissage sur un ensemble d'images préférentiellement sélectionnées. Cependant, même avec cette stratégie de détection, il est probable que le stéganalysateur ait tout de même des difficultés à séparer les images de couverture des images stéganographiées, puisque les deux distributions *cover* et *stégo* sont très proches (à cause de l'utilisation combinée du mécanisme d'insertion ASO et du critère de sélection S^{FLD}).

La figure 6.3 présente quelques exemples d'images *stégo* qui ont été sélectionnées à l'aide du critère de sélection S^{FLD} , pour un *payload* de $\alpha = 0.5$ et $L = 30$ classifieurs FLD. Comme nous pouvons le constater, les images *stégo* sélectionnées, qui ont été jugées comme étant suffisamment sûres pour l'insertion, correspondent aux images bruitées et texturées.

6.5 Discussion et conclusion

Au cours de ce chapitre, nous avons d'abord introduit une nouvelle philosophie de dissimulation des données secrète : *le paradigme de stéganographie par base*, qui offre à l'utilisateur la possibilité de choisir l'image stéganographiée la plus fiable pour la transmission de son message secret. Pour ce faire, nous nous sommes alors intéressés aux méthodes de sélection d'images *cover*/*stégo*, plus particulièrement pour le scénario du *One-time database*. Ensuite, nous avons proposé une nouvelle mesure de sécurité pour la sélection des images stéganographiées. L'approche de sélection proposée est une approche, qui se base sur les informations de l'ensemble de classifieurs FLD de l'oracle ASO, pour le calcul d'un score de sécurité (S^{FLD}) attribué à chaque image *stégo* testée. Le but étant d'utiliser cette mesure de sécurité comme critère de sélection, pour choisir l'image la plus sûre lors de la transmission du message secret, celle avec une distribution très proche d'une distribution *cover*. Plus le score de sécurité est élevé plus l'image *stégo* sélectionnée est considérée comme étant sûre.

Les résultats expérimentaux ont montré que l'utilisation d'une simple mesure de la sécurité (comme la S^{FLD} proposée en section 6.3), a pour effet d'améliorer considérablement la sécurité de la phase de communication. Une étude approfondie de la sécurité pourrait être envisagée dans nos travaux futurs.

Les travaux présentés dans ce chapitre ont fait l'objet d'un article de conférence internationale [Kouider *et al.*, 2012b].

Conclusions et perspectives

La bataille contre l'ignorance se gagne tous les jours, et elle finit par ouvrir sur des perspectives insoupçonnées.

DALAI LAMA - *Bouddhiste*

Préambule

Dans ce chapitre, nous présentons un bilan général de nos travaux de recherche. Dans un premier temps, nous commençons par résumer les différentes contributions que nous avons apporté au domaine de la stéganographie (section 7.1). Ensuite, nous présentons et discutons les différentes perspectives de ces travaux (section 7.2).

Sommaire

7.1	Résumé des contributions	108
7.2	Perspectives	109

7.1 Résumé des contributions

Il ressort de l'état de l'art présenté dans le chapitre 2 que l'utilisation des techniques de stéganographie par modification du médium de couverture est souvent le choix le plus pratique pour la dissimulation numérique des informations secrètes. Parmi les différents méthodes d'insertion présentées dans ce document, nous nous sommes intéressés plus particulièrement aux méthodes de stéganographie adaptatives, qui sont actuellement les approches les plus efficaces de l'état de l'art. L'objectif de ces méthodes est de modifier le médium hôte, de façon à insérer le message secret, tout en minimisant l'impact d'insertion. Pour ce faire, les méthodes adaptatives actuelles (telles que HUGO, MOD,...) font appel à l'utilisation d'une carte de détectabilité, ρ , qui associe à chaque élément de couverture un coût de détectabilité, $\rho_i \in \mathbb{R}^+$, reflétant son niveau de sécurité après la modification. Cette carte de détectabilité, étant très importante pour l'insertion du message, doit refléter au mieux la détectabilité statistique des éléments modifiée, ce qui représente encore un véritable challenge scientifique, en raison du manque de modèles statistiques complets représentant les média numériques actuelles (images, vidéos, ...) de nature empiriques.

Pour répondre à cette problématique, nous avons présenté dans ce manuscrit une procédure automatique et complète pour l'insertion adaptative de données secrètes dans des images numériques naturelles, nommée ASO (Adaptive Steganography by Oracle). Même si de nombreux travaux ont déjà été proposés récemment sur ce sujet, la plupart de ceux-ci ne prennent en considération que la distribution de l'image de couverture lors de l'insertion du message secret, négligeant ainsi la distribution de la base d'images utilisée par l'émetteur. Notre contribution a donc consisté à développer une approche adaptative générique pour la dissimulation des données secrètes, qui permet de prendre en compte non seulement la distribution de l'image de couverture courante, mais également la distribution de l'ensemble d'images utilisées par l'émetteur. L'objectif étant de préserver à la fois les deux distributions, pour garantir un maximum de sécurité. Pour cela, nous avons proposé :

- **une nouvelle méthode pour le calcul la carte de détectabilité basée oracle.** Lors du calcul de la carte de détectabilité utilisée pour l'insertion, l'approche ASO fait appel à un oracle, qui utilise les fonctionnalités de l'ensemble de classifieurs FLD de [Kodovský *et al.*, 2012] pour scinder l'espace des caractéristiques en deux régions distinctes (*cover* et *stégo*), et exploite ensuite cette séparation pour le calcul des coûts de détectabilité de la carte (section 5.3). Grâce à cette démarche, l'approche ASO permet, lors de la dissimulation du message, de prendre en considération à la fois la distribution de l'image de couverture et la distribution de la base d'images utilisée par l'émetteur.
- **un nouveau schéma générique pour l'insertion des données secrètes.** Afin de répondre au besoin de communication secrète, nous avons proposé un schéma au-

tomatique, et non-paramétrique pour la dissimulation adaptative des données secrètes dans le domaine spatial. Contrairement aux méthodes adaptatives de la littérature, le processus de dissimulation ASO est une approche itérative (voir la Figure 5.1), qui vise à améliorer l'indéteçtabilité du message inséré, et garantir ainsi un meilleur niveau de sécurité. Par ailleurs, grâce à sa conception générique, l'approche ASO peut être considérée comme une méthode de stéganographie générale, qui peut être adaptée à n'importe quel autre algorithme de dissimulation (autre que HUGO), et à n'importe quel vecteur caractéristique, pour encore plus de performances.

- **un nouveau paradigme pour l'insertion des données secrètes.** À travers notre schéma de dissimulation ASO, nous introduisons également un nouveau concept en stéganographie qui est *le paradigme de stéganographie par base d'images* (section 6.1). Lors de l'insertion du message secret, le processus ASO prend en entrée une base d'images de couverture complète pour le calcul de la carte, au lieu d'une seule image, et produit également en sortie un ensemble d'images *stégo* au lieu d'une seule image. À la sortie du système, le stéganographe peut donc prendre une seule image (idéalement la plus sûre si c'est possible) pour communiquer son message secret (scénario du one-time database), ou bien diviser son message secret sur plusieurs images *stégo* (scénario de stéganographie par lot).
- **une nouvelle mesure de sécurité pour la sélection des images stéganographiée.** Afin d'améliorer la sécurité de ASO, nous avons également proposé une nouvelle mesure de sécurité pour la sélection des images stéganographiée, offrant ainsi au stéganographe la possibilité de choisir l'image la plus sûre lors du transmission du message secret.

Toutes ces contributions permettent de définir une nouvelle méthodologie de dissimulation de données. L'approche ASO est une **méta-méthode**, basée **oracle**, pour l'insertion adaptative des message secrets, qui permet de **préserver à la fois** la distribution de l'image de couverture et la distribution de la base d'images utilisée par l'émetteur. Les différentes expérimentations, que nous avons effectuées, ont montré que les performances de notre approche sont nettement supérieures à celles des méthodes déjà existantes dans la littérature. L'approche ASO est donc l'une, si ce n'est la meilleure approche du moment.

7.2 Perspectives

À la suite de cette brève synthèse des travaux présentés dans ce manuscrit, il semble raisonnable de se questionner sur les différentes possibilités envisageables pour perfectionner le processus de dissimulation ASO, et analyser ses performances dans différents scénarios. Pour cela, un certain nombre de pistes pourraient être explorées, dont voici quelques-unes des plus prometteuses :

- **réduction de la complexité de ASO.** Une des limites actuelles de notre approche stéganographique ASO est sa complexité en terme du temps de calcul. Nos futurs travaux consisteront donc, dans un premier temps, à optimiser et à diminuer les temps de calcul du processus de dissimulation ASO. Pour cela, une des solutions possibles qui pourrait être envisagée est l'utilisation de vecteurs caractéristiques de taille petite. Le problème est que l'identification des caractéristiques pertinentes pour la stéganographie/stéganalyse est encore véritable challenge. Pour aborder ce problème, orienter nos travaux futurs sur ce sujet serait alors très intéressant.
- **amélioration de la complétude de ASO.** Bien que, les performances de ASO ont été testées par différentes expérimentations, on ne peut cependant pas garantir sa sécurité face à toutes les attaques de la littérature, et ce en raison de la non-complétude du modèle statistique utilisé pour l'insertion. Une des solutions à ce problème serait alors d'étendre la complétude de ASO par l'utilisation de vecteurs caractéristiques mieux choisis et plus couvrants.
- **une étude approfondie de l'apport des mesures de sélection en stéganographie.** À travers une simple mesure de sélection que nous avons mis en place, nous avons pu montré que la sécurité d'un schéma de stéganographie peut encore être améliorée. Il serait donc très intéressant d'étudier et d'approfondir cette piste dans nos travaux futurs. Des mesures de sélection plus précises, ainsi que des tests plus poussés pourraient être envisagés.
- **Travailler sur une modélisation paramétrique de la carte à partir de ASO.** Le schéma ASO actuel utilise pour l'insertion une carte de détectabilité calculée à partir d'un ensemble de classifieurs FLD entraîné à distinguer entre les images *cover* et les images *stégo* de HUGO. Afin d'éviter toute dépendance à un algorithme d'insertion particulier, il pourrait être envisageable par exemple d'utiliser une carte de détectabilité paramétrique construite à partir de l'analyse des probabilités d'insertion de l'algorithme ASO et de l'algorithme HUGO (voir la section 5.6.3).
- **utilisation du schéma ASO pour le contexte de stéganographie par lot.** Le schéma proposé ASO est par construction très adapté à une utilisation dans un contexte de stéganographie par lot (paradigme de stéganographie par base, section 6.1). Il serait donc très intéressant de tester son comportement et ses performances pour ce genre de scénario.
- **analyse du comportement de ASO dans le contexte de la théorie des jeux.** Avec notre schéma de dissimulation ASO, on pourrait également envisager de simuler un équilibre de Nash en reprenant l'insertion probabiliste des approches de théorie des jeux. Il s'agira donc d'étudier le problème d'insertion adaptative dans un contexte de théorie des jeux, ceci en analysant les différents liens entre la carte de détectabilité (les coûts de détectabilité associés aux pixels) et les différentes stratégies menées par les joueurs de la partie (ici dans notre cas le stéganographe et le stéganalyste).

Pour conclure, cette thèse a permis de confirmer l'intérêt des méthodes d'insertion adaptatives dans le domaine de la stéganographie. Les travaux menés durant ses trois années de doctorat ont permis l'aboutissement d'une nouvelle méthodologie adaptative générale pour l'insertion des données secrètes, ainsi qu'à l'ouverture de nombreuses perspectives dans ce domaine.



Liste des publications

Conférence internationales

- S. Kouider, M. Chaumont et W. Puech : Technical Points About Adaptive Steganography by Oracle (ASO), In *EUSIPCO'2012, 20th European Signal Processing Conference 2012*, pages 1703-1707, Bucharest, Romania, 27-31 août, 2012.
- M. Chaumont et S. Kouider : Steganalysis by Ensemble Classifiers with Boosting by Regression, and Post-Selection of Features, In *ICIP'2012, IEEE International Conference on Image Processing*, Lake Buena Vista (suburb of Orlando), Florida, USA, 30 septembre - 3 octobre, 2012.
- S. Kouider, M. Chaumont et W. Puech : Adaptive Steganography by Oracle (ASO), In *ICME'2013, IEEE International Conference on Multimedia and Expo*, San Jose, California, USA, 15-19 juillet, 2013.

Conférence nationales

- S. Kouider, M. Chaumont, et W. Puech : Stéganographie Adaptative par Oracle (ASO), In *CORESA'2012, COmpression et REprésentation des Signaux Audiovisuels*, Lille, France, 24-25 mai, 2012.

Communications orales

- S. Kouider, M. Chaumont, et W. Puech : Stéganographie Adaptative par Oracle (ASO), *Journées Codes et Stéganographie*, Hôtel de la Monnaie, Rennes, France, 18-19 janvier, 2011.
- S. Kouider, M. Chaumont, et W. Puech : Présentation de l'impact d'insertion en stéganographie, *Journées Codes et Stéganographie*, Écoles Militaires de Saint-Cyr Coëtquidan, Rennes, France, 19-20 mars, 2012.
- S. Kouider, M. Chaumont, et W. Puech : Utilisation des outils de classification pour la stéganographie de bases de données visuelles (BDV), *Journée commune TRECVID + Qualité et Protection*, GDR-ISIS, Paris, France, 17 janvier, 2013.



Bibliographie

- [MR, 2000] (2000). Les efforts de la NSA vis-à-vis du Web : la stéganographie. *Le Monde du Renseignement*. Cité page 13.
- [AFP, 2001] (2001). Des messages cachés sur l'internet pour préparer les attentats. Agence Française de Presse. Cité page 13.
- [Abbrugiati, 2006] Abbrugiati, P. (2006). Introduction aux codes correcteurs d'erreurs. Cité page 38.
- [Ahn et Hopper, 2004] Ahn, L. et Hopper, N. (2004). Public-Key Steganography. *In Eurocrypt 2004*, volume 3027 de *Lecture Notes in Computer Science*, pages 323–341, Interlaken, Switzerland. Cité page 19.
- [Astrowsky, 2000] Astrowsky, B. (2000). STEGANOGRAPHY Hidden Images, A New Challenge in the Fight Against Child Porn. *UPDATE*, 13(2). Cité page 13.
- [B. Roue, 2005] 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.
- [Bajorski, 2011] Bajorski, P. (2011). *Statistics for Imaging, Optics, and Photonics*. Wiley, Rochester, New York, United States. Cité page 82.
- [Barbier et Alt, 2008] Barbier, J. et Alt, S. (2008). Practical Insecurity for Effective Steganalysis. *In Information Hiding - 10th International Workshop*, volume 5284 de *Lecture Notes in Computer Science, IH'08*, pages 195–2008, Santa Barbara, (CA) USA. Springer-Verlag. Cité page 20.

- [Barbier *et al.*, 2009] Barbier, J., Alt, S. et Mayer, E. (2009). Modèles de Sécurité en Stéganographie. *In Proc. of Workshop Interdisciplinaire sur la Sécurité Globale, WISG'09*, Troyes, France. Cité page 20.
- [Bas *et al.*, 2011] Bas, P., Filler, T. et Pevný, T. (2011). Break Our Steganographic System — the ins and outs of organizing BOSS. *In Information Hiding - 13th International Workshop*, volume 6958 de *Lecture Notes in Computer Science, IH'11*, pages 59–70, Prague, Czech Republic. Springer-Verlag. Cité pages 30 et 33.
- [Bierbrauer, 2004] Bierbrauer, J. (2004). *Introduction to Coding Theory*. Chapman & Hall/CRC. Cité page 24.
- [Bierbrauer et Fridrich, 2008] Bierbrauer, J. et Fridrich, J. (2008). Transactions on data hiding and multimedia security iii. chapitre Constructing good covering codes for applications in steganography, pages 1–22. Springer-Verlag, Berlin, Heidelberg. Cité page 24.
- [Cachin, 1998] Cachin, C. (1998). An Information-Theoretic Model for Steganography. *In Information Hiding - 2ed International Workshop*, volume 1525, pages 306–318, Portland, Oregon, USA. Springer-Verlag. Cité pages 2 et 19.
- [Cachin, 2004] Cachin, C. (2004). An Information-Theoretic Model for Steganography. *Information and Computation*, 192(1):41–56. Cité page 19.
- [Cachin, 2011] Cachin, C. (2011). Digital Steganography. *In van Tilborg, H. et Jajodia, S., éditeurs : Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 348–352. Springer. Cité page 19.
- [Chaumont et Kouider, 2012] Chaumont, M. et Kouider, S. (2012). Steganalysis by Ensemble Classifiers with Boosting by Regression, and Post-Selection of Features. *In ICIP'2012, IEEE International Conference on Image Processing*, Lake Buena Vista (suburb of Orlando), Florida, USA. Cité page 69.
- [Cogranne, 2011] 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.
- [Comesana et Pérez-González, 2007] Comesana, P. et Pérez-González, F. (2007). On the capacity of stegosystems. *In Multimedia and Security Workshop, MM&Sec '07 Proceedings of the 9th ACM multimedia*, pages 15–24, Dallas, Texas, USA. ACM. Cité page 22.
- [Cortes et Vapnik, 1995] Cortes, C. et Vapnik, V. (1995). Support-Vector Networks. *Mach. Learn.*, 20(3):273–297. Cité page 63.
- [Crandall, 1998] Crandall, R. (1998). Some notes on steganography. Steganography Mailing List. Cité pages 24 et 41.

- [Craver, 1998] Craver, S. (1998). On Public-key Steganography in the Presence of an Active Warden. In *Information Hiding - 2nd International Workshop*, volume 1525 de *Lecture Notes in Computer Science, IH'98*, pages 355–368, Portland, Oregon, USA. Springer-Verlag. Cité page 51.
- [Dijk et Willems, 2001] Dijk, M. et Willems, F. (2001). Embedding Information in Grayscale Images. In *22nd Symposium Information and Communication Theory*, pages 147–154, Benelux, Enschede, The Netherlands. Cité page 24.
- [Djeffal, 2012] Djeffal, A. (2012). *Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données*. Thèse de doctorat, Université Mohamed Khider - Biskra. Cité pages 66 et 67.
- [Dong et al., 2009] Dong, J., Wang, W. et Tan, T. (2009). Multi-class Blind Steganalysis Based on Image Run-Length Analysis. In *Digital watermarking, the 8th international conference, IWDW'09*, pages 199–210, Berlin, Heidelberg. Springer-Verlag. Cité page 67.
- [Ettinger, 1998] 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. Cité page 71.
- [Fillatre, 2011] Fillatre, L. (2011). *Contributions en Détection et Classification Statistique Paramétrique*. Habilitation à diriger des recherches, Université de Technologie de Compiègne. Cité page 71.
- [Filler, 2011] Filler, T. (2011). *IMPERFECT STEGOSYSTEMS – ASYMPTOTIC LAWS AND NEAR-OPTIMAL PRACTICAL CONSTRUCTIONS*. Thèse de doctorat, Binghamton University. Cité page 50.
- [Filler et Fridrich, 2010] Filler, T. et Fridrich, J. (2010). Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4):705–720. Cité pages 30 et 31.
- [Filler et Fridrich, 2011] Filler, T. et Fridrich, J. (2011). Design of Adaptive Steganographic Schemes for Digital Images. In *Media Watermarking, Security, and Forensics XIII, part of IS&T SPIE Electronic Imaging Symposium*, volume 7880, paper. 13, pages F 1–14, San Francisco, CA. Cité pages 28, 34 et 77.
- [Filler et al., 2010] Filler, T., Judas, J. et Fridrich, J. (2010). Minimizing Embedding Impact in Steganography using Trellis-Coded Quantization. In *Media Forensics and Security II, part of IS&T SPIE Electronic Imaging Symposium*, volume 7541, paper. 05, San Jose, CA, USA. Cité pages 29, 31 et 33.
- [Filler et al., 2011] Filler, T., Judas, J. et Fridrich, J. (2011). Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes. *IEEE Transactions on Information Forensics and Security*, 6(3-2):920–935. Cité pages 29, 31, 33, 46 et 85.

- [Filler *et al.*, 2009] Filler, T., Ker, A. et Fridrich, J. (2009). The square root law of steganographic capacity for Markov covers. *In Media Forensics and Security I, part of IS&T SPIE Electronic Imaging Symposium*, volume 7254, San Jose, CA, USA. Cité page 23.
- [Frezza-Buet, 2012] Frezza-Buet, H. (2012). Machines à Vecteurs Supports Didacticiel. Support de cours, Supélec, France. Cité page 64.
- [Fridrich, 2009] Fridrich, J. (2009). *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA. Cité pages 2, 13, 16, 17, 20, 21, 27 et 60.
- [Fridrich et Filler, 2007] Fridrich, J. et Filler, T. (2007). Practical Methods for Minimizing Embedding Impact in Steganography. *In Security, Steganography, and Watermarking of Multimedia Contents IX, part of IS&T SPIE Electronic Imaging Symposium*, volume 6505, pages 02–03, San Jose, CA. Cité page 31.
- [Fridrich *et al.*, 2002a] Fridrich, J., Goljan, M. et Hogeia, D. (2002a). Attacking the OutGuess. *In Multimedia and Security Workshop, MM&Sec '02 Proceedings of the 5th ACM multimedia*, Juan-les-Pins, France. ACM. Cité page 59.
- [Fridrich *et al.*, 2002b] Fridrich, J., Goljan, M. et Hogeia, D. (2002b). Steganalysis of JPEG Images : Breaking the F5 Algorithm. *In Information Hiding - 5th International Workshop*, volume 2578 de *Lecture Notes in Computer Science, IH'02*, pages 310–323, Noordwijkerhout, The Netherlands. Springer-Verlag. Cité page 59.
- [Fridrich *et al.*, 2003] Fridrich, J., Goljan, M., Hogeia, D. et Soukal, D. (2003). Quantitative steganalysis of digital images: estimating the secret message length. *Multimedia Systems*, 9(3):288–302. Cité page 52.
- [Fridrich *et al.*, 2005] Fridrich, J., Goljan, M., Lisonek, P. et Soukal, D. (2005). Writing on Wet Paper. *In Security, Steganography, and Watermarking of Multimedia Contents VII, part of IS&T SPIE Electronic Imaging Symposium*, volume 5681, pages 328–340, San Jose, California, USA. Cité pages 24, 33, 44, 45 et 76.
- [Fridrich et Kodovský, 2012] Fridrich, J. et Kodovský, J. (2012). Rich Models for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882. Cité pages 52, 53, 61, 63, 87 et 91.
- [Fridrich *et al.*, 2011a] 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. Cité pages 52, 53, 62, 63, 85 et 86.
- [Fridrich *et al.*, 2011b] Fridrich, J., Kodovský, J., Holub, V. et Goljan, M. (2011b). Steganalysis of content-adaptive steganography in spatial domain. *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. Cité pages 53, 61 et 63.
- [Fridrich *et al.*, 2001] Fridrich, J. J., Goljan, M. et Du, R. (2001). Detecting LSB Steganography in Color and Gray-Scale Images. *IEEE MultiMedia*, 8(4):22–28. Cité page 58.
- [Galand et Kabatiansky, 2003] Galand, F. et Kabatiansky, G. (2003). Information Hiding by Coverings. *In IEEE Information Theory Workshop, ITW2003*, pages 151–154, Paris, France. Cité page 24.
- [Guan *et al.*, 2011] Guan, Q., Dong, J. et Tan, T. (2011). Blind quantitative steganalysis based on feature fusion and gradient boosting. *In Digital watermarking, the 9th international conference, IWDW'10*, pages 266–279, Seoul, Korea. Springer-Verlag. Cité page 52.
- [Guermeur, 2007] Guermeur, Y. (2007). *SVM Multiclasses, Théorie et Applications*. Habilitation à diriger des recherches, Nancy I. Cité page 67.
- [Gul et Kurugollu, 2011] Gul, G. et Kurugollu, F. (2011). A New Methodology in Steganalysis : Breaking Highly Undetectable Steganography (HUGO). *In Information Hiding - 13th International Workshop, Lecture Notes in Computer Science, IH'11*, pages 71–84, Prague, Czech Republic. Springer-Verlag. Cité page 53.
- [Hasan et Boris, 2006] Hasan, M. et Boris, F. (2006). *Svm : Machines à vecteurs de support ou séparateurs à vastes marges*. Rapport technique, Versailles St Quentin, France. Cité page 64.
- [Hérodote, 1985] Hérodote (1985). *L'Enquête : Livres I à IV*, volume 1 de *Collection Folio*. Traduit par A. Barguet. Editions Gallimard. Cité page 10.
- [Hérodote, 1990] Hérodote (1990). *L'Enquête : Livres V à IX*, volume 2 de *Collection Folio*. Traduit par A. Barguet. Editions Gallimard. Cité page 10.
- [Holub et Fridrich, 2012] Holub, V. et Fridrich, J. (2012). Designing Steganographic Distortion Using Directional Filters. *In IEEE Workshop on Information Forensic and Security, WIFS'12*, Tenerife, Spain. Cité pages 28, 34, 52 et 76.
- [Holub et Fridrich, 2013] Holub, V. et Fridrich, J. (2013). Digital Image Steganography Using Universal Distortion. *In Proceedings of the first ACM workshop on Information hiding and multimedia security, IH&MMSec '13*, pages 59–68, Montpellier, France. ACM. Cité pages 28, 34 et 76.
- [Holub *et al.*, 2013] Holub, V., Fridrich, J. et Denmark, T. (2013). Random Projections of Residuals as an Alternative to Co-occurrences in Steganalysis. *In Media Watermarking, Security, and Forensics XV, part of IS&T SPIE Electronic Imaging Symposium*, volume 8665, San Francisco, California, USA. Cité page 61.

- [Hopper, 2004] Hopper, N. (2004). *Toward a Theory of Steganography*. Thèse de doctorat, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Cité page [19](#).
- [Hopper, 2005] Hopper, N. (2005). On Steganographic Chosen Coverttext Security. In Caires, L., Italiano, G., Monteiro, L., Palamidessi, C. et Yung, M., éditeurs : *Automata, Languages and Programming, 32nd International Colloquium, ICALP*, volume 3580 de *Lecture Notes in Computer Science*, pages 311–323, Lisbon, Portugal. Springer. Cité page [19](#).
- [Hopper et al., 2002] Hopper, N., Langford, J. et Ahn, L. (2002). Provably Secure Steganography. In *Crypto 2002*, volume 2442 de *Lecture Notes in Computer Science*, pages 77–92, Santa Barbara, CA, USA. Cité page [19](#).
- [Judge, 2001] Judge, J. (2001). Steganography : Past, Present and Future. SANS. Cité page [13](#).
- [Kahn, 1996] Kahn, D. (1996). The History of Steganography. In *Information Hiding - 1st International Workshop*, volume 1174 de *Lecture Notes in Computer Science, IH'96*, pages 1–5, Cambridge, U.K. Springer-Verlag. Cité page [13](#).
- [Katzenbeisser et Petitcolas, 2000] Katzenbeisser, S. et Petitcolas, F. A. P. (2000). *Information Hiding Techniques for Steganography and Digital Watermarking*. Cambridge University Press, Norwood, MA, USA, 1st édition. Cité page [61](#).
- [Kelley, 2001] Kelley, J. (2001). Terrorist instructions hidden online. USA Today. Cité page [13](#).
- [Ker, 2006] Ker, A. (2006). Batch steganography and pooled steganalysis. In *Information Hiding - 8th International Workshop*, volume 4437 de *Lecture Notes in Computer Science, IH'06*, pages 265–281, Alexandria, VA, USA. Springer-Verlag. Cité pages [53](#) et [98](#).
- [Ker, 2007a] Ker, A. (2007a). A Capacity Result for Batch Steganography. *Signal Processing Letters*, 14(8):525–528. Cité page [22](#).
- [Ker, 2007b] Ker, A. (2007b). The Ultimate Steganalysis Benchmark? In *Multimedia and Security Workshop, MM&Sec '07 Proceedings of the 9th ACM multimedia*, pages 141–148, Dallas, Texas, USA. ACM. Cité page [20](#).
- [Ker, 2010] Ker, A. (2010). The Square Root Law In Stegosystems With Imperfect Information. In *Information Hiding - 12th International Workshop*, volume 6387 de *Lecture Notes in Computer Science, IH'10*, pages 145–160, Calgary, Alberta, Canada. Springer-Verlag. Cité page [23](#).
- [Ker et al., 2013] Ker, A., Bas, P., Böhme, R., Cogramme, R., Craver, S., Filler, T., Fridrich, J. et Pevný, T. (2013). Moving Steganography and Steganalysis from the Laboratory into Real World. In *Proceedings of the first ACM workshop on Information hiding and multimedia security, IH&MMSec '13*, pages 45–58, Montpellier, France. ACM. Cité page [29](#).

- [Ker et Pevný, 2011] 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. Cité page 54.
- [Ker et Pevný, 2012a] Ker, A. et Pevný, T. (2012a). Batch Steganography in the Real World. In *Multimedia and Security Workshop, MM&Sec '12 Proceedings of the 14th ACM multimedia*, pages 1–10, Coventry, United Kingdom. ACM. Cité page 54.
- [Ker et Pevný, 2012b] Ker, A. et Pevný, T. (2012b). Identifying a steganographer in realistic and heterogeneous data sets. In *Media Watermarking, Security, and Forensics IV, part of IS&T SPIE Electronic Imaging Symposium*, volume 8303, San Francisco, California, USA. Cité page 54.
- [Kerckhoffs, 1883] Kerckhoffs, A. (1883). La cryptographie militaire. *Journal des sciences militaires*, IX(3):5–83. Cité page 51.
- [Kermani et Jamzad, 2005] Kermani, Z. et Jamzad, M. (2005). A Robust Steganography Algorithm Based On Texture Similarity Using Gabor Filter. In *IEEE Symposium on Signal processing and Information Technology*, pages 578–582. Cité page 99.
- [Kharrazi et al., 2006] Kharrazi, M., Sencar, T. et Memo, N. (2006). Cover Selection for Steganographic Embedding. In *ICIP'2006, IEEE International Conference on Image Processing*, pages 117–120, Atlanta, GA, USA. Cité page 99.
- [Kodovský et al., 2011] Kodovský, J., Filler, T., Fridrich, J. et Holub, V. (2011). On Dangers of Overtraining Steganography to Incomplete. Cover Model. In *Multimedia and Security Workshop, MM&Sec '11 Proceedings of the 13th ACM multimedia*, pages 69–76, Buffalo, NY, USA. ACM. Cité pages 34, 52 et 76.
- [Kodovský et Fridrich, 2009] Kodovský, J. et Fridrich, J. (2009). Calibration revisited. In *Multimedia and Security Workshop, MM&Sec '09 Proceedings of the 11th ACM multimedia*, pages 63–74, Princeton, New Jersey, USA. ACM. Cité page 59.
- [Kodovský et Fridrich, 2011] Kodovský, J. 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. Cité pages 53, 68, 69, 72, 80, 91 et 96.
- [Kodovský et Fridrich, 2012a] Kodovský, J. et Fridrich, J. (2012a). JPEG-Compatibility Steganalysis Using Block-Histogram of Recompression Artifacts. In *Information Hiding - 14th International Workshop*, volume 7692 de *Lecture Notes in Computer Science, IH'12*, pages 78–93, Berkeley, CA, USA. Springer-Verlag. Cité page 52.
- [Kodovský et Fridrich, 2012b] Kodovský, J. et Fridrich, J. (2012b). Steganalysis of JPEG Images Using Rich Models. In *Media Watermarking, Security, and Forensics XIV, part of*

- IS&T SPIE Electronic Imaging Symposium*, volume 8303, San Francisco, California, USA. Cité page 61.
- [Kodovský *et al.*, 2008] Kodovský, J., Fridrich, J. et Holub, V. (2008). On Completeness of Feature Spaces in Blind Steganalysis. In *Multimedia and Security Workshop, MM&Sec '08 Proceedings of the 10th ACM multimedia*, pages 123–132, Oxford, UK. ACM. Cité pages 28, 77 et 90.
- [Kodovský *et al.*, 2012] Kodovský, J., Fridrich, J. et Holub, V. (2012). Ensemble Classifiers for Steganalysis of Digital Media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444. Cité pages 52, 53, 68, 69, 77, 87, 98, 101 et 108.
- [Kouider *et al.*, 2012a] Kouider, S., Chaumont, M. et Puech, W. (2012a). Stéganographie Adaptative par Oracle (ASO). In *Compression et REprésentation des Signaux Audiovisuels, CORESA'12*, Lille, France. Cité page 96.
- [Kouider *et al.*, 2012b] Kouider, S., Chaumont, M. et Puech, W. (2012b). Technical Points About Adaptive Steganography by Oracle (ASO). In *EUSIPCO'2012, 20th European Signal Processing Conference 2012*, pages 1703–1707, Bucharest, Romania. Cité page 106.
- [Kouider *et al.*, 2013] Kouider, S., Chaumont, M. et Puech, W. (2013). Adaptive Steganography by Oracle (ASO). In *IEEE International Conference on Multimedia and Expo, ICME'2013*, San Jose, California, USA. Cité page 96.
- [Le et Kurosawa, 2003] Le, T. V. et Kurosawa, K. (2003). Efficient Public Key Steganography Secure Against Adaptively Chosen Stegotext Attack. *IACR Cryptology ePrint Archive*, 2003:244. Cité page 19.
- [Lubenko et Ker, 2012a] 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. Cité pages 53 et 69.
- [Lubenko et Ker, 2012b] Lubenko, I. et Ker, A. (2012b). Steganalysis with mismatched covers : do simple classifiers help? In *Multimedia and Security Workshop, MM&Sec '12 Proceedings of the 14th ACM multimedia*, pages 11–18, Coventry, United Kingdom. ACM. Cité pages 53 et 69.
- [Miche *et al.*, 2010] 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. Cité page 52.
- [Pasquet *et al.*, 2013] Pasquet, J., 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. Cité pages 53 et 70.

- [Peterson et Weldon, 1972] Peterson, W. et Weldon, E. (1972). *Error-correcting Codes*. MIT Press. Cité page 38.
- [Pevný, 2008] Pevný, T. (2008). *Kernel Methods in Steganalysis*. Thèse de doctorat, Binghamton University. Cité page 68.
- [Pevný *et al.*, 2010] Pevný, T., Bas, P. et Fridrich, J. (2010). Steganalysis by Subtractive Pixel Adjacency Matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224. Cité pages 61, 62 et 68.
- [Pevný *et al.*, 2010] Pevný, T., Filler, T. et Bas, P. (2010). Using High-Dimensional Image Models to Perform. Highly Undetectable Steganography. In *Information Hiding - 12th International Conference*, volume 6387 de *Lecture Notes in Computer Science, IH'10*, pages 161–177, Calgary, AB, Canada. Springer-Verlag. Cité pages 28, 30, 33, 68, 76, 80 et 85.
- [Pevný et Fridrich, 2006] 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. Cité page 67.
- [Pevný et Fridrich, 2007a] Pevný, T. et Fridrich, J. (2007a). Merging Markov and DCT Features for Multi-Class JPEG Steganalysis. In *Media Watermarking, Security, and Forensics IX, part of IS&T SPIE Electronic Imaging Symposium*, volume 6505, pages 03–04, San Francisco, California, USA. Cité pages 59 et 67.
- [Pevný et Fridrich, 2007b] Pevný, T. et Fridrich, J. (2007b). Merging Markov and DCT features for multiclass JPEG steganalysis. In *Media Watermarking, Security, and Forensics IX, part of IS&T SPIE Electronic Imaging Symposium*, volume 6505, San Francisco, California, USA. Cité page 61.
- [Pevný et Fridrich, 2008] Pevný, T. et Fridrich, J. (2008). Novelty detection in blind steganalysis. In *workshop on Multimedia and security, part of MM&Sec'08 Proceedings of the 10th ACM multimedia*, pages 167–176, New York, NY, USA. ACM. Cité page 53.
- [Pevný *et al.*, 2012] Pevný, T., Fridrich, J. et Ker, A. (2012). From Blind to Quantitative Steganalysis. *IEEE Transactions on Information Forensics and Security*, 7(2):445–454. Cité page 52.
- [Pfitzmann, 1996] Pfitzmann, B. (1996). Information Hiding Terminology. In *Proceedings of the first International Workshop on Information Hiding*, numéro 1174 de IH '96, pages 347–350, Cambridge, England. Springer Verlag. Cité page 14.
- [Provos, 2001] Provos, N. (2001). Defending Against Statistical Steganalysis. In *The 10th USENIX Security Symposium*, Washington, D.C., USA. Cité page 27.
- [Provos et Honeyman, 2001] Provos, N. et Honeyman, P. (2001). Detecting steganographic content on the internet. Rapport technique, University of Michigan. Cité page 57.

- [Renold *et al.*, 2003] Renold, E., Creighton, S., Atkinson, C. et Carr, J. (2003). Images of abuse : A review of the evidence on child pornography. Rapport technique, National Society for the Prevention of Cruelty to Children (NSPCC). Cité page 13.
- [Roth, 2006] Roth, R. (2006). *Introduction to Coding Theory*. Cambridge University Press. Cité page 38.
- [Sajedi et Jamzad, 2009] Sajedi, H. et Jamzad, M. (2009). Secure Cover Selection Steganography. In *Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance, ISA '09*, pages 317–326, Seoul, Korea. Springer-Verlag. Cité page 100.
- [Salas, 2010] Salas, H. (2010). LA STEGANOGRAPHIE MODERNE : L'ART DE LA COMMUNICATION SECRETE. Mémoire de Master, Laboratoire d'Informatique de Robotique et de Micro- électronique de Montpellier, Université de Montpellier 2. Non cité.
- [Schaathun, 2012] Schaathun, H. (2012). *Machine Learning in Image Steganalysis*. Wiley-IEEE Press. Cité page 60.
- [Schöttle et Böhme, 2012] Schöttle, P. et Böhme, R. (2012). A Game-Theoretic Approach to Content-Adaptive Steganography. In *Information Hiding - 14th International Workshop*, volume 7692 de *Lecture Notes in Computer Science, IH'12*, pages 125–141, Berkeley, CA, USA. Springer-Verlag. Cité page 71.
- [Sharp, 2001] Sharp, T. (2001). An implementation of key-based digital signal steganography. In *Information Hiding - 4th International Workshop*, volume 2137 de *Lecture Notes in Computer Science, IH'01*, pages 13–26, Berlin, Heidelberg. Springer-Verlag. Cité page 26.
- [Sieberg, 2001] Sieberg, D. (2001). Bin Laden exploits technology to suit his needs. CNN. Cité page 13.
- [Simmons, 1983] Simmons, G. (1983). The prisoners' problem and the subliminal channel. In *Advances in Cryptology. Proc. of Crypto'83*, pages 51–67. Plenum Press, New York. Cité pages 2, 14, 50 et 98.
- [Upham, 1997] 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. Cité pages 27 et 57.
- [van Damme, 1991] van Damme, E. (1991). *Stability and Perfection of Nash Equilibria*. Springer-Verlag. Cité page 71.
- [Vapnik, 1995] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc, New York, NY, USA. Cité page 63.

- [Wang et Moulin, 2008] Wang, Y. et Moulin, P. (2008). Perfectly Secure Steganography : Capacity, Error Exponents, and Code Constructions. *IEEE Transactions on Information Theory, Special Issue on Security*, 55(6):2706–2722. Cité pages 17 et 22.
- [Watrigan, 2009] Watrigan, R. (2009). Utilisation des codes correcteurs d’erreurs en stéganographie : De l’algorithme F5 et sa stéganalyse aux codes à papier mouillé. Rapport L3, Université de Nîmes. Non cité.
- [Westfeld, 2001] 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. Cité pages 24, 27, 33, 41, 59 et 76.
- [Westfeld et Pfitzmann, 1999] Westfeld, A. et Pfitzmann, A. (1999). Attacks on Steganographic Systems - Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools—and Some Lessons Learned. In *Information Hiding - 3th International Workshop*, volume 1768 de *Lecture Notes in Computer Science, IH’99*, pages 61–76, Dresden, Germany. Springer-Verlag. Cité pages 26, 55 et 57.
- [Zheng et Cox, 2007] Zheng, L. et Cox, J. (2007). JPEG Based Conditional Entropy Coding for Correlated Steganography. In *IEEE International Conference on Multimedia and Expo, ICME’2007*, pages 1251–1254, Beijing, China. Cité page 100.
- [Zitzmann, 2013] Zitzmann, C. (2013). *Détection statistique d’information cachée dans des images naturelles*. Thèse de doctorat, Université de Technologie de Troyes (UTT). Cité pages 70 et 71.



Table des figures

2.1	La stéganographie linguistique	11
2.2	La stéganographie dans le domaine artistique	12
2.3	Utilisation des partitions musicales pour la stéganographie.	13
2.4	Schéma stéganographique par modification	18
2.5	Décomposition en plans de bits d'une image en niveaux de gris	25
2.6	LSB substitution	26
2.7	LSB Matching	27
2.8	LSB Matching	28
2.9	Distorsion non-additive	30
3.1	Syndrome Trellis Codes (STC)	47
4.1	Exemple d'un histogramme d'une image avant et après insertion LSB	56
4.2	SVM binaire	65
4.3	Exemple illustrant le cas où les données sont non-linéairement séparables	65
4.4	Exemple de classifieurs SVM multiclassés	66
4.5	SVM binaire	67
5.1	Schéma de Stéganographie Adaptative par Oracle (ASO)	78
5.2	Illustration du principe de fonctionnement d'un classifieur FLD	81
5.3	Illustration du principe de calcul des variations des vecteurs caractéristiques.	84
5.4	Niveau de sécurité de ASO vs HUGO, pour le vecteur caractéristique SRMQ1 de dimension 12753	89
5.5	Niveau de sécurité de ASO vs HUGO, pour le vecteur caractéristique SRM de dimension 34671	89

5.6	La probabilité d'insertion à 0.2 de HUGO vs ASO-1	93
5.7	Analyse des cartes de détectabilité de HUGO et ASO-1	94
6.1	SVM binaire	101
6.2	Comparaison du niveau de sécurité ($\mathcal{B}_1^{(\alpha)}$ vs $\mathcal{B}_2^{(\alpha)}$)	103
6.3	Mesure de sécurité SS^{FLD}	105

Liste des tableaux

5.1	Paramètres des caractéristiques MINMAX utilisées pour ASO	86
5.2	Comparaison des performances des 3 itérations de l'algorithme ASO	91

Abstract

Steganography is the art of secret communication. The goal is to hide a secret message in an unsuspecting object in such a way that no one can detect it. Nowadays, with the Internet spread and the emergence of digital supports (audio files, videos, or images), several philosophies of designing steganographic methods were proposed. One of the most usual embedding methods used with real digital images is the adaptive embedding algorithms, which is based on the modification of the cover image with a guarantee of a certain security level. These methods represent an important progress in steganography.

In this Ph.D. Thesis, we present a fully automated procedure for the adaptive embedding of secret data in digital images. For this, after recalling the recent concepts of adaptive steganography, we first introduce a clear formalism to define a new "meta-method" steganographic approach based on "oracle", which we called ASO (Adaptive Steganography by Oracle). Then, we define a new steganographic paradigm called "the steganography by database paradigm", and propose a new selection criterion to further enhance the security of the transmission phase of ASO. Experimental results show that our embedding approach ASO provides the highest level of steganographic security. It is then currently the best or one of the best approaches of the state of the art.

Keywords: *Steganography, Steganalysis, Ensemble Classifiers, Oracle, Detectability map, Security.*

Résumé

La stéganographie est l'art de la communication secrète. L'objectif est de dissimuler un message secret dans un médium anodin de sorte qu'il soit indétectable. De nos jours, avec la généralisation d'Internet et l'apparition des supports numériques (fichiers audio, vidéos ou images), plusieurs philosophies de conception de schéma stéganographique ont été proposées. Parmi les méthodes actuelles appliquées aux images numériques naturelles, nous trouvons les méthodes d'insertion adaptative, dont le principe repose sur la modification du médium de couverture avec une garantie d'avoir un certain niveau de sécurité. Ces méthodes représentent une véritable avancée en stéganographie.

Dans ce manuscrit, après avoir rappelé les concepts récents de stéganographie adaptative, nous présentons une procédure automatique et complète pour l'insertion adaptative de données secrètes dans des images numériques naturelles. L'approche proposée est une « méta-méthode » basée « oracle », appelée ASO (Adaptive Steganography by Oracle), qui permet de préserver à la fois la distribution de l'image de couverture et la distribution de la base d'images utilisée par l'émetteur. Notre approche permet d'obtenir des résultats nettement supérieurs aux méthodes actuelles de l'état de l'art, et est donc l'une, si ce n'est la meilleure approche du moment. Par ailleurs, nous définissons également un nouveau paradigme en stéganographie qui est la stéganographie par base, ainsi qu'une nouvelle mesure de sélection pour les images stéganographiées, permettant d'améliorer encore plus les performances de notre schéma d'insertion. Les différentes expérimentations, que nous avons effectuées sur des images réelles, ont confirmé la pertinence de cette nouvelle approche.

Mots clefs : *Stéganographie, Stéganalyse, Ensemble de classifieurs, Oracle, Carte de détectabilité, Sécurité.*
