UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

ÉCOLE DOCTORALE MATHEMATIQUES ET INFORMATIQUE

ÉCOLE SUPERIEURE POLYTECHNIQUE

Année : 2017 N° d'ordre : 107

THESE DE DOCTORAT UNIQUE

Présentée pour obtenir le grade de Docteur de l'université Cheikh Anta DIOP de Dakar

Mention : Informatique et Télécommunications

Spécialité : Télécommunications

par

Birahime DIOUF

Titre : Sécurité multimédia : proposition de techniques de dissimulation d'informations numériques basées sur les codes polaires.

Soutenue le 04/11/2017 devant le jury composé de :

Président	Dorothé AZILINON	Professeur	UCAD
Rapporteur	Morgan BARBIER	Professeur	ENSICAEN, France
Rapporteur	Ridha BOUALLEGUE	Professeur	SUPCOM, Tunisie
Examinateur	Mamadou Lamine NDIAYE	Maître de conférences	UCAD
Examinateur	Fabé Idrissa BARRO	Maître de conférences	UCAD
Directeur de thèse	Sidi Mohamed FARSSI	Professeur	UCAD
Co-directeur de thèse	Khaly TALL	Maître de conférences	UCAD

REMERCIEMENTS

Après avoir rendu grâce à Dieu et à son Prophète Mohamed (PSL), je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'aboutissement de cette thèse.

Je remercie tous d'abord mon directeur de thèse Pr. Sidi Mohamed FARSSI de m'avoir donné l'opportunité de réaliser cette thèse et pour la confiance qu'il m'a accordée, ainsi que pour l'aide qu'il m'a apportée dans la réalisation de cette thèse. Je remercie également Pr. Khaly TALL, mon co-directeur, pour le soutien et l'appui pour la réalisation de cette thèse. Je n'oublie pas de remercier Dr. Idy DIOP d'avoir encadré cette thèse et pour sa disponibilité et ses conseils. Vos avis et suggestions m'ont étaient d'un grand apport.

Je remercie tous les membres du jury pour l'intérêt qu'ils ont porté à ma thèse. Je tiens à remercier Pr. Dorothé AZILINON qui, a bien voulu me faire l'honneur de présider mon jury. Je remercie également Morgan BARBIER, Professeur à l'Université de Caen Basse-Normandie et ENSICAEN et Ridha BOUALLEGUE, Professeur à SUPCOM Tunis d'avoir accepté de rapporter ma thèse. Vos remarques pertinentes et constructives ont contribué à améliorer la qualité de ce manuscrit. Mes remerciements vont à l'endroit de Pr. Mamadou Lamine NDIAYE et Pr. Fabé Idrissa BARRO d'avoir accepté d'examiner cette thèse.

Le soutien familial a été un point important sans lequel je n'aurai certainement pas pu réaliser cette thèse. Je souhaite, de tout mon cœur, remercier mon défunt père Cheikh DIOUF, Puisse Dieu, le Tout Puissant, l'avoir en sa sainte miséricorde ; ma maman Amy THIAW, longue vie à elle, que ne pourrais jamais remercier assez, si j'ai pu réussir dans mes études, c'est grâce à ton soutien et assistance morale et ta patience durant toutes ces longues années. Je remercie mes frères (Yéli, Djiby, Mouhamet, Khadim) et mes sœurs (Amy et Sokhna). Je remercie toute la famille DIOUF, THIAW et FALL (oncles, tantes, cousins et cousines) pour leur soutien indéfectible en mon endroit.

Enfin, je remercie tous mes amis et collègues du Laboratoire LIMBI et LIRT plus particulièrement à O. KHOUMA, M. DIOUF, L. SANE, L. MBOUP, I. GAYE, O. SADIO, I. A. R. Ndiaye, P. Ndiaye, Ndiaye DIOP, H. B. DIOUF, M. BA, … dont la gentillesse et la bonne humeur quotidienne contribuent à l'ambiance de travail agréable. Je remercie aussi mes amis I. DIONE, A. K. DIOP, M. DEME, M. DIOP, M. KANE, … pour les bons moments qu'on passe ensemble, ainsi que les enseignants et tout le personnel du DGI/ESP.

Je remercie toutes les personnes qui me sont chères, que j'ai oubliées et qui se reconnaitront !

DEDICACES

Je dédie ce travail
A la mémoire de mon cher et regretté père,
A ma chère mère,
En témoignage de ma profonde affection et mon amour indéfectible,
Qu'ils trouvent ici une récompense aux sacrifices déployés à mon égard.
A tous les membres de ma famille,
A tous mes amis,
A tous mes enseignants de l'école primaire à l'université.

LISTE DES PUBLICATIONS

✦ Articles de revues internationales :

[1] **B. Diouf**, I. Diop, S. M. Farssi, ***"Performances of Polar Codes in Steganographic Embedding Impact Minimization,"*** *In Proceedings of the Advanced Communication Technology (ICACT),* and CACT Transactions on Advanced Communications Technology (ICACT-TACT), vol. 5, no. 5, pp. 927–935, South Korea, September 2016. [Online] http://www.icact.org/journal.asp, or http://ieeexplore.ieee.org/document/7890246/.

[2] **B. Diouf**, I. Diop, S. M. Farssi et M. Chaumont, ***«Minimisation de l'impact d'insertion en stéganographie avec les codes polaires »***, 6ème édition du colloque COmpression et REprésentation des Signaux Audiovisuels (CORESA'2013), pp. 153-159, Le Creusot, France, 28 et 29 novembre 2013, [Online] https://liris.cnrs.fr/coresa/articles/2013/papers/27_coresa2013_submission_17.pdf.

[3] **B. Diouf**, I. Diop, S. M. Farssi, ***"Near Optimal Embedding Steganography in Spatial Domain using Successive Cancellation List Decoding of Polar Codes,"*** Submitted to IEEE Transaction on Information Forensic and Security.

[4] **B. Diouf**, I. Diop, K. Wone, M. Diouf, S. M. Farssi, K. Tall, O. Khouma, ***"Polar Coding Steganographic Embedding Using Successive Cancellation,"*** Accepted and presented in CNRIA Workshop 2017, [Online] http://interdisciplinarysolutions.org/2017/show/accepted-papers To be appear in Springer.

✦ Conférences internationales :

[5] I. Diop, **B. Diouf**, S. M. Farssi, K. Tall, P. A. Fall, A. K. Diop, K. Sylla, ***"Using of Polar Codes in Steganography,"*** *In Proceedings of the 2nd International Conference on Advances in Computer Science and Engineering (CSE 2013),* vol. 42, pp. 262-266, Atlantis Press, Los Angeles, USA, July 1-2, 2013. [Online] http://www.atlantis-press.com/php/download_paper.php?id=6915, indexing and abstracting Atlantis Press.

[6] **B. Diouf**, I. Diop, S. M. Farssi and O. Khouma, ***"Minimizing Embedding Impact in Steganography Using Polar Codes,"*** *In Proceedings of 4th IEEE International Conference on Multimedia and Computing Systems (ICMCS'14),* pp. 105-111, Marrakesh, Morocco, April 14-16, 2014. [Online] http://ieeexplore.ieee.org/document/6911224/, indexing and abstracting IEEE.

[7] **B. Diouf**, I. Diop, S. M. Farssi and O. Khouma, ***"Practical Polar Coding Method to Minimize the Embedding Impact in Steganography,"*** In *Proceedings of the IEEE International Science and Information (SAI) Conference,* pp. 1230-1235, London, United

Kingdom, 28-30 July, 2015. [Online]: http://ieeexplore.ieee.org/document/7237301/, indexing and abstracting IEEE.

[8] **B. Diouf**, I. Diop, K. Wone, S. M. Farssi, O. Khouma, M. Diouf, K. Tall, ***"Adaptive Linear Programming of Polar Codes to Minimize Additive Distortion in Steganography,"*** In *Proceedings of the IEEE International Science and Information (SAI) Conference*, pp. 1086-1092, London, United Kingdom, 13-15 July, 2016. [Online]: http://ieeexplore.ieee.org/document/7556113/, indexing and abstracting IEEE.

[9] M. Diouf, I. Diop, I. Dioum, M. Farssi, **B. Diouf**, K. Tall, ***"Study of Polar Codes MIMO Channels Polarization using MIMO System,"*** *IEEE International Science and Information Conference*, pp. 681-690, London, United Kingdom, 13-15 July, 2016. [Online]: http://ieeexplore.ieee.org/document/7556056/, indexing and abstracting IEEE.

[10] M. Diouf, I. Diop, I. Dioum, M. Farssi, **B. Diouf**, K. Tall, ***"Soft Output Detection for MIMO Systems using Binary Polar Codes,"*** *In Proceedings of 6th IEEE International Conference on Multimedia and Computing Systems (ICMCS'16)*, pp. 400-404 Marrakesh, Morocco, 29 September-1 October, 2016. http://ieeexplore.ieee.org/document/7905658/, indexing and abstracting IEEE.

[11] M. Diouf, I. Diop, I. Dioum, **B. Diouf**, K. Tall, M. Farssi, L. Sané, ***"Enhanced MIMO Systems performances by concatenating small Polar Coding to Spatial Time Block Codes,"*** Accepted and presented in CNRIA Workshop 2017, [Online] http://interdisciplinarysolutions.org/2017/show/accepted-papers, To be appear in Springer.

[12] O. Khouma, M. L. Ndiaye, I. Diop, S. M. Farssi, A. K. Diop, **B. Diouf**, J. J. Montois, ***"Classification model of spikes morphology using principal components analysis in drug-resistant epilepsy,"*** Accepted and presented in CNRIA Workshop 2017, [Online] http://interdisciplinarysolutions.org/2017/show/accepted-papers, To be appear in Springer.

[13] O. Khouma, M. L. Ndiaye, I. Diop, S. Diaw, A. K. Diop, S. M. Farssi, **B. Diouf**, K. Tall, J. J. Montois, ***"New Clustering of the Spikes Morphology Based on Dynamic Clouds in Partial Epilepsy,"*** *International Science and Information Conference*, pp. 354-360, London, United Kingdom, 13-15 July, 2016. [Online]: http://ieeexplore.ieee.org/document/7556006/, indexing and abstracting IEEE.

[14] O Khouma, M. L. Ndiaye, S. M. Farssi, J. J. Montois, I. Diop, **B. Diouf**, ***"A Comparative Methods of Spike Detection in Epilepsy,"*** *IEEE International Science and Information Conference*, pp. 749-745, London, United Kingdom, 28-30 July, 2015. [Online]: http://ieeexplore.ieee.org/document/7237226/, indexing and abstracting IEEE.

TABLE DES MATIERES

TABLE DES FIGURES

LISTE DES TABLEAUX

GLOSSAIRE

ALP (Adaptive Linear Programming) : la programmation linéaire adaptative.

BCH (codes) **:** les codes indépendamment découverts par Bose et Chaudhuri (BCH binaires) d'une part et Hocquenghem (BCH non binaires) d'autre part.

B-DMC (Binary-input Discrete Memoryless Channel) : canal discret sans mémoire à entrée binaire.

BEC (Binary Erasure Channel) : canal d'effacement binaire.

BOSS (Break Our Stego-System) : compétition pour les attaques des schémas de stéganographie.

BP (Belief Propagation) : décodage de propagation de croyance.

bpp (bit per pixel) : ou bit par pixel est le taux d'insertion.

BSC (Binary Symmetric Channel) : canal binaire symétrique.

CRC (Cyclic Redundancy Check) : technique de detection d'erreurs de transmission.

DCT (Discret Cosinus Transform) : représentation pour les images JPEG.

FLD (Fisher Linear Discriminant) : classifieur binaire utilisé en stéganalyse.

GRS (Generalized Reed-Solomon codes) : codes de Reed-Solomon Généralisés.

HILL (HIgh Low Low) : méthode de calcul des coûts de modification.

HUGO (Highly Undetectable steGanOgraphy) : schéma de stéganographie adaptatif.

JPEG (Joint Photographic Expert Group) : norme de compression d'images.

LDPC (Low-Density Parity-Check) : code à faible densité.

LLR (Log-Likelihood Ratio) : rapport de vraisemblance logarithmique.

LP (Linear Programming) : la programmation linéaire.

LLR (Log-Likelihood Ratio) : rapport de vraisemblance logarithme.

LR (Likelihood Ratio) : rapport de vraisemblance

LSB (Least Significant Bit) : bit de poids faible dans la représentation binaire d'un octet.

MAP (Maximum a Posteriori Probability) ou **ML** (Maximum Likelihood) : décodage optimal des codes correcteurs d'erreurs.

MSE (Mean Square Error) : erreur quadratique moyenne.

MVGG (Multivariate Generalized Gaussian) : méthode de calcul des coûts de modification.

PC (Polar Code) : code polaire.

PCS (Polar Coding Steganography) : premier schéma de stéganographie basé code polaire.

PNG (Portable Network Graphics) : format d'image en niveaux de gris.

PSNR (Peak Signal to Noise Ratio) : rapport de signal crête sur bruit.

RGB (Red Green Blue) : type d'image en couleur.

RS (Reed-Solomon) : les codes découverts par Reed et Solomon.

SRM (Spatial Rich Model) : modèle de caractéristique dans le domaine spatial.

SC (Successive Cancellation) : décodage d'annulation successive des codes polaires.

SCAN (Soft CANcellation) : décodage à décisions souples.

SCL (SC List) : décodage d'annulation successive utilisant une liste.

SPAM (Subtractive Pixel Adjacency Matrix): espace de caractéristiques, à partir duquel, sont vecteurs de caractéristiques des pixels voisins pour le calcul des coûts de modification.

STC (Syndrome Trellis Codes) : codes de syndrome en treillis.

SVM (Support Vector Machine) : classifieur binaire utilisé en stéganalyse.

UNIWARD (UNIversal WAvelet Relative Distortion) : méthode de calcul des coûts de modification.

WOW (Wavelet Obtained Weights) : méthode de calcul des coûts de modification.

RESUME

La sécurité de l'information est devenue un problème central pour toute personne. Une réponse à cette sécurité garantissant la confidentialité dans la communication est offerte par la stéganographie. Elle consiste à dissimuler une information secrète dans un médium, ayant une apparence anodine, de manière indétectable. Les outils de la théorie de l'information, généralement utilisés en transmission numérique, sont exploités pour améliorer la sécurité des schémas de stéganographie modernes. Nous utilisons, notamment, les codes polaires qui représentent, actuellement, la première et seule famille de codes permettant d'atteindre les deux limites établies par Shannon sur le codage canal et source.

Nos propositions dans cette thèse s'articulent autour de deux axes. Le premier regroupe le premier schéma de stéganographie basé sur les codes polaires ainsi que des améliorations de celui-ci pour la minimisation du nombre d'éléments de couverture modifiés (profil constant) avec la possibilité de verrouiller les éléments les plus sensibles (papier mouillé). Le deuxième axe concerne les méthodes de stéganographie adaptatives basées sur les codes polaires. Nous en avons proposé deux schémas adaptatifs. Le premier utilise le décodage par programmation linéaire adaptative ALP et permet de minimiser la fonction de distorsion établie à partir des coûts de modifications des éléments de couverture. Nous avons montré que ce schéma offre une efficacité d'insertion supérieure à celle du STC. Nous avons proposé un deuxième schéma adaptatif basé sur le décodage d'annulation successive en liste SCL améliorant le précédent en termes d'efficacité d'insertion en garantissant le succès de l'insertion. L'optimalité de ce schéma est contrôlée par le paramètre représentant la taille de la liste. Nous avons également montré que ce schéma minimise la fonction de distorsion.

Mots Clés : annulation successive, codes à papier mouillé, codes correcteurs d'erreurs, codes polaires, décodage en liste, distorsion, matrix embedding, programmation linéaire, stéganographie, sécurité.

ABSTRACT

The information security has become a central problem for anyone. A solution to this security that assures the confidentiality in communication is provided by steganography. It consists to conceal secret information in a medium, with unsuspected appearance, such as it was undetectable. The tools of information theory, commonly used in digital transmission, are used to improve the security of modern steganography systems. We use, more particularly, the polar codes which are currently the first and only family of codes that achieve the two bounds established by Shannon in channel and source coding.

Our proposals in this thesis revolve around two points. The first point includes, on the one hand the first steganography scheme based on polar codes and, on the other hand the improvements for minimizing the number of modified cover elements (constant profile) with the possibility to lock the most sensitive ones (wet paper). The second point concerns the adaptive steganographic methods based on polar codes. We have proposed two adaptive schemes. The first uses the adaptive linear programming decoding and allows minimizing the distortion function established from embedding cost changes of the cover elements. We have shown that this scheme offers better embedding efficiency than STC. We proposed a second adaptive scheme based on the successive cancellation list decoding SCL improving the previous in terms of embedding efficiency ensuring the success of the embedding. The optimality of this scheme is controlled by the list size parameter. We have also shown that this scheme minimizes the distortion function.

Keywords: adaptive, distortion, errors correcting code, linear programming, list decoding, matrix embedding, polar codes, security, steganography, successive cancellation, wet paper

INTRODUCTION GENERALE

« 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)

Aujourd'hui, avec le développement de l'internet qui permet un partage de données multimédias (voix, textes, images ou vidéos) à longueur de journée, le contrôle de l'information est devenu un enjeu économique et sécuritaire pour les Etats, les entreprises ou même les particuliers. Les techniques de sécurisation d'informations telles que la cryptographie, protègent les informations mais ne garantissent pas la discrétion. Cependant, souvent même l'existence de la communication doit être secrète. Dans ces conditions, nous pouvons recourir à la *stéganographie* qui consiste à dissimuler une information secrète dans des médiums de couverture insoupçonnés de telle sorte que seul le destinataire soit au courant de l'existence de la communication. L'objectif principal est de communiquer sans éveiller le moindre soupçon. L'image étant un support de prédilection des nouvelles technologies de communication, c'est le type de médium que nous utilisons dans ce manuscrit. La stéganographie peut être utilisée à des fins malveillantes. Dans ce cas, on utilise la stéganalyse pour détecter la présence du message dissimulé. Les modèles de stéganalyse actuels sont définis en utilisant les outils d'apprentissage et de classification. Le stéganalyste utilise une base d'images à partir de laquelle un ensemble de caractéristiques statistiques sont extraites et seront ensuite réparties en deux sous-ensembles d'apprentissage et de test. Le classifieur détermine la limite de décision sur un ensemble d'apprentissage et applique ensuite cette limite à l'ensemble de test.

Dans la stéganographie moderne ou numérique, qui a véritablement débuté vers 1996 [1], la majorité des méthodes actuelles sont basées sur la minimisation de l'impact d'insertion du message. Dans ce cas, l'insertion doit se faire de sorte que les modifications soient le moins perceptibles possibles. Dans le domaine spatial, les bits du message sont insérés au niveau des pixels de l'image de couverture (a) : en faisant le moins de modifications possibles et (b) : en les portant dans les endroits les moins sensibles de l'image. Pour répondre au premier problème (a), Crandall a introduit la technique de matrix embedding [2]. Une relation entre les codes et le problème de minimisation du nombre de pixels modifiés (profil constant) est établie par Bierbrauer [3]. La première implémentation de la technique de matrix embedding a vu le jour avec l'algorithme F5 de Westfeld [4] dans lequel les codes Hamming sont utilisés. Par la suite,

d'autres schémas ont implémenté les codes Golay [5], BCH (Bose-Chaudhuri-Hocquenghem), [6], [7], RS (Reed-Solomon) [8], LDGM (Low Density Generator Matrice) [9] avec la construction ZZW (Zhang-Zhang-Wang) [10], LDPC (Low Density Parity Check) [11], STC (Syndrome Trellis Codes) [12], etc. Le deuxième problème (b) est réglé par la stéganographie adaptative au contenu qui dissimule le message secret dans les zones les plus difficilement détectables de l'image comme les zones texturées ou les zones bruitées. Le principe est composé de deux tâches complémentaires. La première consiste à définir efficacement les coûts de modification des pixels de l'image. Plusieurs méthodes sont proposées comme HUGO (Highly Undetectable steGO) [13], ASO (Adaptive Steganography by Oracle)[14], WOW (Wavelet Obtained Weights) [15], UNIWARD (UNIversal WAvelet Relative Distortion) [16], HILL (HIgh Low Low) [17] et MiPOD (Minimizing the Power of Optimal Detector) [18]. La seconde étape repose sur la minimisation de la fonction de distorsion définie à partir des coûts de modification. Cela se fait via une technique de codage pratique. Pour y parvenir, les codes STC [19] et LDPC [20] sont exploités pour une insertion adaptative. Les codes à papier mouillé permettent de verrouiller les pixels interdits de modification lors de l'insertion. La plupart des méthodes proposées se focalisent sur l'une ou l'autre des deux tâches et utilisent une méthode existante pour l'autre tâche. Dans cette thèse, nous nous sommes intéressés à la deuxième tâche et utilisons les *codes polaires* (*polar codes*, en anglais) pour minimiser la fonction de distorsion et approcher d'avantage la limite théorique correspondant à l'insertion optimale.

Claude Shannon a posé les bases de la théorie de l'information en établissant deux théorèmes fondamentaux ; le premier sur le codage canal [21] et le second sur le codage source [22]. Pour le codage canal, il a montré qu'il existe des séquences de codes permettant d'atteindre la capacité du canal de transmission. Depuis, plusieurs codes sont proposés dans l'état de l'art comme LDPC [23] et Turbo codes [24] qui ont permis d'approcher cette limite. C'est dans ce contexte qu'Arikan a introduit, en 2009, les codes polaires [25], comme étant les premiers codes permettant d'atteindre la capacité du canal de transmission. Les codes polaires sont caractérisés par une construction explicite basée sur un phénomène appelé *polarisation de canal* et des algorithmes d'encodage et de décodage de faible complexité $O(n\,logn)$. Le second théorème de Shannon caractérise le taux minimal nécessaire pour atteindre une distorsion donnée et établit une limite de taux-distorsion pour le codage source. Les codes polaires sont également démontrés comme étant les premiers codes qui atteignent cette limite; ils sont particulièrement optimaux pour les problèmes de Slepian-Wolf, de Wyner-Ziv et de

Gelfand-Pinsker [26]. En outre, les codes polaires ont une construction explicite et une structure récursive qui les rendent particulièrement adaptés à une implémentation pratique et efficace. Plusieurs types de décodage des codes polaires sont proposés telles que le SC (Successive Cancellation) [25], le SCL (SC-List) [27], le BP (Belief Propagation) [28], le LP (Linear Programming) [29], l'ALP (Adaptive LP) [30] et le SCAN (Soft CANcellation) [31].

Les remarquables propriétés des codes polaires évoquées plus haut et leur optimalité au problème d'insertion stéganographique soulignée par Filler et al. [19], motivent l'intérêt que nous portons à ce sujet. Nous nous sommes fixés comme objectifs de proposer les codes polaires pour remplacer les autres codes de l'état de l'art aussi bien pour la minimisation du nombre de modifications du médium de couverture, pour la technique du papier mouillé que pour une insertion adaptative. Dans le cadre de cette thèse, nous utiliserons les images numériques et l'insertion se fera dans le domaine spatial. Cette thèse est organisée comme suit.

- Le chapitre 1 décrit la théorie des codes correcteurs d'erreurs et les différents concepts nécessaires pour la compréhension des schémas de stéganographie.
- Nous parlerons, dans le chapitre 2, de la stéganographie, de la stéganalyse, des modèles d'images existants et des différents schémas proposés en utilisant les codes.
- Dans le chapitre 3, nous traiterons du nouveau paradigme de codage canal, de la construction, de l'encodage et du décodage des codes polaires.
- Nous avons prouvé, dans le chapitre 4, l'applicabilité des codes polaires en stéganographie en proposant un premier schéma basé, d'une part sur la minimisation du nombre de pixels modifiés, et d'autre part sur le verrouillage des pixels très sensibles à la modification (papier mouillé). Afin de réduire la complexité, nous avons proposé une autre méthodologie qui exploite la structure particulière des codes polaires. Des tests sont effectués sur des images numériques dont les pixels sont permutés avant insertion du message secret.
- Dans le dernier chapitre (chapitre 5), nous proposons deux schémas adaptatifs. Le premier exploite les ressemblances entre le problème d'insertion stéganographique et le décodage ALP ainsi que sa propriété ML (Maximum Likelihood) pour minimiser la fonction de distorsion additive convenablement définie. Le deuxième schéma adaptatif, basé sur le décodage SCL qui offre aux codes polaires des taux d'erreurs de décodage plus faibles que ceux des codes LDPC et Turbo-codes, améliore le précédent en termes d'efficacité.

Chapitre 1: LES CODES CORRECTEURS D'ERREURS

Introduction

Aujourd'hui on trouve des codes correcteurs d'erreurs dans la quasi-totalité des systèmes de communication numériques comme le stockage numérique (CD, DVD), la télévision numérique (DVB), les réseaux téléphoniques mobiles, les réseaux sans fil WiMax, les communications par satellite, … Les codes correcteurs d'erreurs sont utilisés pour corriger les erreurs survenues lors de la transmission des données et sont répartis en deux grandes familles : les codes en bloc et les codes convolutifs. Nous allons commencer par donner quelques concepts de base sur la théorie des codes, puis parler de quelques exemples de codes en bloc linéaires binaires (codes de Hamming, BCH, LDPC, …) et non binaires (codes RS) et de leur décodage et enfin, terminer par la présentation des codes convolutifs.

1.1 Définitions des concepts de base

Avant de présenter des exemples de codes correcteurs d'erreurs, nous rappelons la définition de quelques notions de base sur les codes.

1.1.1 Caractéristiques d'un code correcteur d'erreurs

Lors de la transmission de l'information numérique sur un canal, il est possible qu'il y'ait des perturbations qui viennent modifier les symboles qui constituent l'information. Le codage canal est la technique destinée à la correction des erreurs survenues lors de la transmission. Les codes utilisés à cet effet sont dits codes correcteurs d'erreurs.

Un bon code correcteur d'erreurs est un code qui répond aux critères suivants [32]:

- on doit pouvoir détecter et corriger un nombre raisonnable d'erreurs ;
- l'algorithme d'encodage doit être suffisamment rapide (faible complexité) ;
- l'algorithme de décodage (corrections incluses) doit être suffisamment rapide ;
- il doit offrir une capacité suffisamment grande (proche de la capacité du canal).

D'autres caractéristiques des codes, non moins importantes, sont à considérer. Les codes sont caractérisés par la distance minimale, une matrice génératrice qui intervient lors du codage et une matrice de contrôle de parité qui est utilisée au décodage.

1.1.2 Distance minimale d'un code correcteur d'erreurs

Avant de définir la distance minimale il est important de donner deux définitions [33].

Définitions 1.1 :

Soient c et c' deux mots d'un code donné :

- Le *poids de Hamming* d'un mot de code c représente le nombre de symboles non nuls de c. On le note par w_H.

Exemple 1.1 :

Soit un mot de code binaire $\mathrm{c} = \mathbf{1}0\mathbf{1}\mathbf{1}0\mathbf{1}00$ alors son poids est donné par $w_H(\mathrm{c}) = 4$.

- La *distance de Hamming* entre deux mots de code c et c', notée par d_H, est définie comme étant le nombre de positions où on note une différence entre les éléments de c et c'.

Remarque 1.1 :

La distance entre deux mots de code c et c' est aussi égale au poids de leur somme symbole par symbole. Cela provient du fait que la somme de deux symboles est nulle dans le cas où ils sont identiques et non nulle dans le cas contraire.

Exemple 1.2 :

Soient $\mathrm{c} = \mathbf{0}0\mathbf{1}011\mathbf{0}0$ et $\mathrm{c}' = \mathbf{1}0\mathbf{0}0111\mathbf{1}0$ alors $d_H(\mathrm{c}, \mathrm{c}') = 3$.

$\mathrm{c}'' = \mathrm{c} + \mathrm{c}' = \mathbf{1}0\mathbf{1}000\mathbf{1}0$, $w_H(\mathrm{c}'') = 3$. Nous avons bien $d_H(\mathrm{c}, \mathrm{c}') = w_H(\mathrm{c} + \mathrm{c}') = 3$.

- La *distance minimale* d_{min} d'un code est égale à la plus petite distance de Hamming entre ses mots de code pris deux à deux.

On note par $C(n, k, d_{min})$ ou par $C(n, k)$ le code en bloc linéaire C de longueur n, de dimension k et de distance minimale d_{min}. Le vecteur $(a_i, \dots, a_j)$ est noté par a_i^j.

1.1.3 Matrice génératrice

Une matrice génératrice G d'un code $C(n, k)$ est une matrice de k lignes et n colonnes à éléments dans $\mathbb{F}_q$ (corps de Galois à $q = 2^p$ éléments), dont les lignes constituent une base de C. Elle permet d'associer à un bloc de données $\mathrm{u} = u_1^k$ un mot de code $\mathrm{c} = c_1^n$ par la relation $c_1^n = u_1^k \mathrm{G}$. Un code $C(n, k)$ admet plusieurs matrices génératrices. En effet, en permutant les lignes ou les colonnes d'une matrice génératrice ou encore en ajoutant, à une ligne, une ou plusieurs autres lignes, on obtient une autre matrice génératrice donnant le même ensemble de mots de code mais avec une association différente entre mots de code et mots d'information.

1.1.4 Matrice de contrôle de parité

A chaque code en bloc linéaire $C(n,k)$ de matrice génératrice G, on peut associer un code en bloc linéaire dual $C(n,n-k)$, qui vérifie que tout mot du code dual est orthogonal[1] à tout mot du code $C(n,k)$. Le dual du code $C(n,k)$ est donc un sous-espace vectoriel de $\mathbb{F}_q^n$ constitué de 2^{n-k} mots de code de n symboles [32]. Ce sous-espace vectoriel est l'orthogonal du sous-espace vectoriel constitué des 2^k mots du code $C(n,k)$. Il en résulte que tout mot c du code $C(n,k)$ est orthogonal aux lignes de la matrice génératrice H de son code dual $C(n,n-k)$ $\mathrm{cH^T} = 0$ et $\mathrm{G \cdot H^T} = 0$, T désigne la transposition. Ceci conduit à une propriété très importante :

- *Un mot $a = a_1^n$ donné est un mot de code si et seulement si*

$$a\mathrm{H^T} = 0 \tag{1.1}$$

Au décodage, nous pouvons utiliser cette propriété pour vérifier si le mot reçu est un mot de code et détecter ainsi la présence d'erreurs. C'est pourquoi la matrice H est appelée la matrice de contrôle de parité du code $C(n,k)$.

Une matrice génératrice est sous forme systématique si elle s'écrit $\mathrm{G} = [\mathrm{I}_k\ \mathrm{P}]$ avec I_k la matrice identité $k \times k$ et P une matrice $k \times (n-k)$ utilisée pour calculer les $(n-k)$ symboles de redondance. La matrice de contrôle de parité est donnée par $\mathrm{H} = [\mathrm{P^T}\ \mathrm{I}_{n-k}]$.

1.1.5 Pouvoir de correction d'un code linéaire

On appelle *capacité de correction* (resp. *de détection*) d'un code C le plus grand entier cap_c (resp. cap_d) tel qu'on soit toujours capable de corriger cap_c (resp. détecter cap_d) erreurs. La théorie des codes indique qu'un code en bloc linéaire C permet de détecter $d_{min} - 1$ erreurs et de corriger $\lfloor (d_{min} - 1)/2 \rfloor$ erreurs. Ainsi

$$cap_d = d_{min} - 1 \tag{1.2}$$

$$cap_c = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor \tag{1.3}$$

où $\lfloor q \rfloor$ désigne la partie entière de q.

[1] Deux vecteurs x_1^n et y_1^n sont dits orthogonaux si leur produit scalaire est nul ($\sum_{j=1}^{n} x_j y_j = 0$).

Définitions 1.2 :

La borne de Singleton pour un code en bloc linéaire est définie par :

$$d_{min} \leq n - k + 1. \tag{1.4}$$

Les codes qui sont tels que la borne de Singleton est atteinte (égalité dans (1.4)) sont dits MDS (Maximum Distance Separable).

1.2 Codes en bloc linéaires à symboles binaires

La famille des codes en bloc (encore appelés codes algébriques) linéaires est vaste et la plupart de ces codes sont binaires. Nous n'en donnerons que quelques exemples qui sont utilisés en stéganographie.

1.2.1 Les codes cycliques

Les codes cycliques représentent la classe la plus importante des codes en bloc linéaires.

Définitions 1.3 :

On appelle code cyclique de longueur n tout $\mathbb{F}_2$ −sous-espace vectoriel C de $\mathbb{F}_2^n$ stable par décalage circulaire [33]. Soit $C(n,k)$ un code sur un corps fini $\mathbb{F}_2$, $C(n,k)$ est dit cyclique si :

i) $C(n,k)$ est un code linéaire et

ii) $(c_0, c_1, \dots c_{n-2}, c_{n-1})$ est un mot de code entraine que $(c_{n-1}, c_0, c_1, \dots, c_{n-2})$ l'est aussi.

Les mots de code c et les mots d'information u sont souvent représentés par des polynômes. Ainsi, nous leur associons respectivement à $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ et $u(x) = u_0 + u_1 x + \cdots + u_{k-1}x^{k-1}$, où c_i et u_i prennent leurs valeurs dans $\mathbb{F}_2$.

Le diviseur unitaire $g(x)$ de $x^n + 1$, dans $\mathbb{F}_2[x]$ (ensemble des polynômes à coefficients dans $\mathbb{F}_2$) associé à C est appelé son polynôme générateur. Cela signifie que, pour définir le code C, il suffit de déterminer $g(x)$; ce qui revient à chercher ses racines. En trouvant les racines de $x^n + 1$, nous parviendrons ainsi à déterminer les celles de $g(x)$ car les racines de $g(x)$ sont aussi racines de $x^n + 1$. Le polynôme $x^n + 1$ a pour racine 1 dans $\mathbb{F}_2$. Si l'on veut qu'il ait d'autres racines, il faudra "imaginer" de nouveaux éléments qui ne sont pas dans $\mathbb{F}_2$, de la même façon qu'on a construit le corps des nombres complexes en « imaginant » un nouveau nombre i tel que $i^2 = -1$. Par conséquent nous considérons une extension de $\mathbb{F}_2$ qu'on appellera *corps de décomposition* de $x^n + 1$. Les racines de $g(x)$ sont appelées les zéros du code cyclique C. Donc le code cyclique peut être défini à partir de ses zéros.

Définitions 1.4:

- On appelle α une ***racine primitive $n-$ ème de l'unité*** dans une extension de $\mathbb{F}_2$ si :
  - ✓ α est une racine primitive c'est-à-dire ses puissances $1, \alpha, \cdots, \alpha^{n-1}$ sont des racines et sont toutes différentes ;
  - ✓ $\alpha^n = 1$ (n est dit ordre de α).
- Pour tout entier j, avec $0 \leq j \leq n$, on note par $cl(j)$ la ***classe cyclotomique*** de n une partie de $\{1, \cdots, n\}$ stable par multiplication par 2 et minimale par inclusion ; donc définie par :

$$cl(j) = \{j, 2j, \cdots, 2^{p-1}j \ mod \ n\} \quad \text{avec} \quad 2^p j \ mod \ n = j \ mod \ n \tag{1.5}$$

où p est le plus petit entier tel que n divise $2^p - 1$ ($\alpha \in \mathbb{F}_{2^p}$).

Précisons qu'ici l'égalité $n = 2^p - 1$ n'est pas forcément vérifiée. Ce cas de figure ne se présente que si $n + 1$ est une puissance de 2 (cas des codes de Hamming). Dans le cas contraire, on se ramène au plus petit entier $n' > n$ tel que $n' = 2^p - 1$.

- *Le **polynôme minimal*** de α^s (polynôme de degré minimal annulant α^s) sur $\mathbb{F}_2$ est :

$$m_{\alpha^s}(x) = \prod_{i \in cl(s)} (x - \alpha^i), \tag{1.6}$$

où les α^i, avec $i \in cl(s)$, sont appelés conjugués de α^s.

Le polynôme générateur $g(x)$ d'un code cyclique C possède les propriétés suivantes [32]:

a) $g(x)$ est l'unique polynôme unitaire de plus petit degré sur $\mathbb{F}_2$ engendrant C :

$$C = \{q(x)g(x) \ mod \ (x^n + 1) \mid q(x) \in \mathbb{F}_2[x]\}\,; \tag{1.7}$$

$g(x)$ divise $x^n + 1 \Leftrightarrow$ il existe donc un polynôme $h(x)$ dit de contrôle tel que :

$$g(x)h(x) = x^n + 1 \tag{1.8}$$

b) Soit $r = \deg(g)$, et soit $g(x) = \sum_{i=0}^{r} g_i x^i$ où $g_r = 1$. Alors la dimension de C est $k = n - r$; de plus, les polynômes $g(x), xg(x), \cdots, x^{k-1}g(x)$ forment une base de C. La matrice correspondante est donnée par :

$$\mathrm{G} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \ldots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}. \tag{1.9}$$

C'est une matrice génératrice du code C.

c) Soit α une racine primitive $n - ième$ de l'unité dans une extension de $\mathbb{F}_2$ et $m_{\alpha^s}(x)$ le polynôme minimal associé à α^s sur $\mathbb{F}_2$, alors

$$g(x) = \prod_{s \in J} m_{\alpha^s}(x) \implies (g(\alpha^s) = 0)_{s \in J}, \tag{1.10}$$

où J est un sous-ensemble de représentants des classes cyclotomiques modulo n.

1.2.2 Codes de Hamming

Les codes de Hamming sont définis par $C(n,k) = C(2^p - 1, 2^p - p - 1)$, avec p un entier strictement positif. La matrice de contrôle de parité d'un tel code est obtenue en portant, à ses colonnes, la représentation binaire des nombres de 1 à n. Chaque colonne est constituée de $p = n - k$ symboles binaires. Ainsi H est formée de p lignes et $2^p - 1 = n$ colonnes.

Propriété 1.1 :

La distance minimale d'un code de Hamming est toujours égale à 3.

Ainsi la capacité de correction $cap_c = t = 1$ et la capacité de détection $cap_d = 2$.

Donc si l'erreur est de poids supérieur à 1, le code ne permet pas la correction.

Exemple 1.3 :

Soit un code de Hamming de paramètre $p = 3$. Les mots de code et les blocs de données sont alors respectivement constitués de $n = 7$ et $k = 4$ symboles binaires. Une matrice de contrôle de parité est la suivante :

$$\mathrm{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} = [\mathrm{P}^{\mathrm{T}} \ \ \mathrm{I}_3] \tag{1.11}$$

et la matrice génératrice correspondante est égale à :

$$\mathrm{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [\mathrm{I}_4 \ \ \mathrm{P}] \tag{1.12}$$

1.2.3 Les codes BCH binaires

Les codes BCH sont des codes cycliques. Ils sont indépendamment découverts par Bose et Chaudhuri (BCH binaires) d'une part et Hocquenghem (BCH non binaires) d'autre part. On construit un code BCH à partir du paramètre $\delta = 2t + 1$ ou $\delta = 2t + 2$ (la distance construite du code). La distance minimale d_{min} d'un code BCH est toujours supérieure ou égale à sa distance construite. Ce code permet donc de corriger au moins t erreurs. Les codes BCH

binaires sont définis dans un corps de Galois $\mathbb{F}_2$ alors que les racines sont définies dans $\mathbb{F}_{2^p}$. Les codes de Hamming constituent un cas particulier ($t = 1$) des codes BCH.

Un code BCH est dit ***primitif*** (resp. ***non-primitif***) quand les racines de son polynôme générateur sont des puissances d'un élément primitif (resp. non-primitif) α de $\mathbb{F}_q$.

Le polynôme générateur $g(x)$ admet $\delta - 1$ racines: α^{l+i} avec $0 \leq i \leq \delta - 2$ et $l = 1$ ou $l = 0$.

- ✓ Pour $l = 1$ le polynôme générateur est donné par [32]:

$$g(x) = ppmc\big(m_\alpha(x), m_{\alpha^3}(x), \cdots, m_{\alpha^{2t-1}}(x)\big). \tag{1.13}$$

où $m_{\alpha^i}(x)$ est le polynôme minimal associé à α^i et $ppmc$ signifie plus petit multiple commun.

Les paramètres du code BCH primitif sont :

$$\begin{cases} n = 2^p - 1 \\ k \geq 2^p - 1 - pt \\ d_{min} = 2t + 1 \end{cases} \tag{1.14}$$

- ✓ Pour $l = 0$

La distance construite est $\delta = 2t + 2$ et le polynôme générateur s'écrit :

$$g(x) = ppmc\big(m_{\alpha^0}(x), m_{\alpha^1}(x), m_{\alpha^3}(x), \cdots, m_{\alpha^{2t-1}}(x)\big). \tag{1.15}$$

La procédure de détermination des polynômes minimaux est identique au cas $l = 1$. On peut remarquer que plus t augmente, plus le nombre de polynômes à déterminer augmente et plus la construction du code est complexe.

Les codes BCH sont utilisés dans des applications telles que les communications satellitaires.

1.2.4 Les codes LDPC

Les codes Low Density Parity Check (LDPC) ont été découverts par Gallager [23] en 1962 mais n'ont été développé dans la littérature qu'en 1999. Cela est dû au fait que ces codes demandaient une grande complexité calculatoire qui n'était pas disponible au moment de leur découverte mais qui est devenue possible grâce à l'avancée de la technologie en électronique. La matrice de contrôle de parité H de ces codes est de faible densité (creuse) ; elle contient plus de 0 que de 1. Elle peut être représentée soit sous forme matricielle ou sous forme d'un graphe bipartite appelé graphe de Tanner. La méthode de décodage proposée initialement repose sur un algorithme de Propagation de Croyances BP (Belief Propagation) qui est itératif. Pour faciliter

l'encodage, Richardson et Urbanke [34] ont proposé un prétraitement de la matrice de contrôle de parité avant l'opération d'encodage. L'objectif de ce prétraitement est de mettre la matrice H sous une forme presque triangulaire inférieure. Le décodage BP appliqué aux codes LDPC permet une communication fiable tout en approchant la limite de Shannon (la capacité du canal). En effet, plusieurs applications ont adopté les codes LDPC dans les systèmes de communications numériques tels que les réseaux sans fil LANs (IEEE 802.11n), WiMax (IEEE 802.16e), la télévision numérique par satellite DVB-S2 (Digital Video Broadcast-Satellite),…

1.3 Codes en bloc à symboles non binaires

1.3.1 Les codes de Reed-Solomon (RS)

Les codes de Reed-Solomon, désignés par l'acronyme RS, sont introduits par Irving Reed et Gustave Solomon en 1960 [35]. Ce sont des codes BCH non binaires, de longueur $n = 2^p - 1$ dans un corps de Galois $\mathbb{F}_{2^p}$ avec $p \geq 2$. En ajoutant t symboles de vérification, un code RS peut détecter jusqu'à $2t$ symboles erronés et peut corriger jusqu'à t symboles. Pour construire les codes de RS, deux approches sont proposées.

- ***Première approche pour construire des codes de Reed-Solomon Généralisés (GRS)***

Soit $u_0^{k-1} = (u_0, \cdots, u_{k-1})$ le bloc de k symboles d'information à transmettre, pris dans un corps de Galois $\mathbb{F}_q$. Soient $P(x) = u_0 + u_1 x + \cdots + u_{k-1}x^{k-1}$ sa représentation polynômiale.

✓ Un code GRS de longueur $n \leq q$ et de dimension k est une famille de mots correspondant à des polynômes de degré inférieur à k évalués sur une sous-famille de $\mathbb{F}_q$.

✓ Soit $\gamma = \{\gamma_0, \gamma_1, \cdots, \gamma_{n-1}\}$ une sous-famille de $\mathbb{F}_q$. On appelle $ev(P)$, l'application qui, à un polynôme P de degré strictement inférieur à k, à coefficients dans $\mathbb{F}_q$, associe un élément de $[\mathbb{F}_q]^n$ qui est l'évaluation de P en chacun des éléments de la famille γ [8].

$$ev(P) = \big(P(\gamma_0), P(\gamma_1), \cdots, P(\gamma_{n-1})\big). \tag{1.16}$$

✓ Le code de Reed-Solomon généralisé est défini par :

$$GRS(n,k) = \big\{ev(P) : P \in \mathbb{F}_q[x]_{<k}\big\}. \tag{1.17}$$

✓ L'application $ev(P)$ est une application linéaire et $\mathbb{F}_q[x]_{<k}$ est l'ensemble des polynômes de degré strictement inférieur à k.

✓ Les codes GRS sont optimaux car ils sont MDS (Maximum Distance Separable) ; la distance minimale d'un code $GRS(n,k)$ est :

$$d_{min} = n - k + 1. \tag{1.18}$$

✓ Pour tout vecteur v à coefficients dans $\mathbb{F}_q$, il existe un unique polynôme V de degré inférieur ou égal à $n-1$ tel que $ev(V) =$ v.

Définition 1.5 :

Soit C un code de Reed-Solomon de support $S = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$. Une matrice génératrice de C est sous la forme [35], [36] :

$$\mathrm{G} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ (\alpha_1)^2 & (\alpha_2)^2 & (\alpha_3)^2 & \cdots & (\alpha_n)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\alpha_1)^{k-1} & (\alpha_2)^{k-1} & (\alpha_3)^{k-1} & \cdots & (\alpha_n)^{k-1} \end{bmatrix} \tag{1.19}$$

avec le système matriciel suivant :

$$[c_0 \ c_1 \ \cdots \ c_{n-1}] = [u_0 \ u_1 \ \cdots \ u_{k-1}]\mathrm{G} \tag{1.20}$$

- ***Approche basée sur la détermination du polynôme générateur***

C'est l'approche utilisée avec les codes BCH. Elle consiste à déterminer le polynôme générateur $g(x)$ du code, diviseur de $x^n + 1$.

Les racines d'un polynôme générateur d'un code de Reed-Solomon, de distance construite δ, sont sous la forme α^{l+i} avec $0 \leq i \leq \delta - 2$ et $l = 0$ ou $l = 1$ où α est un élément primitif du corps de Galois $\mathbb{F}_q$ et est :

$$g(x) = ppmc(m_{\alpha^l}(x), m_{\alpha^{l+1}}(x), \cdots, m_{\alpha^{l+\delta-2}}(x)). \tag{1.21}$$

Le polynôme minimal $m_{\alpha^{l+i}}(x)$ à coefficients dans $\mathbb{F}_q$ admet, comme unique racine, l'élément α^{l+i} qui lui est associé [32]. Par conséquent nous aurons :

$$g(x) = (x - \alpha^l)(x - \alpha^{l+1}) \cdots (x - \alpha^{l+\delta-2}). \tag{1.22}$$

En résumé, un code de Reed-Solomon sur un corps de Galois $\mathbb{F}_q$ de pouvoir de correction t donné est un code de longueur $n = q - 1$, de dimension $k = n - 2t$, de distance minimale $d_{min} = \delta = n - k + 1 = 2t + 1$.

Les codes RS sont utilisés dans beaucoup d'applications courantes telles que les CD (Compact Disc), les DVD (Digital Versatile Disc), etc.

1.4 Le décodage des codes en bloc linéaires

1.4.1 Détection des erreurs

Soit $C(n,k)$ un code en bloc linéaire et H^{T} la transposée de sa matrice de contrôle de parité. Soit a un mot quelconque, la quantité $\mathrm{s}(a) = a\mathrm{H}^{\mathrm{T}}$ est appelée ***syndrome*** de a. Soit $\mathrm{r} = \mathrm{c} + \mathrm{e}$ le mot reçu lorsque le mot de code $\mathrm{c} \in C$ est transmis ; e étant la configuration d'erreurs survenues lors de la transmission. Nous avons, d'après (1.1), $\mathrm{s}(\mathrm{c}) = \mathrm{c}\mathrm{H}^{\mathrm{T}} = 0 \Leftrightarrow \mathrm{c} \in C$. Quand on calcule le syndrome du mot reçu r, on obtient :

$$\mathrm{s}(\mathrm{r}) = \mathrm{r}\mathrm{H}^{\mathrm{T}} = \mathrm{c}\mathrm{H}^{\mathrm{T}} + \mathrm{e}\mathrm{H}^{\mathrm{T}} = \mathrm{e}\mathrm{H}^{\mathrm{T}} = \mathrm{s}(\mathrm{e}). \tag{1.23}$$

D'après cette expression, le mot reçu et l'erreur ont le même syndrome. Nous pouvons donc dire que le syndrome ne dépend pas du mot reçu (« patient ») mais dépend de l'erreur (« maladie »). Deux cas de figure peuvent se présenter :

- le syndrome est nul ; on conclut qu'il n'y a pas d'erreurs ou du moins pas d'erreurs détectables. Notons qu'un syndrome nul ne permet pas d'affirmer qu'il n'y a aucune erreur. En effet, une combinaison particulière d'erreurs peut aboutir à un syndrome nul.
- le syndrome est non nul ; il existe au moins une erreur. Les symboles non nuls de e représentent les positions des erreurs. Dans le cas où e n'admet qu'un seul symbole non nul, soit $e_j \neq 0$, le syndrome est égal à la colonne j de la matrice de contrôle de parité. Sinon s s'écrit comme combinaison de plusieurs colonnes de H.

On appelle *coset* ou *classe latérale* d'un syndrome s l'ensemble définit par :

$$\mathcal{C}(\mathrm{s}) = \{a \in \{0,1\}^n \mid a\mathrm{H}^{\mathrm{T}} = \mathrm{s}\} \tag{1.24}$$

Le *leader* du coset est le vecteur de poids minimal appartenant à ce coset.

1.4.2 Correction des erreurs de transmission

- ***Décodage à maximum de vraisemblance a posteriori***

Le décodage Maximum Likelihood (ML) sélectionne le mot de code $\hat{\mathrm{c}}$ le plus probable (ayant la plus petite distance de Hamming avec le mot reçu r) parmi tous les mots de code

possibles de manière à maximiser la probabilité Pr(r|c). Si on considère que la probabilité d'erreur $p_e < \frac{1}{2}$, on aura [33] :

$$\hat{c} = a \Leftrightarrow \mathrm{d}_H(\mathrm{r}, a) \leq \mathrm{d}_H(\mathrm{r}, \mathrm{b}),\ \forall\, \mathrm{b} \in C(n,k). \tag{1.25}$$

Cette procédure de décodage devient difficile à mettre en œuvre si le nombre de mots de code est important ; ce qui est souvent le cas pour les codes en blocs les plus utilisés.

- ***Décodage à partir du syndrome***

Comme son nom l'indique, ce type de décodage utilise le calcul de syndrome pour corriger les erreurs. Nous savons, d'après la relation (1.23), que le mot reçu $\mathrm{r} = \mathrm{c} + \mathrm{e}$ et la séquence d'erreurs e ont même syndrome. Le problème de décodage peut donc se résumer à trouver le vecteur e de poids minimal dans la classe latérale de r. Pour ce faire, on peut utiliser, soit une table de correspondance entre les syndromes et les configurations d'erreurs associées, soit calculer le vecteur d'erreurs. La table est composée de deux colonnes et sur chaque ligne se trouve un syndrome et le vecteur d'erreurs correspondant. Le calcul de la séquence d'erreurs peut se faire par évaluation de la position des erreurs et de la valeur de leurs amplitudes. Autrement dit, on cherche les positions i pour lesquelles la composante e_i de la séquence d'erreurs e est non nul et, dans ce cas, on détermine la valeur de e_i. Ce type de décodage est utilisé avec les codes RS et les codes BCH binaires en utilisant une approche polynômiale. Après avoir trouvé e, il suffit de poser $\mathrm{c} = \mathrm{e} + \mathrm{r}$ pour avoir le mot de code qui a été émis. Le décodage est réussi si $w_H(\mathrm{e}) \leq t$.

- ***Décodage par propagation de croyances***

L'algorithme de décodage par propagation de croyances BP (Belief Propagation) est un algorithme itératif basé sur un échange d'informations probabilistes entre les différents nœuds (de variable et de contrôle) d'un graphe représentant le code considéré. L'information échangée entre les nœuds est sauvegardée et réutilisée lors des itérations suivantes.

1.5 Codes convolutifs

Les codes convolutifs peuvent être systématiques ou non [32]. Ils s'appliquent sur une suite infinie de données et génèrent des séquences de mots codés infinies. Pour ces codes, chaque bloc de n éléments binaires en sortie dépend non seulement des k éléments binaires présents en

entrée mais aussi des m blocs de k éléments binaires précédents, souvent k vaut 1. $m+1$ s'appelle la *longueur de contrainte*. Ainsi, le taux de codage r est égal à k/n.

1.5.1 Encodage des codes convolutifs

L'opération d'encodage utilisent des polynômes générateurs et est réalisé à l'aide de registres à décalage et de OU exclusifs (XOR). Considérons par exemple un codeur constitué de trois étages et deux sorties $s_1 = d_k \oplus d_{k-1} \oplus d_{k-2}$ et $s_2 = d_k \oplus d_{k-2}$, où d_k est le bit d'entrée et $\oplus$ le XOR.

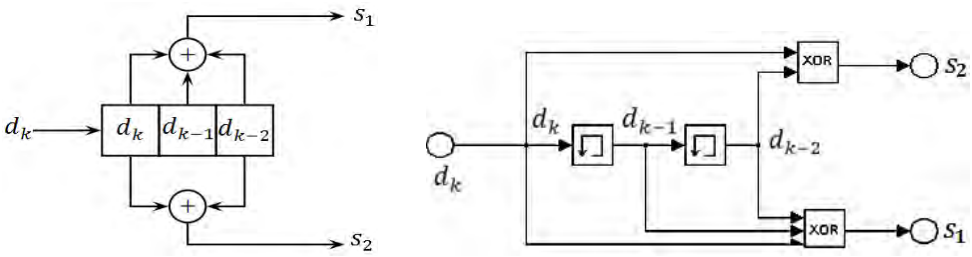

Figure 1.1 : Codeur convolutif, à gauche symbolique et à droite implémentation.

Le bit d'entrée d_k peut prendre 0 ou 1 pour le même état des registres d_{k-1} et d_{k-2}, les différents états possibles (d_k, d_{k-1}, d_{k-2}) sont donc $(000, 100, 001, 101, 010, 110, 011, 111)$ et les sorties respectives (s_1, s_2) sont $(00, 11, 11, 00, 10, 01, 01, 10)$. Ainsi le treillis est formé de nœuds reliés par des branches : les nœuds et les branches représentent respectivement les différents états possibles du codeur et les différentes transitions possibles d'un nœud à un autre lors de l'arrivée d'un bit d'entrée. Le treillis du code sera représenté comme suit :

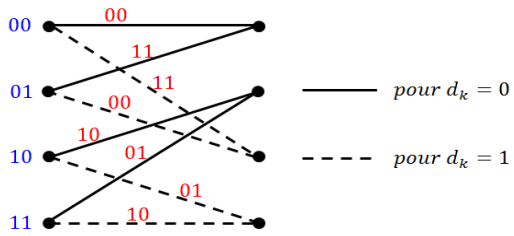

Figure 1.2 : Représentation en treillis d'un code convolutif.

Partant, par exemple de l'état 00, l'arrivée d'un **0** mène le codeur à l'état 00 et l'arrivée d'un **1** mène le codeur à l'état 10. A chaque branche, nous pouvons associer le mot codé (sortie), soit les 2 bits de code ici.

1.5.2 Décodage des codes convolutifs : décodage de Viterbi

Le décodage le plus courant des codes convolutifs est basé sur l'algorithme de Viterbi [37] qui décode un flot de données encodées à l'aide d'un encodeur convolutif. L'algorithme de Viterbi peut se faire avec des entrées fermes (on reçoit uniquement les bits codés) ou souples (on les reçoit avec des bits de fiabilité). L'algorithme de Viterbi utilise une représentation en treillis. Il consiste à rechercher dans l'arbre le chemin qui correspond à la séquence la plus probable, c'est-à-dire celle qui est à la distance minimale de la séquence reçue.

Avec le codeur de la Figure 1.2, supposons que l'on envoie la séquence binaire 00 1**1** 10 11 00 et qu'on reçoive 00 1**0** 10 11 00. La représentation en treillis de l'ensemble des chemins possibles est la suivante :

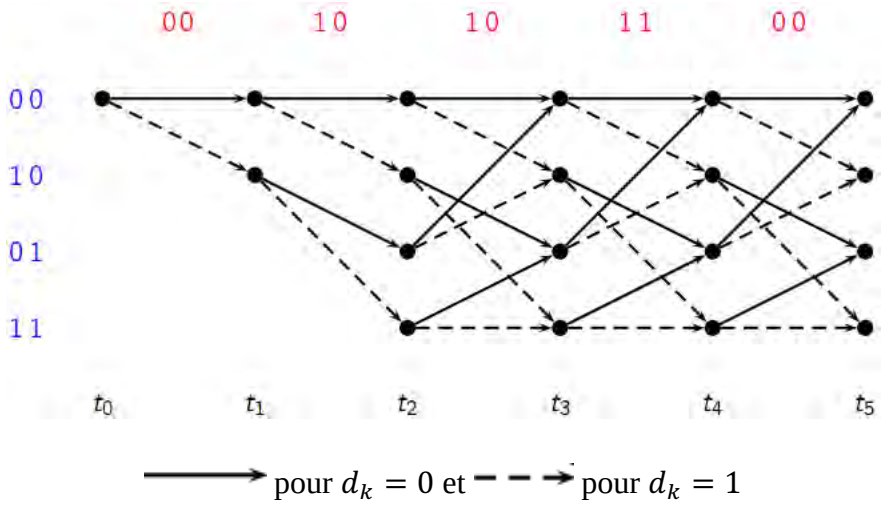

Figure 1.3 : Représentation en treillis du décodeur convolutif.

Partant de l'état initial 00, on ne peut atteindre que deux états : 00 ou 10 (voir Figure 1.2). L'état 00 est atteint si le bit entrant est 0. Dans ce cas, la séquence de sortie est 00. Une sortie à 00 est en accord avec la séquence reçue 00 : il n'y a donc pas d'erreur (0). En revanche, l'état 10 est atteint si le bit entrant est 1. La séquence de sortie est 11, ce qui diffère de 2 bits par rapport à la séquence reçue. Donc les deux états au temps t_1 sont atteints en faisant 0 ou 2 erreurs (Figure 1.4). Le même raisonnement est fait au temps t_2 à partir des deux nœuds résultants de t_1. Les métriques (en vert) correspondent au cumul des erreurs depuis la racine (état 00). Notons que lorsque deux chemins différents aboutissent à un même nœud, celui qui a une métrique plus élevée est abandonné (supprimé). Cela continue jusqu'au temps t_5. Comme nous pouvons le constater sur la Figure 1.4, la métrique minimale est 1.

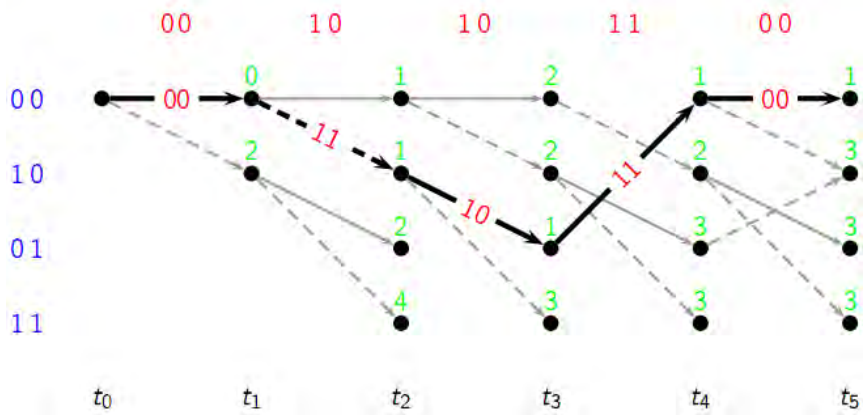

Figure 1.4 : Décodage de Viterbi sous forme de treillis.

Finalement, le chemin le plus vraisemblable est celui ayant le moins d'erreur, c'est-à-dire celui produisant la sortie la plus proche de la séquence reçue. Ici, c'est le chemin représenté en trait gras qui sera la séquence décodée (Figure 1.4). Ainsi, l'algorithme de Viterbi fournit en sortie 00 11 10 11 00 qui correspond à la séquence décodée 0 1 0 0 0.

Les codes convolutifs sont utilisés dans les systèmes de communications sans fil, la télévision numérique par satellite DVB-S et la téléphonie mobile.

Conclusion

Nous venons de passer en revue des notions très importantes en théorie des codes correcteurs d'erreurs et quelques exemples de ces codes en donnant leurs différentes caractéristiques. Les codes de Hamming, les codes BCH, les codes de Reed-Solomon et les codes convolutifs sont des codes correcteurs d'erreurs qui s'appliquent dans plusieurs domaines de la technologie. Ils sont aussi utilisés en stéganographie.

Chapitre 2: STEGANOGRAPHIE DANS LE DOMAINE SPATIAL

Introduction

Les techniques de dissimulation d'informations telles que la stéganographie sont anciennes et attirent de plus en plus l'intérêt de bon nombre de scientifiques. Le but de ce chapitre est d'introduire la stéganographie, la stéganalyse, ainsi que les différentes techniques et méthodes utilisées. Nous commençons par définir la stéganographie et les caractéristiques des schémas de stéganographie. Les différentes techniques de stéganographie, à savoir les méthodes de remplacement et de correspondance de LSB, le matrix embedding et la technique du papier mouillé, sont ensuite décrites. Nous allons également expliquer la minimisation de l'impact d'insertion, les différents scénarios de stéganalyse et les méthodes permettant de définir les fonctions de distorsion pour la stéganographie adaptative. Nous terminons ce chapitre en présentant les principaux schémas de stéganographie utilisant les codes correcteurs d'erreurs tels que Hamming [4], BCH [6], [7], RS [8], STC [12] et LDPC [11].

2.1 Généralités sur la stéganographie

Nous allons noter par $\mathrm{x} = (x_1, \dots, x_n) \in \mathcal{X} = \{I\}^n$ l'image de couverture et x_i son $i^{ème}$ pixel. Le message secret $\mathrm{m} = (m_1, \dots, m_m) \in \mathcal{M} = \{0,1\}^m$ est inséré en modifiant légèrement l'image de couverture. Cela donne une image stégo $\mathrm{y} = (y_1, \dots, y_n) \in \mathcal{Y} = I_1 \times \dots \times I_n$, où $I_i \subset I$ tel que $x_i \in I_i$. Pour les images en niveaux de gris sur 8-bits, utilisées dans cette thèse, $I = \{0, \dots, 255\}$. Pour des soucis de simplification, nous allons indifféremment désigner par les mêmes notations les images et les portions qui en découlent. Par exemple, l'image de couverture sera appelé x et un segment de l'image de couverture sera aussi désigné par x.

2.1.1 Définitions de la stéganographie

La sécurisation de l'information réunit plusieurs techniques comme la cryptographie qui assure la sécurité de la communication. Cependant, souvent même l'existence de la communication nécessite d'être secrète. Le message d'une telle communication secrète peut être incorporé dans d'autres communications insoupçonnées en utilisant la stéganographie. Cette technique permet de dissimuler une information secrète dans des médiums de couverture numériques tels que l'image, le son ou la vidéo de sorte qu'elle soit indétectable. Ainsi, contrairement à la cryptographie qui a pour seul objet la robustesse, la stéganographie vise

d'abord la confidentialité de l'existence de la communication (message) avant la robustesse. La tâche de l'expéditeur (ou stéganographe) [38] est de dissimuler son message secret (ou payload) dans le médium de couverture de telle sorte que seul le destinataire soit au courant de l'existence du message. Dans la stéganographie moderne ou numérique, il existe deux manières de définir de bons algorithmes stéganographiques pour les images numériques. La première méthode repose sur la définition d'un bon modèle de couverture. La seconde et plus moderne approche est basée sur la dissimulation du message secret tout en minimisant une certaine fonction de distorsion. Cette dernière approche est plus flexible et est à la base de la majorité des méthodes de stéganographie actuelles. La stéganographie moderne a véritablement débuté vers 1996. Son développement est favorisé par celui de l'internet qui permet un partage intense de données multimédias à longueur de journée. La stéganographie moderne utilise donc ces médiums pour la dissimulation d'informations secrètes. Les utilisations peuvent être à des fins de protection et de sécurisation d'informations personnelles secrètes ou à des fins malveillantes. Selon certains médias, les attentats du 11 septembre 2001, perpétrés aux Etats-Unis, auraient été planifiés en utilisant des messages secrets cachés au sein d'images numériques échangées à travers internet. Dans le cadre de cette thèse, nous utiliserons les images numériques et l'insertion se fera dans le domaine spatial.

2.1.2 Les différents types et formats d'images

Les algorithmes de stéganographie dépendent de la structure des données dans lesquelles se fait l'insertion. Les modifications devant être imperceptibles, il faut donc altérer le support dans les endroits les plus discrets, ce qui dépend fortement du type de données (image, audio, vidéo, …) et de leur format de représentation (JPEG, GIF, MP3, ...).

2.1.2.1 La notion de pixel

Une image est constituée d'un ensemble de points appelés pixels (**pic**ture **el**ement). Le pixel représente le plus petit élément constitutif d'une image numérique. L'ensemble des pixels est contenu dans un tableau à deux dimensions constituant l'image.

2.1.2.2 Les différents types d'images

La propriété de base d'une image est son *mode*. Il en existe de quatre types :

- ✓ *les images binaires (noir et blanc)*

Une image binaire est représentée par une matrice dont les éléments (pixels) sont des valeurs binaires (0 pour le noir et 1 pour le blanc).

- ✓ *les images en teintes ou en densités ou en niveaux de gris*

Dans une image en niveaux de gris de taille (M, N), chaque pixel est représenté par son intensité lumineuse, de 0 (noir) à 255 (blanc) pour les types entiers sur un octet. Entre les deux valeurs nous avons les nuances de gris.

- ✓ *les images RGB* (*Red Green Blue*) *ou RVB* (*Rouge Vert Bleu*)

La nomination RGB indique que chaque pixel de l'image est représenté par un niveau de rouge, un niveau de vert et un niveau de bleu. Un pixel utilise donc 3 octets (24 bits) pour coder un triplé d'entiers. Une image RGB est donc de taille $(M, N, 3)$.

- ✓ *les images indexées ou images palettisées*

La couleur de chaque pixel est codée sur un seul octet dans une table de couleurs. Cette table de couleurs de taille $(N, 3)$, la palette de couleurs (color map), définit jusqu'à 256 couleurs utilisables dans l'image en donnant la décomposition RGB de chaque couleur sur 3 octets.

2.1.2.3 Les différents formats d'images

Les formats d'image sont nombreux et variés. Nous avons, par exemple, PNG (Portable Network Graphics), GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group), etc. Le domaine spatial concerne les images fixes non compressées pouvant être représentées par différents formats BMP, RAW, TIFF, PGM ... etc. Dans le domaine JPEG, les images sont compressées et représentées en blocs de 8×8 pixels avec des coefficients DCT (Discret Cosinus Transform) quantifiés.

Notons que pour nos tests dans ce manuscrit, nous utiliserons des images numériques contenues dans la base BOSS [39] d'images en niveaux de gris de même taille 512×512 et de même format PGM.

2.1.3 Le problème ou modèle des prisonniers

Bien que la communauté des stéganographes se soit constituée dans les années 90, G. J. Simmons pose en 1983 le socle de la stéganographie moderne en définissant la notion de canal subliminal [2]. Pour illustrer son propos, il reprend le problème des prisonniers. Dans ce contexte, Alice et Bob sont deux prisonniers enfermés dans deux cellules séparées l'une de l'autre. Ils sont autorisés à communiquer par l'intermédiaire d'une gardienne Eve, comme indiqué sur la Figure 2.1. Si celle-ci soupçonne qu'ils élaborent un plan pour s'échapper, elle s'autorise à mettre fin à la communication entre les deux détenus et de lourdes sanctions leur seront infligées. De plus, Eve a la possibilité de modifier les messages. Dans une telle situation l'utilisation de messages chiffrés éveillerait des soupçons. Ainsi, la seule alternative d'Alice et

Bob est de s'envoyer des messages innocents et de dissimuler l'information secrète dans ceux-ci. Pour ce faire, ils mettent en place un canal de transmission (par l'intermédiaire des messages) qui n'est pas visible pour Eve ; ce canal est appelé canal subliminal.

Nous supposons tout d'abord qu'Alice et Bob se sont échangés au préalable une clé secrète stéganographique que la gardienne ignore. Leur plan est réussi s'ils arrivent à communiquer sans éveiller le moindre soupçon de la part d'Eve.

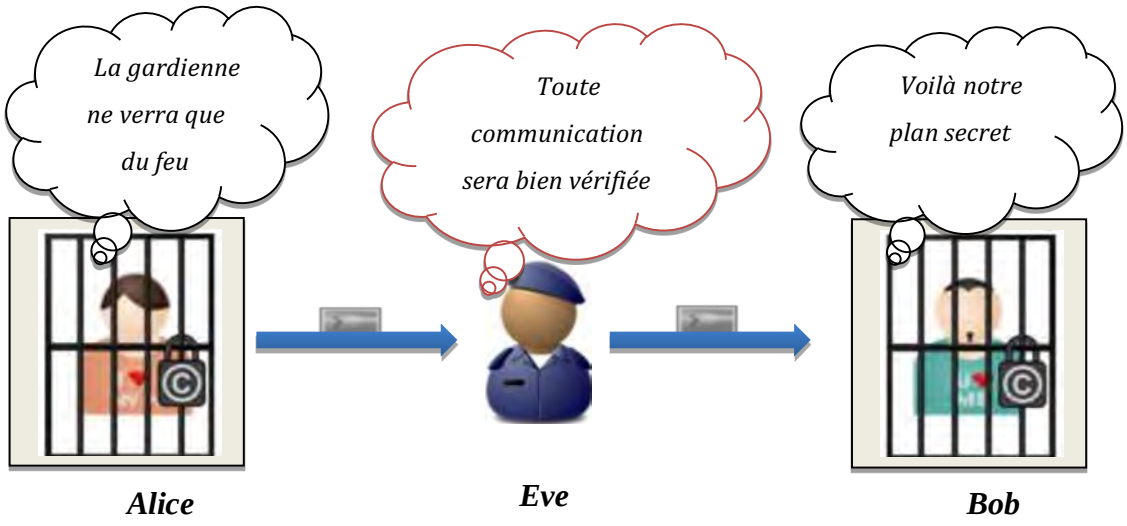

Figure 2.1 : Problème des prisonniers.

2.1.4 Définition des schémas de stéganographie

Les schémas de stéganographie sont définis en se basant sur le problème des prisonniers que nous venons de présenter. En effet, ils s'appuient sur deux fonctions : l'une utilisée par l'émetteur (Emb) du message et l'autre par le destinataire (Ext), comme le décrit la Figure 2.2. Soit $\mathbb{F}_2^m = \{0,1\}^m$ l'ensemble des combinaisons possibles pour le message à insérer et $\mathbb{F}_2^n = \{0,1\}^n$ l'ensemble de tous les vecteurs de couverture susceptibles d'être utilisés.

- ✓ La fonction d'insertion (*embedding* en anglais) est définie par $Emb : \{0,1\}^n \times \{0,1\}^m \mapsto \{0,1\}^n$. Elle fournit l'image stégo à partir de l'image de couverture et le message.
- ✓ La fonction d'extraction est représentée par $Ext : \{0,1\}^n \mapsto \{0,1\}^m$ et retourne le message secret en prenant en entrée l'image stégo.
- ✓ Ces deux fonctions peuvent être associées à une clè secète partagée entre l'émetteur et destinataire du message afin d'accroitre la sécurité.

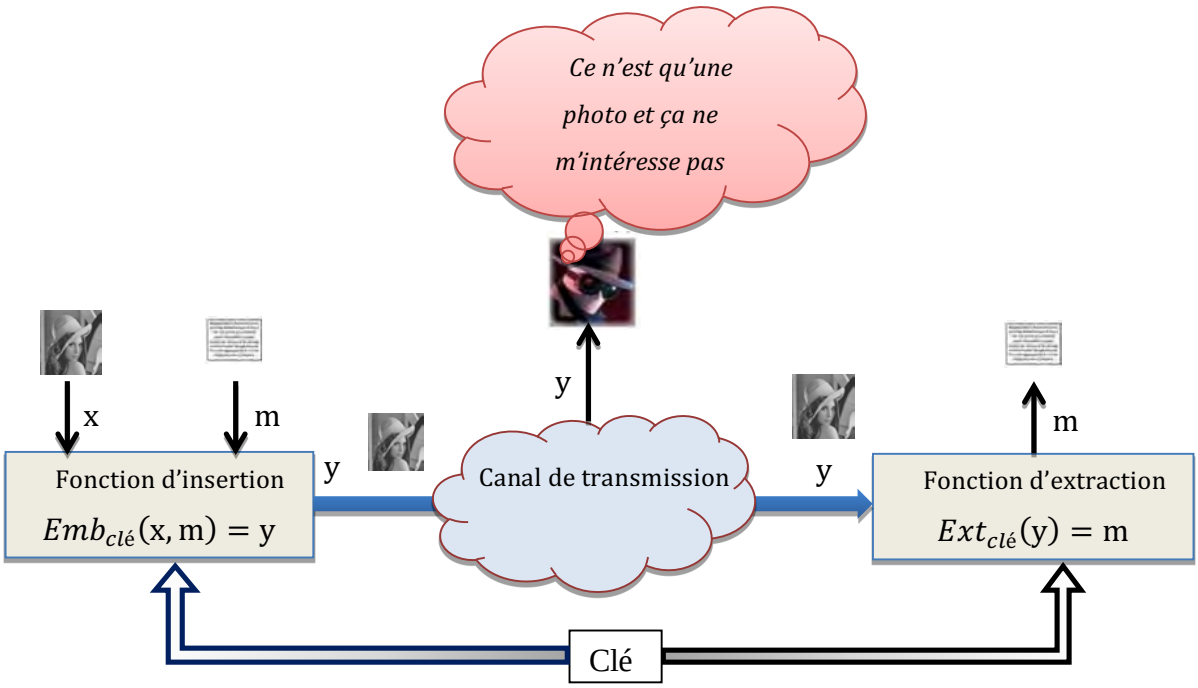

Figure 2.2 : Schéma global de la stéganographie.

Notons que le message peut être sous forme pseudo-aléatoire après avoir été préalablement compressé ou crypté par exemple.

Après avoir défini un schéma de stéganographie, il convient de donner ses caractéristiques.

2.1.5 Les caractéristiques d'un schéma de stéganographie

L'objectif est d'insérer le maximum de bits tout en modifiant le moins le support. Il nous faut donc connaître les paramètres d'évaluation d'un schéma de stéganographie parmi lesquels nous pouvons citer :

- ✓ La ***capacité d'insertion*** : c'est le nombre maximal de bits que nous pouvons insérer dans un médium de couverture. Si nous utilisons des images en niveaux de gris, nous pouvons cacher un bit dans chaque pixel avec une capacité d'insertion égale à la taille de l'image.
- ✓ Nous définissons le ***taux d'insertion*** comme étant le nombre de bits insérés du message par élément du support et la ***densité de changement*** par la proportion de composantes modifiées du médium de couverture.
- ✓ L'***efficacité d'insertion*** : c'est le nombre de bits du message secret par unité de distorsion. C'est le rapport du taux d'insertion par la densité de changement. Si l'impact de toutes les modifications est le même, on peut mesurer l'efficacité d'insertion comme le nombre de bits du message par modification du support.

- ✓ Le ***PSNR (Peak Signal to Noise Ratio)*** : ou rapport de signal crête sur bruit est une mesure (en dB) de distorsion entre deux images, souvent utilisée en traitement d'images. Il est calculé à partir du MSE (Mean Square Error) ou erreur quadratique moyenne. Soient respectivement I_o et I_r les images originale et reconstruite de même taille $M \times N$. Le PSNR et le MSE sont donnés par :

$$PSNR(I_o, I_r) = 10 \log_{10} \frac{Din^2}{MSE(I_o, I_r)} \tag{2.1}$$

$$MSE(I_o, I_r) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \big(I_o(i,j) - I_r(i,j)\big)^2 \tag{2.2}$$

où Din est la dynamique (la valeur maximale d'un pixel). Si les pixels sont codés sur 8 bits $Din = 2^8 - 1 = 255$. *Plus la valeur du PSNR est importante, plus les images comparées sont semblables.* Un PSNR de plus de $35\,dB$ entre deux images signifie qu'il n'y a aucune différence visible entre ces deux images [40]. Si le PSNR est inférieur à $20\,dB$ alors les deux images sont très différentes.

2.2 Les différentes techniques de stéganographie

Plusieurs schémas ont été proposés par les stéganographes. Ces schémas sont basés principalement sur trois techniques : la technique du LSB (Least Significant Bit) par remplacement et correspondance, la technique du matrix embedding et celle du papier mouillé. La technique de base est celle du LSB et les autres techniques constituent des améliorations et compléments de celle-ci.

2.2.1 La technique du remplacement de LSB

Le LSB désigne le bit le moins significatif (bit de poids le plus faible) dans la représentation binaire d'un nombre. La technique la plus simple utilisée dans la stéganographie numérique est celle de la substitution de LSB. Comme son nom le laisse entendre, cette technique remplace le LSB des pixels de l'image de couverture par les bits du message. Une image obtenue par modification des bits de poids faible ne paraît pas suspecte car elle est visuellement proche de l'originale. En effet, après insertion, soit la valeur du pixel ne change pas, soit elle ne varie que d'une unité (incrémentation $+1$ si la valeur du pixel est paire et décrémentation -1 pour le cas d'une valeur impaire) contrairement aux autres positions où un changement d'un seul bit peut

faire varier grandement la valeur du pixel. Le processus de l'insertion par substitution de LSB peut être illustré par le schéma ci-après :

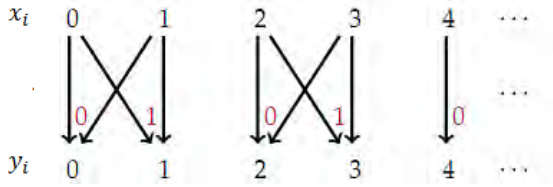

Figure 2.3 : Processus d'insertion par la technique de substitution de LSB.

Les nombres en noirs représentent les valeurs des pixels avant insertion (au-dessus) et après insertion (en dessous) et ceux en rouges désignent les bits du message à insérer. Pour la méthode de remplacement de LSB, l'ensemble des valeurs des pixels après insertion est $I_i = \{x_i, \bar{x}_i\}$, pour tout i, où $\bar{x}_i$ représente le pixel x_i dont le LSB est modifié.

Exemple 2.1 :

Considérons les pixels d'une image en niveaux de gris ci-dessous. Pour dissimuler le message binaire suivant **100110011**, on remplace les LSBs de l'image de couverture par les bits du message.

00100111	11101001	11001000
00100111	11001000	11101000
11001000	00100111	11101001
	⋮	

00100111	11101000	11001000
00100111	11001001	11101000
11001000	00100111	11101001
	⋮	

Même si la technique du LSB paraît sûre par attaque visuelle, elle ne l'est pas pour autant par analyse des statistiques de l'image stégo et par conséquent, un attaquant peut se concentrer sur l'étude des statistiques des LSBs. En effet, un pixel de valeur impaire sera soit inchangé ou décrémenté mais jamais incrémenté. De même, un pixel de valeur paire sera soit inchangé ou incrémenté mais jamais décrémenté. Cette asymétrie introduit une anomalie statistique dans l'histogramme d'intensité. En effet, les valeurs de pixel 0 et 1, 2 et 3, etc., vont, en moyenne, présenter la même fréquence dans une image stégo. La présence de ces comportements suspects peut être exploitée par la stéganalyse afin de déceler la présence d'un message secret. Fort de ces constats, une amélioration de la technique par substitution de LSB est proposée.

2.2.2 La technique de LSB matching (par correspondance de LSB)

Cette méthode de stéganographie, aussi appelée insertion ± 1, a été proposée en vue d'apporter une solution au problème statistique noté avec la méthode de stéganographie par substitution. C'est une version légèrement améliorée de la stéganographie par substitution de LSB. La différence est que, plutôt que de remplacer le LSB par le bit du message, la valeur du pixel correspondant est incrémentée $(+1)$ ou décrémentée (-1) aléatoirement si la valeur du LSB doit être changée (c'est-à-dire si le bit à insérer diffère de celui du LSB). En effet, contrairement à la substitution de LSB, la correspondance de LSB n'introduit pas les artefacts caractéristiques sur la distribution statistique du premier ordre de l'image. Par conséquent, les méthodes d'attaque spécifiquement dédiées à la détection de la stéganographie par substitution de LSB sont inefficaces pour détecter la correspondance de LSB. Heureusement pour la stéganalyse, d'autres anomalies statistiques sont créées et apportent encore une discrimination (distinction) entre les images de couverture et stégo. Cependant, ces anomalies sont plus subtiles et la précision de la discrimination est nettement plus faible que pour le remplacement de LSB. La technique de remplacement de LSB est ternaire et $I_i = \{x_i - 1, x_i, x_i + 1\}$. L'algorithme par correspondance de LSB peut être résumé comme suit :

$$y_i = \begin{cases} x_i & si \;\; LSB(x_i) = m_i \\ x_i + 1 & si \;\; (LSB(x_i) \neq m_i) \;\; et \;\; (dr < 0 \; ou \; x_i = 0) \\ x_i - 1 & si \;\; (LSB(x_i) \neq m_i) \;\; et \;\; (dr > 0 \; ou \; x_i = 255) \end{cases} \tag{2.3}$$

où dr est une variable aléatoire $i.i.d.$ avec une distribution uniforme sur $\{-1, +1\}$, x_i et y_i sont respectivement les valeurs des pixels de l'image de couverture et stégo. Le processus d'insertion ± 1 peut être illustré par le schéma ci-dessous :

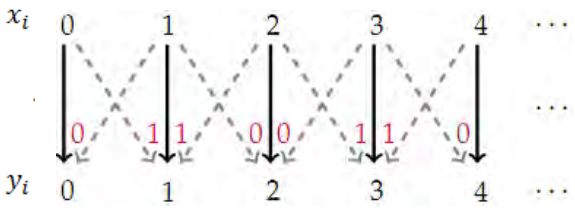

Figure 2.4 : Processus d'insertion par la technique de correspondance de LSB.

2.2.3 La technique du Matrix Embedding

Considérons le vecteur x constitué des LSBs de l'image de couverture, le message m et une matrice de contrôle de parité H d'un code correcteur d'erreurs. Le principe du matrix

embedding (ou codage matriciel) [2] consiste à trouver le vecteur stégo y le plus proche de x tel que :

$$yH^T = m \tag{2.4}$$

En remplaçant y par x + e (e le vecteur des modifications) on aura :

$$eH^T = m - xH^T \tag{2.5}$$

Cette dernière égalité montre que le vecteur e admet pour syndrome $m - xH^T$. Le problème consistera donc :

- ✓ pour l'émetteur, à trouver le vecteur e de poids minimal dans le coset $\mathcal{C}(m - xH^T)$ (l'ensemble des vecteurs de même taille que x et de syndrome $m - xH^T$) et ensuite l'additionner avec x pour trouver y ;
- ✓ pour le récepteur, à faire le produit matriciel $m = yH^T$ afin de trouver le message m.

Tous les pixels de l'image de couverture sont susceptibles d'être modifiés. Si nous voulons garder inchangés certains d'entre eux, nous utilisons la technique du papier mouillé.

2.2.4 La technique du papier mouillé

Supposons que nous voulions écrire sur un papier mouillé en partie, la solution sera d'écrire sur les parties sèches du papier. En utilisant cette description, nous définissons la technique du papier mouillé. Celle-ci utilise la technique de matrix embedding décrite précédemment. La méthode de sélection de pixels consiste à choisir les pixels dont la modification est la moins perceptible par le système visuel humain et ayant le moins d'effets statistiques sur l'image. Cette méthode de sélection des pixels peut être partagée avec le destinataire du message. Cela signifie qu'elle est la même pour toute image de couverture ou partagée juste après la communication. Dans les deux cas, cela pose soit un problème de performance soit un problème de sécurité. La technique du papier mouillé vient proposer une solution à cette règle de sélection des pixels [41].

Nous considérons que les bits d'indices appartenant à $\mathcal{J} \subset [1, \cdots, n]$ sont à verrouiller. Puisque nous verrouillons les bits d'indice dans $\mathcal{J}$, nous avons $e_i = 0$ pour tout $i \in \mathcal{J}$ car nous devons avoir $x_i = y_i$. Nous pouvons résoudre le problème de matrix embedding soit directement avec les deux relations précédentes tout en tenant compte du fait des composantes nulles de e ($e_i = 0, i \in \mathcal{J}$) soit en élaguant les lignes de H^T (les colonnes de H) d'indices dans $\mathcal{J}$. Pour la seconde approche, considérons $H_{\mathcal{J}}$ la matrice obtenue après avoir élagué les

colonnes de H et les vecteurs $e_{\mathcal{J}}$, $x_{\mathcal{J}}$, $y_{\mathcal{J}}$ et $m_{\mathcal{J}}$ désignant respectivement les simplifications des vecteurs e, x, y et m en éliminant les éléments dont les indices appartiennent à $\mathcal{J}$. Nous aurons ainsi [41]:

$$y_{\mathcal{J}}H_{\mathcal{J}}^{T} = m_{\mathcal{J}} \quad \text{et} \tag{2.6}$$

$$e_{\mathcal{J}}H_{\mathcal{J}}^{T} = m_{\mathcal{J}} - x_{\mathcal{J}}H_{\mathcal{J}}^{T} \tag{2.7}$$

La résolution se fera avec les nouvelles relations obtenues. Avec la technique du papier mouillé, le destinataire du message n'a pas besoin de connaître les positions choisies pendant l'insertion pour extraire le message. Il peut retrouver ce message en calculant le syndrome du vecteur stégo reçu.

2.2.5 Codage par syndrome

La technique appelée codage par syndrome peut être utilisée dans la pratique pour réaliser le problème de matrix embedding. Soit $LSB: I_i \rightarrow \{0,1\}$ définie par $LSB(y) = y \; mod \; 2$. L'émetteur et le récepteur utilisent, respectivement, les fonctions d'insertion et d'extraction $Emb: \mathcal{X} \times \mathcal{M} \rightarrow \mathcal{Y}$ et $Ext: \mathcal{Y} \rightarrow \mathcal{M}$ satisfaisant à

$$Ext\big(Emb(x, m)\big) = m \quad \forall x \in \mathcal{X}, \forall m \in \mathcal{M}. \tag{2.8}$$

L'insertion est considérée comme étant universelle, car la fonction de distorsion entre x et y $D(x, y)$ est inconnue du côté du récepteur. Pour un code linéaire C de longueur de bloc n et de dimension $n - m$

$$\begin{aligned} Emb(x, m) &= \underset{LSB(y) \in \mathcal{C}(m)}{\arg\min} \; D(x, y) \\ Ext(y) &= LSB(y) \cdot H^{T} = m \end{aligned}, \tag{2.9}$$

où $LSB(y) = \big(LSB(y_1), \dots, LSB(y_n)\big)$, $H \in \{0,1\}^{m \times n}$ désigne une matrice de contrôle de parité du code C, $\mathcal{C}(m) = \{z \in \{0,1\}^n \mid zH^{T} = m\}$ est le coset correspondant au message secret m. Nous supposons que toutes ces opérations sont binaires. Le codage par syndrome atteint la capacité (la limite théorique) pour le problème d'insertion lorsqu'un code linéaire aléatoire est utilisé. Un tel code n'est pas pratique en raison de la complexité exponentielle du décodeur qui permettrait de chercher le coset binaire optimal de (2.9). Plusieurs codes sont utilisés pour approcher cet optimum.

2.3 Stéganographie et minimisation de distorsion

2.3.1 Définition d'une fonction de distorsion pour une insertion binaire

Pour les images de couverture numériques, l'expéditeur dissimule son message tout en minimisant la fonction de distorsion D entre x et y. Cette fonction de distorsion peut être écrite sous forme additive dans laquelle les modifications n'interagissent pas [19] :

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \rho_i(\mathrm{x}, y_i) \tag{2.10}$$

où $\rho_i: \mathcal{X} \times I_i \to \mathbb{R}$, désigne le coût de remplacer le pixel x_i par y_i. La valeur ρ_i dépend de l'ensemble de l'image de couverture x. Ce qui permet à l'expéditeur de tenir compte des interdépendances entre les différents pixels [13], [42]. Pour une opération d'insertion binaire, sachant que $I_i = \{x_i, \bar{x}_i\}$ nous pouvons réécrire (2.10) comme suit :

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \rho_i^{min} + \sum_{i=1}^{n} \varrho_i \cdot [\, \rho_i^{min} < \rho_i(\mathrm{x}, y_i)]. \tag{2.11}$$

où $\varrho_i = |\rho_i(\mathrm{x}, x_i) - \rho_i(\mathrm{x}, \bar{x}_i)| \geq 0$, $\rho_i^{min} = min\,\{\rho_i(\mathrm{x}, x_i), \rho_i(\mathrm{x}, \bar{x}_i)\}$ et [P] la valeur logique donnant 1 si P est vrai et 0 sinon. Puisque $\sum_{i=1}^{n} \rho_i^{min}$ ne dépend pas de l'image stégo y, la minimisation de D sur y revient à minimiser le second terme qui peut être décrite par [19]

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \varrho_i \cdot [\, y_i \neq x_i]. \tag{2.12}$$

2.3.2 Problème stéganographique et les limites théoriques

Bien qu'il n'y ait aucune mesure pratique et générale de la sécurité en stéganographie, la plupart des systèmes de stéganographie adaptatifs, basés sur la minimisation d'une fonction de distorsion liée à la détectabilité statistique, améliorent la sécurité de la stéganographie. En général, plus la distorsion est faible, moins l'image de couverture est altérée. Supposons que l'image stégo y est une variable aléatoire sur $\mathcal{Y}$ et sa distribution est notée par $\pi_{\mathrm{x}}(\mathrm{y}) \triangleq Pr(\mathrm{Y} = \mathrm{y}|\mathrm{x})$. Alors, la charge maximale que l'émetteur peut transmettre au récepteur tout en introduisant une distorsion attendue $E_{\pi_{\mathrm{x}}}[D]$ est l'entropie, notée $h(\pi_{\mathrm{x}})$ [43], [44]. Deux méthodes existent pour l'insertion d'un message : le DLS (Distortion-Limited Sender) et le PLS (Payload-Limited Sender). Cependant, le PLS est plus utilisé que le DLS. Cette technique

sera utilisée et consiste à déterminer la distribution π_{x} qui permet l'insertion d'un message de taille fixe m bits avec une distorsion minimale

$$\begin{aligned} &\underset{\pi_{\mathrm{x}}}{minimiser} && E_{\pi_{\mathrm{x}}}[D] = \sum_{y\in\mathcal{Y}} \pi_{\mathrm{x}}(\mathrm{y})\, D(\mathrm{x},\mathrm{y}) \\ &sous\ contrainte && h(\pi_{\mathrm{x}}) = -\sum_{y\in\mathcal{Y}} \pi_{\mathrm{x}}(\mathrm{y}) \log_2 \pi_{\mathrm{x}}(\mathrm{y}) = m \end{aligned} . \tag{2.13}$$

La distribution π_{x} optimale est sous la forme d'une distribution de Gibbs [43]

$$\pi_{\mathrm{x}}(\mathrm{y}) = \frac{\exp\left(-\lambda D(\mathrm{x},\mathrm{y})\right)}{Z(\lambda)} \overset{(a)}{=} \prod_{i=1}^{n} \frac{\exp\left(-\lambda \rho_i(\mathrm{x}, y_i)\right)}{Z_i(\lambda)} \triangleq \prod_{i=1}^{n} \pi_{\mathrm{x},i}(y_i), \tag{2.14}$$

où $Z(\lambda) = \sum_{\mathrm{y}\in\mathcal{Y}} \exp\left(-\lambda D(\mathrm{x},\mathrm{y})\right)$ est le facteur de normalisation, $Z_i(\lambda) = \sum_{y_i\in I_i} \exp\left(-\lambda \rho_i(\mathrm{x}, y_i)\right)$ et le paramètre $\lambda \in [0,\infty[$ est obtenu en résolvant $h(\pi_{\mathrm{x}}) = m$. Notons qu'il est possible de simuler l'insertion optimale π_{x} en remplaçant chaque pixel x_i avec la probabilité $\pi_{\mathrm{x},i}$. Pour évaluer les algorithmes de codage stéganographiques, on peut comparer leur efficacité d'insertion définie par $e(\alpha) = \alpha n / E_{\pi_{\mathrm{x}}}[D]$ (en unité de bits/distorsion) avec la limite théorique supérieure de (2.13), où $\alpha = m/n$ est la charge relative.

2.3.3 Minimisation d'impact d'insertion et codes à papier mouillé

De façon générale, le canal à papier mouillé peut être considéré comme une généralisation de la technique de sélection de pixels. Celle-ci est basée sur le concept d'écriture sur un papier humide. Dans ce contexte, l'expéditeur choisit les pixels pour lesquels leurs changements sont moins perceptibles par le système visuel humain et avec le moins d'effets statistiques sur l'image de couverture. Les pixels sélectionnés pour l'insertion ne sont connus que par l'expéditeur ; le récepteur n'a pas besoin de les connaître lors de l'extraction du massage. Si chaque pixel possède un coût d'insertion $\rho_i = 1$, la minimisation de la distorsion D est équivalente à minimiser le nombre de changements apportés sur le médium de couverture. On parle dans ce cas de *profil constant*. Souvent dans la pratique, il peut arriver qu'un ensemble de pixels de l'image de couverture soit interdit de changement. De tels pixels nécessitent que $I_i = \{x_i\}$ et sont appelés pixels « mouillés », avec $\rho_i = \infty$; l'algorithme de stéganographie doit garder ces pixels inchangés. Les autres pixels, appelés pixels « secs » sont affectés de $\rho_i < \infty$ et sont autorisés à être modifiés. Dans ce cas, on parle ce *canal à papier mouillé* qui est caractérisé par l'humidité relative $\tau = |\{i|\rho_i = \infty\}|/n$, où $|E|$ désigne la cardinalité de

l'ensemble E. Le codage par syndrome peut aussi être implémenté avec les codes à papier mouillé [41], [45], [46]. La charge relative est mesurée par $\alpha = m/|\{x_i|\ \rho_i < \infty\}|$.

L'efficacité d'un schéma de stéganographie est déterminée par la résistance face à la stéganalyse qui revêt diverses formes.

2.4 Stéganalyse ou détection de stéganographie

2.4.1 Définition de la stéganalyse

La stéganalyse (ou analyse stéganographique) est à la stéganographie, ce que représente la cryptanalyse à la cryptographie. La cryptanalyse a pour but de décrypter un message, sans connaitre la clé. Par analogie, le principe de la stéganalyse consiste à extraire l'information dissimulée sans connaissance de la clé du stégo-système. Une étude de la probabilité de réussite d'une attaque par force brute pour extraire l'information cachée montre la difficulté de la tâche. D'autant plus que le message peut avoir été préalablement crypté avant son insertion. C'est pour cette raison que les objectifs de la stéganalyse sont beaucoup plus modestes que l'extraction du message caché. Le problème de stéganalyse peut être formalisé par le problème de test d'hypothèses suivant :

$$\begin{cases} \mathcal{H}_0 : \text{ l'image z ne contient pas un message caché (couverture)} \\ \mathcal{H}_1 : \text{ l'image z contient un message caché (stégo)} \end{cases}$$

Le stéganalyste, doit décider entre ces deux hypothèses pour juger si oui ou non le médium est stéganographié. Il existe deux voire trois grandes classes de stéganalyse en fonction des objectifs recherchés et des moyens utilisés.

- ***La stéganalyse active (à gardien actif)*** : son objectif est d'empêcher la communication du message secret en modifiant la couverture de manière à supprimer le contenu dissimulé et en même temps conserver le contenu perceptible. Par exemple, pour les méthodes de stéganographie par substitution de LSB ou par correspondance de LSB, mettre tous les LSBs de tous les pixels de l'image stégo à 0 ou à 1, efface l'éventuel message caché.
- ***La stéganalyse passive (à gardien passif)*** : qui, comme son nom l'indique, ne s'autorise pas la modification du média lors de l'analyse. Son objectif se limite à détecter la présence d'informations cachées (faire le test d'hypothèses).
- ***La stéganalyse quantitative (à gardien malicieux)*** : dans ce cas, le stéganalyste va plus loin que détecter la présence d'un message caché ; il essaye d'estimer la taille ou même extraire

le message si possible. S'il parvient à extraire le message, il peut simplement se limiter à l'exploiter ou même remplacer le contenu par un autre message qu'il désire.

2.4.2 La sécurité stéganographique

2.4.2.1 Formalisation théorique de la sécurité

La notion de sécurité ou d'indétectabilité d'un schéma de stéganographie est définie par sa résistance face à une attaque. Elle mesure la capacité de l'attaquant (la gardienne Eve) à détecter la présence d'un message secret dans un médium de couverture. Pour définir cette sécurité Cachin [47] a utilisé la distance de Kullbak-Liebler $D_{KL}(P_{\mathcal{X}}||P_{\mathcal{Y}})$ qui représente l'entropie relative de la distribution $P_{\mathcal{X}}$ de l'ensemble des médiums de couverture $\mathcal{X}$ par rapport à celle de $P_{\mathcal{Y}}$ de l'ensemble des stégo médiums $\mathcal{Y}$:

$$D_{KL}(P_{\mathcal{X}}||P_{\mathcal{Y}}) = \sum_{x\in\mathcal{X}} P_{\mathcal{X}}(x)\log_2\frac{P_{\mathcal{X}}(x)}{P_{\mathcal{Y}}(x)} \tag{2.15}$$

Avec cette définition, la sécurité d'un schéma stéganographique dépend de l'incapacité de l'adversaire à distinguer entre les deux distributions $P_{\mathcal{X}}$ et $P_{\mathcal{Y}}$. Ainsi, un schéma de stéganographie est considéré comme étant parfaitement sûr si $D_{KL}(P_{\mathcal{X}}||P_{\mathcal{Y}}) = 0$, et comme ε-sûr si $D_{KL}(P_{\mathcal{X}}||P_{\mathcal{Y}}) < \varepsilon$. Plus ε est grand, plus la probabilité qu'un message secret soit détecté est grande. Les modèles de sécurité actuelles sont définis en utilisant les outils d'apprentissage et de classification. Le stéganalyste utilise une base d'images à partir de laquelle il extrait un ensemble de caractéristiques statistiques qui seront ensuite réparties en deux sous-ensembles d'apprentissage (supervisée en général) et de test. Le classifieur détermine la limite de décision sur un ensemble d'apprentissage et applique ensuite cette limite à l'ensemble de test. La définition et la formalisation théorique de la sécurité d'un schéma de stéganographie reste un problème ouvert.

2.4.2.2 Quelques règles pratiques pour la sécurité stéganographique

En pratique, étant donné qu'il n'existe pas une formalisation précise de la notion de sécurité, il est difficile pour un stéganographe de garantir la sécurité de son schéma face à toutes les attaques existantes. Cependant, il est possible de garantir le minimum de sécurité face aux attaques triviales en respectant quelques règles de base. Tout d'abord, il faut s'assurer que le médium de couverture n'est utilisé qu'une seule fois, afin d'éviter toutes les attaques par différence pour lesquelles la probabilité de détection sera égale à 1. Ensuite, il faut vérifier que

la taille de la clé est suffisamment grande pour se prémunir contre les attaques exhaustives sur la clé stéganographique. En outre, on peut contourner les attaques visuelles en modifiant le médium de couverture de sorte que les modifications ne soient pas visibles à l'œil nu. Il est aussi important de préserver au mieux la distribution et la statistique des éléments de couverture à modifier.

2.4.3 Les différents types de stéganalyse

En stéganalyse, plusieurs scénarii existent et définissent un certain nombre de règles et d'hypothèses en fonction des informations dont le stéganalyste dispose sur l'algorithme d'insertion utilisé.

2.4.3.1 Stéganalyse ciblée (targeted steganalysis)

Elle exploite les faiblesses de l'algorithme d'insertion pour lequel elle est destinée. Une telle attaque ne peut souvent pas être étendue à d'autres algorithmes d'insertion. De nombreuses attaques ciblées ont été décrites sur les méthodes d'insertion par LSB dans les deux domaines spatial [48] et JPEG [49].

2.4.3.2 Stéganalyse aveugle (blind) ou non ciblée (universelle)

Plus généraliste que la stéganalyse ciblée, elle ressemble plus à la notion empirique de la stéganalyse car elle représente les objets de couverture et stégo dans un espace de caractéristiques dans lequel ils peuvent être séparés par un algorithme d'apprentissage supervisé. Lorsque le stéganalyste connaît la taille du message, le problème de test d'hypothèses est implémenté en utilisant des classifieurs binaires tels que le FLD (Fisher Linear Discriminant) ou à l'aide de SVM (Support Vector Machine) avec un noyau linéaire ou gaussien. Si aucune information sur le message n'est disponible, Eve a plusieurs options parmi lesquels elle peut essayer d'estimer le message en utilisant la stéganalyse quantitative.

2.4.3.3 Stéganalyse quantitative

Cette classe de stéganalyse estime la taille du message. Les méthodes peuvent être soit ciblées comme pour Jsteg [50] ou basées sur des caractéristiques [51]. Les méthodes basées caractéristiques effectuent souvent une régression statistique pour l'apprentissage de la correspondance entre l'espace des caractéristiques et la taille du message permettant de réutiliser les correspondances de caractéristiques développées pour la stéganalyse aveugle.

2.4.4 Construction des algorithmes de stéganalyse

Généralement, un algorithme de stéganalyse peut être construit en utilisant soit une détection statistique ou une méthode d'apprentissage machine (machine learning). Les deux approches ont des avantages ainsi que des limites et continuent de coexister. La première approche détermine le détecteur à partir d'un modèle statistique de la source de couverture. Elle s'applique aux algorithmes d'insertion simples comme la substitution et correspondance de LSB [52], [53] et peuvent ne pas être facilement adaptés aux algorithmes d'insertion adaptatifs. La stéganalyse de ces schémas nécessite des modèles permettant des dépendances complexes entre les pixels voisins. La deuxième approche formalise la détection comme un problème de classification. Tout d'abord, l'image est représentée par un vecteur de caractéristiques, qui peut être considéré comme une réduction de la dimensionnalité heuristique. Ensuite, une base d'images de couverture et stégo est utilisée pour construire le détecteur en utilisant les outils d'apprentissage supervisé. Le principal avantage de cette approche est que l'on peut facilement construire des détecteurs pour les algorithmes d'insertion arbitraires. En outre, pour un médium de couverture connu, ces détecteurs fonctionnent habituellement nettement mieux que ceux dérivés de modèles de couverture simples.

Les résultats des tests peuvent être utilisés pour dresser la matrice de confusion ainsi que la courbe de ROC (Receiver Operating Characteristic) obtenue en traçant $1 - P_{MD}(P_{FA})$ comme une fonction de P_{MD} en fonction P_{FA}, où P_{FA} et P_{MD} sont les probabilités de fausse-alarme (faux positif) et d'échec de détection (faux négatif). La courbe de ROC peut être réduite à une mesure de détection scalaire. Les classifieurs des détecteurs doivent ainsi être implémentés de telle sorte à minimiser la probabilité d'erreur de classification totale

$$P_E = min_{P_{FA}} \frac{1}{2}(P_{FA} + P_{MD}), \tag{2.16}$$

La sécurité peut être évaluée en mesurant la valeur moyenne de P_E obtenu sur l'ensemble de test de la base d'images.

2.5 Méthodes adaptatives de calcul de fonction de distorsion

On peut classer les algorithmes de stéganographie d'images suivant deux critères. Le premier critère concerne le domaine d'insertion et les informations disponibles du côté de l'émetteur : spatial, transformée (JPEG) et side-informed JPEG. Dans le domaine spatial les

messages sont insérés en modifiant les pixels de l'image de couverture alors que dans le domaine JPEG, ce sont les coefficients DCT (Discrete Cosines Transform) quantifiés qui sont modifiés. Les algorithmes side-informed utilisent les connaissances sur les coefficients DCT non-quantifiés de l'image avant compression. Nous nous limiterons dans ce manuscrit aux méthodes d'insertion dans le domaine spatial.

Un autre critère de classification existe et se base sur les objectifs des algorithmes d'insertions. Ainsi, on distingue principalement deux approches : les méthodes basées sur la préservation de modèles statistiques [54] qui visent la conservation de la distribution des vecteurs caractéristiques de l'image de couverture et les méthodes adaptatives qui ont pour but de minimiser la fonction de distorsion entre l'image de couverture et celle stégo. Cette fonction de distorsion est définie en attribuant à chaque élément de couverture un coût de détectabilité ρ_i, avec $i \in \{1, \ldots, n\}$, modélisant l'impact de la modification sur la sécurité. La carte de détectabilité $\rho = \{\rho_i\}$ est généralement associée à une fonction de distorsion $D(\mathrm{x}, \mathrm{y})$ qui est minimisée sous la contrainte d'un payload fixe. Le problème est de trouver le moyen le plus efficace permettant de calculer les coûts ρ_i qui reflètent le mieux la détectabilité statistique lors de la modification. La stéganographie adaptative au contenu dissimule le message secret dans les zones de l'image qui sont les plus difficilement détectables (par exemple les zones texturées et les zones bruitées comme les contours). Par conséquent, avant d'insérer un message, il est d'abord nécessaire de sélectionner les zones les plus sûres de l'image. Une fois les coûts de distorsion déterminés, l'étape suivante consiste à utiliser une méthode d'insertion efficace et optimale. Pour ce faire la plupart des méthodes actuelles utilisent le STC pour approcher la limite théorique. Nous allons proposer un nouveau paradigme de codage permettant approcher d'avantage cette limite théorique. L'avantage des méthodes adaptatives par minimisation d'impact d'insertion, par rapport aux méthodes par préservation de modèles, réside dans le fait qu'elles sont plus généralistes et même plus sûres.

2.5.1 L'algorithme F5 et nsF5

L'algorithme F5 [4] est basé sur le LSB Matching et insère les bits du message secret au niveau des coefficients AC DCT non-nuls de l'image JPEG. Bien que basé dans le domaine JPEG, nous le citons ici car c'est le premier algorithme implémentant la technique de matrix embedding, présentée dans la Section 2.2.4, combinée avec les codes de Hamming. L'algorithme F5 réduit le nombre total de modifications en considérant que tous les éléments de couverture ont le même risque de détectabilité lorsqu'ils sont modifiés $\rho_i = 1$, pour tout i.

L'algorithme nsF5 (non-shrinkage F5) [55] est une version améliorée permettant de corriger les faiblesses de l'algorithme F5. Il utilise la technique de papier mouillé, avec des coûts de détectabilité à deux valeurs $\rho_i \in \{1, \infty\}$. Pour verrouiller les zones sensibles à la modification, on leur affecte un coût de détectabilité $\rho_i = \infty$ et les autres auront $\rho_i = 1$.

2.5.2 HUGO et HUGO-BD

L'algorithme HUGO (Highly Undetectable steGO) [13], proposé lors de la compétition BOSS, a joué un rôle très déterminant dans le développement des algorithmes de dissimulation adaptatifs actuels. Les coûts de modification sont calculés à partir de la différence des vecteurs de caractéristiques des quatre pixels voisins extraits de l'image de couverture et sa version stégo dans l'espace de caractéristiques SPAM (Subtractive Pixel Adjacency Matrix). Les pixels qui, après modification, changent le plus le vecteur de caractéristique, seront affectés d'un coût de détectabilité élevé. Ainsi, HUGO porte les modifications dans les zones texturées et les contours. Une modification d'HUGO appelée HUGO BD (Bounding Distortion) [43] définit un coût $\rho_i \in \{0, \infty\}$ paramétré et utilise la construction de Gibbs avec une insertion ternaire pour minimiser la fonction de distorsion non-additive bornée.

2.5.3 Méthodes basées sur transformées en ondelettes WOW et UNIWARD

Les algorithmes WOW (Wavelet Obtained Weights) [15] et UNIWARD (UNIversal WAvelet Relative Distortion) [16] fonctionnent dans le domaine d'ondelettes et utilisent des filtres directionnels pour calculer les coûts de distorsion des pixels de l'image. WOW utilise une banque de filtres passe-haut directionnels pour obtenir les résidus directionnels (liés à la prévisibilité du pixel dans une direction donnée) qui seront agrégés pour déterminer l'impact d'insertion des différents pixels de l'image de couverture. Il assigne un coût élevé ($\rho_i = \infty$) aux pixels prévisibles dans au moins une direction par des filtres directionnels (zones lisses et contours) et un coût faible aux pixels imprévisibles dans toutes les directions (zones texturées ou bruitées). L'algorithme ainsi créé résiste mieux à la stéganalyse utilisant des modèles riches SRM (Spatial Rich Model) [56] par rapport à HUGO. S-UNIWARD est une variante spatiale de la famille des distorsions UNIWARD, similaire à WOW, qui exprime la distorsion comme des modifications relatives entre les images de couverture et stégo. La directivité concentre les modifications dans les parties de l'objet de couverture qui sont difficiles à modéliser dans plusieurs directions. Sa fonction coût peut être étendue aux domaines JPEG (J-UNIWARD et SI-UNIWARD).

2.5.4 Méthode utilisant un oracle ASO

Une autre approche différente pour le calcul des coûts de modification (non paramétrique) appelée ASO (Adaptive Steganography by Oracle) [14], [57], est proposée et utilise l'ensemble de classifieurs FLD [58] comme oracle. Son originalité est qu'elle prend en compte non seulement le modèle de distribution de l'image de couverture mais également le modèle de toute la base d'images utilisée par l'émetteur.

2.5.5 Méthode utilisant des filtres passe haut et passe bas HILL

Une nouvelle fonction de coûts HILL (HIgh Low Low) [17] est proposée pour assurer que tous les pixels dans les zones texturées aient des coûts relativement faibles ce qui n'était pas le cas avec WOW et S-UNIWARD. La nouvelle fonction de coût est obtenue en utilisant un filtre passe-haut pour localiser les parties les moins prévisibles de l'image et deux filtres passe-bas permettant aux faibles valeurs de coûts d'être plus concentrées. HILL est une modification de l'algorithme de WOW dans laquelle les trois noyaux directionnels de Daubechies sont remplacés par un noyau KB (Ker-Böhme) [58] (filtrage passe-haut). Le résiduel KB est, en plus, filtré avec un filtre passe bas moyenneur 3×3. Les coûts résultants sont à nouveau filtrés passe-bas avec un noyau assez grand 15×15. Le filtrage passe-bas des coûts améliore la sécurité empirique parce que les coûts sont rendus plus uniformes. HILL a de meilleures performances par rapport à HUGO, WOW et S-UNIWARD en termes de résistance à la stéganalyse avec SRM [56].

2.5.6 Méthodes basées modèle MVGG et MiPOD

MVGG (MultiVariate Generalized Gaussian) [59] est une méthode basée sur la préservation de modèles statistiques qui permet aux modifications dans les zones fortement texturées de porter sur plus d'un bit (amplitude 1 ou 2). D'abord, l'expéditeur estime les paramètres du modèle de couverture (variances locales des pixels) comme variables aléatoires gaussiennes indépendantes mais non identiquement distribuées avec des variances différentes (MultiVariate Gaussian ou MVG). Ensuite, les coûts de détectabilité des pixels sont calculés en minimisant la détectabilité statistique exprimée sous la forme de la divergence Kullback-Leibler (KL) de la distribution de couverture par rapport à celle du stégo dans la limite asymptotique d'un petit payload. Ainsi, les coûts dépendent non seulement de la couverture mais aussi du message. Contrairement à MVGG qui minimise la divergence KL, MiPOD (Minimizing the Power of Optimal Detector) [18] minimise la puissance du détecteur le plus performant. Ce résultat est

obtenu sans l'hypothèse supplémentaire d'un petit payload. MiPOD permet également de considérer différents types de gardiens et fournit la plus petite détectabilité empirique. La distorsion liée à la détectabilité "detectability-limited sender" adapte le payload pour une couverture donnée de telle sorte que l'insertion n'excède pas un niveau de détectabilité imposé.

2.5.7 Synchronisation des modifications Synch-A

Pour définir une fonction de distorsion non-additive l'approche Synch [60] est utilisée. Elle commence par affecter un schéma additif puis construit à partir de celui-ci une fonction de distorsion non-additive simple qui impose une synchronisation des modifications adjacentes. La construction de Gibbs peut être utilisée pour réaliser l'insertion dans la pratique. L'avantage de la synchronisation des modifications est lié au fait que les détecteurs de la stéganalyse moderne utilisent des statistiques d'ordre supérieur et à la difficulté d'estimer avec précision la sélection de canal d'un schéma d'insertion non-additive. Les deux diminuent la précision des détecteurs basés SRM. Le système de synchronisation, appliqué aux coûts de MVG et de HILL, améliore la sécurité empirique lorsqu'il est testé avec des détecteurs basés SRM [56].

2.6 Schémas de stéganographie avec les codes correcteurs d'erreurs

Depuis, l'introduction de la technique de matrix embedding utilisant les codes correcteurs d'erreurs en stéganographie, plusieurs codes sont proposés dans la littérature parmi lesquelles nous ne citerons que les codes de Hamming [4], BCH [6], [7], RS [8], STC [12] et LDPC [11].

2.6.1 Schéma de stéganographie basé sur les codes de Hamming

La première implémentation de la technique du matrix embedding a vu le jour avec le travail de Westfield. Ce dernier a proposé un schéma de stéganographie (algorithme F5) basé les codes de Hamming [4].

Les codes de Hamming binaires sont des codes linéaires $C(n = 2^p - 1, k = 2^p - 1 - p)$ avec une matrice de contrôle de parité H de taille $(n-k)\times n = p\times(2^p-1)$ dont les colonnes sont les représentations binaires des nombres $1, \cdots, 2^p - 1 = n$ (voir chapitre 1).

Exemple 2.2 :

Une matrice de contrôle de parité H pour $p = 3$ est la suivante :

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad (2.17)$$

Soit x le vecteur de couverture devant contenir le message m.

Tout syndrome $s = m - xH^T \in \mathbb{F}_2^3$ est soit égal à $0_{\mathbb{F}_2^3}$ soit égal à une colonne de la matrice de contrôle de parité H. Dans le premier cas ($s = (0,0,0)$), aucune modification du vecteur de couverture n'est nécessaire car le vecteur stégo coïncide avec le vecteur de couverture et l'extraction du message, à la réception, se fera simplement par le produit matriciel $xH^T = m$. Dans le second cas, si nous définissons $dec(s)$ comme étant la représentation décimale de s, le vecteur d'erreur $e = (0,\cdots,0,1,0,\cdots,0)$, avec 1 à la $dec(s) - ème$ position, est le vecteur de poids minimal (leader du coset $\mathcal{C}(s)$) car $eH^T = s$. De façon générale, la position du 1 est déterminée par celle du syndrome s au niveau des colonnes de H. Ces remarques peuvent être généralisées pour tout $p \in \mathbb{N}$. Par conséquent le syndrome $s = m - xH^T$ est égale à :

- ✓ $0_{\mathbb{F}_2^p}$ avec une probabilité de $\frac{1}{2^p} \mapsto$ pas de modification (0) du support ;
- ✓ une colonne de H avec une probabilité de $\frac{2^p-1}{2^p} \mapsto$ une modification (1) du support.

Nous pouvons, à partir de là, calculer l'efficacité d'insertion du schéma de stéganographie par codage de Hamming.

Nous insérons p bits de message dans un support de taille $n = 2^p - 1$. Ainsi, nous faisons en moyenne :

$nbre_{modif}(p) = 0 * \frac{1}{2^p} + 1 * \frac{2^p-1}{2^p} = 1 - 2^{-p}$ modifications par bloc pendant l'insertion.

Ainsi l'efficacité d'insertion est :

$$eft_{inst}(p) = \frac{p}{1 - 2^{-p}}. \tag{2.18}$$

La charge relative (ou taille relative du message) est :

$$charge_{relative}(p) = \frac{p}{2^p - 1}. \tag{2.19}$$

Exemple 2.3 :

Considérons le vecteur de couverture $x = (1,0,1,0,1,1,0)$ et le message $m = (1,0,1)$. Le calcul du syndrome $s = m - xH^T = (1,0,1) - (0,0,1) = (1,0,0)$. Ce qui est égal à la transposée de la quatrième colonne de H donnée par (2.17). Ainsi le vecteur d'erreur sera donné par $e = (0,0,0,1,0,0,0)$ et le vecteur stégo par $y = (1,0,1,1,1,1,0)$.

2.6.2 Schéma de stéganographie utilisant les codes BCH

Après les codes de Hamming, les codes BCH ont été introduits en stéganographie par Schönfeld et Winkler [6] en vue d'améliorer l'efficacité d'insertion. Ils ont proposé deux manières de calcul de syndrome. Une première approche basée sur la recherche d'un leader de coset en utilisant une matrice de contrôle de parité H et une seconde approche, moins

complexe, utilisant un polynôme générateur $g(\theta)$ pour la recherche des racines. L'utilisation des tables de correspondance par Zhang et al. [7] a permis de réduire la complexité temporelle comparée à l'approche basée sur la recherche exhaustive des racines [6].

Le décodage par syndrome avec les codes BCH

Les codes BCH (n, k, t) peuvent corriger jusqu'à t erreurs, où n est la longueur des mots de code et k est la dimension du code. Un bloc de données binaires, par exemple les valeurs des LSBs du médium de couverture $\{x_0, x_1, \cdots, x_{n-1}\}$ sur $\mathbb{F}_2$, peut être représenté par un polynôme de θ sur $\mathbb{F}_{2^p}$ tel que $\mathrm{x}(\theta) = x_0 + x_1\theta + x_2\theta^2 + \cdots + x_{n-1}\theta^{n-1}$. L'insertion du message m dans le vecteur de couverture x produit un vecteur stégo y qui est représenté par $\mathrm{y}(\theta) = y_0 + y_1\theta + y_2\theta^2 + \cdots + y_{n-1}\theta^{n-1}$. Soit e le vecteur représentant les positions des bits modifiés et sa représentation polynômiale $\mathrm{e}(\theta) = e_0 + e_1\theta + e_2\theta^2 + \cdots + e_{n-1}\theta^{n-1}$. Les relations entre les vecteurs de couverture x, d'erreurs e et stégo y sont :

$$\mathrm{m} = \mathrm{y}\mathrm{H}^{\mathrm{T}} \tag{2.20}$$

$$\mathrm{y}(\theta) = \mathrm{x}(\theta) + \mathrm{e}(\theta) \;\Leftrightarrow\; \mathrm{y} = \mathrm{x} + \mathrm{e}$$

$$\mathrm{s} = \mathrm{m} - \mathrm{x}\mathrm{H}^{\mathrm{T}} = \mathrm{e}\mathrm{H}^{\mathrm{T}} \tag{2.21}$$

Soit $C(n, k, t)$ un code BCH primitif sur le corps de Galois $\mathbb{F}_{2^p}$ (la longueur $n = 2^p - 1$) et la distance construite $\delta = 2t + 1$ (on peut corriger jusqu'à t erreurs). Soit α une racine primitive $n - ième$ de l'unité. Nous avons les racines $\alpha, \alpha^2, \ldots, \alpha^{\delta-1}$.

Considérons que le polynôme $e(\theta)$ a ν coefficients non nuls (modifications) se trouvant aux niveau des positions $(j_1, j_2, \cdots, j_\nu)$ où les j_i sont les indices des coefficients à modifier pour cacher le message m avec $0 \leq j_1 < j_2 < \cdots < j_\nu \leq n$. Le polynôme $\mathrm{e}(\theta)$ est alors :

$$\mathrm{e}(\theta) = \theta^{j_1} + \theta^{j_2} + \cdots + \theta^{j_\nu}. \tag{2.22}$$

Les coefficients du syndrome s de e sont donnés par :

$$s_l = e(\alpha^l) = (\alpha^{j_1})^l + (\alpha^{j_2})^l + \cdots + (\alpha^{j_\nu})^l,\ l = 1, \ldots, 2t = \delta - 1. \tag{2.23}$$

Le polynôme localisateur d'erreurs s'exprime par [7] :

$$\sigma(\theta) = (\theta + \beta_1)(\theta + \beta_2)\cdots(\theta + \beta_\nu) = \theta^\nu + \textstyle\sum_{j=1}^{\nu} \sigma_j \theta^{\nu-j} \tag{2.24}$$

avec $\beta_i = \alpha^{j_i}$. Après développement et identification de $\sigma(\theta)$ on a :

$$\begin{aligned}\sigma_1 &= \beta_1 + \beta_2 + \cdots + \beta_\nu \\ \sigma_2 &= \beta_1\beta_2 + \beta_2\beta_3 + \cdots + \beta_{\nu-1}\beta_\nu \\ &\vdots \\ \sigma_\nu &= \beta_1\beta_2\cdots\beta_{\nu-1} + \beta_2\beta_3\cdots\beta_\nu \\ \sigma_\nu &= \beta_1\beta_2\cdots\beta_\nu\end{aligned} \quad . \tag{2.25}$$

D'après ces égalités, la connaissance des coefficients $\sigma_1, \sigma_2, \cdots, \sigma_\nu$ du polynôme $\sigma(\theta)$ permet de calculer ses racines $\beta_1, \beta_2, \cdots, \beta_\nu$ et les positions à modifier j_i à partir de la relation $\beta_i = \alpha^{j_i}$. Mais le calcul de ces racines n'est pas facile.

Un algorithme utilisant des tables de correspondance [7], avec $t = 2$, est proposé pour calculer les racines quadratiques et cubiques du polynôme localisateur d'erreurs $\sigma(\theta)$.

Les tables de correspondances (Look-up tables)

Après le calcul du syndrome et des coefficients du polynôme localisateur d'erreurs $\sigma(\theta)$ (de degré égal au nombre d'erreurs), l'étape suivante consiste à chercher les racines de $\sigma(\theta)$ et chaque racine nous donne la position d'une erreur. Le calcul de syndrome et des racines du polynôme localisateur d'erreurs présente une grande complexité temporelle. Grâce aux tables de correspondance, on peut réduire le temps de calcul des racines du polynôme localisateur d'erreurs $\sigma(\theta)$ comme le montre Zhang et al. [7]. Une table de correspondance est une structure de données utilisée pour remplacer un calcul par une opération de consultation qui, souvent est plus simple. Le gain en vitesse peut être significatif, car rechercher une valeur en mémoire est souvent plus rapide qu'effectuer un calcul important.

La capacité de correction des codes BCH en stéganographie t est fixée à 2, d'où une capacité d'insertion de $n - k = 2t = 4$. Cette valeur est supérieure à celle des codes de Hamming mais reste toujours faible. Ce qui limite leur utilisation à des médiums de couverture de taille n petite. Une amélioration est donc nécessaire.

2.6.3 Application des codes de Reed-Solomon en stéganographie

L'utilisation des codes de Reed-Solomon dans la stéganographie est rendue possible grâce au travail de Galand et Fontaine [8]. La version des codes GRS (chapitre 1) est utilisée pour définir un schéma de stéganographie à papier mouillé permettant le verrouillage des bits au niveau desquels les modifications seraient plus perceptibles pour les porter sur les bits restants du support.

Schéma de stéganographie utilisant les codes GRS

Considérons un vecteur x de n symboles de $\mathbb{F}_q$, extrait du vecteur de couverture y, et un message m. L'objectif est de transformer x en y tel que celui-ci contienne le message m en changeant au plus ℓ composantes de x. Cela se traduit avec les fonctions Emb et Ext par :

$$\forall(\mathrm{x},\mathrm{m}) \in \mathbb{F}_q^n \times \mathbb{F}_q^m, \quad Ext\big(Emb(\mathrm{x},\mathrm{m})\big) = \mathrm{m}, \tag{2.26}$$

$$\forall(\mathrm{x},\mathrm{m}) \in \mathbb{F}_q^n \times F_q^m, \quad d_H\big(\mathrm{x}, Emb(\mathrm{x},\mathrm{m})\big) \leq \ell. \tag{2.27}$$

Les fonctions sont définies comme suit :

$$\begin{gathered} Emb(\mathrm{x},\mathrm{m}) = \mathrm{x} + \mathrm{e} = \mathrm{y} \\ Ext(\mathrm{y}) = \mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{m} \end{gathered} \tag{2.28}$$

La technique du papier mouillé permet de verrouiller certaines positions qui ne doivent pas être modifiées. Ainsi le symbole e_i du vecteur d'erreurs e, $i \in \{0,\cdots,n-1\}$, est non nul pour $y_i \neq x_i$ (avec modification) et nul si $y_i = x_i$ (sans modification).

Soient $\gamma_0,\ldots,\gamma_{n-1} \in \mathbb{F}_q$ fixés, on construit la matrice Γ dont la $i-$ème ligne est égale à $ev(\theta^i)$ [8].

$$\Gamma = \begin{pmatrix} ev(\theta^0) \\ ev(\theta^1) \\ \vdots \\ ev(\theta^{n-1}) \end{pmatrix} = \begin{pmatrix} \gamma_0^0 & \gamma_1^0 & \cdots & \gamma_{n-1}^0 \\ \gamma_0^1 & \gamma_1^1 & \cdots & \gamma_{n-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_0^{n-1} & \gamma_1^{n-1} & \cdots & \gamma_{n-1}^{n-1} \end{pmatrix} \tag{2.29}$$

Soit $Coeff(X)$ le vecteur composé des coefficients de X dans $\mathbb{F}_q$. On peut vérifier que $Coeff(X) \cdot \Gamma = ev(X)$. La matrice Γ est inversible et son inverse Γ^{-1} peut être subdivisé en deux blocs [8] :

$$\Gamma^{-1} = \Big(\underbrace{A}_{k\ colonnes} \quad \underbrace{B}_{n-k\ colonnes} \Big). \tag{2.30}$$

Les $n-k$ dernières colonnes de Γ^{-1} (la matrice B) sont choisies comme étant celles de la transposée de la matrice de contrôle de parité (soit H^{T}) d'un code GRS.

Si L est égal au rayon couverture R du code ayant comme matrice de contrôle de parité H, un tel vecteur e existe toujours mais son calcul est très complexe.

Les codes GRS permettent de résoudre ce problème car la recherche d'un vecteur de syndrome $\mathrm{m} = (m_0,\ldots,m_{n-1-k})$ est aisée. Considérons le polynôme $M(\theta)$ qui a comme coefficient m_i pour le monôme θ^{k+i}, $i \in \{0,\ldots,n-1-k\}$. Dans ce cas, $ev(M) \cdot \mathrm{H}^{\mathrm{T}} = \mathrm{m}$. Soit X le polynôme représenté par $ev(X) = \mathrm{x}$ et P un polynôme de degré inférieur à k tel que $P(\gamma) = M(\gamma) - X(\gamma)$ pour au moins k composantes γ dans $\{\gamma_0,\ldots,\gamma_{n-1}\}$. Avec un tel P, le vecteur $\mathrm{e} = ev(M-X-P)$ a au moins k coordonnées égales à zéro. Puisque M et X sont connus, la recherche de e se résumera au calcul de P.

Le vecteur stégo est donné par :

$$\mathrm{y} = \mathrm{x} + \mathrm{e} = ev(X) + ev(M - X - P) = ev(M - P). \quad (2.31)$$

Soit une position $i \in \mathcal{J}$ verrouillée alors nous avons :

$$\begin{aligned} y_i &= ev(M-P)_i = M(\gamma_i) - P(\gamma_i) \\ &= X(\gamma_i) = ev(X)_i = x_i \end{aligned} \quad (2.32)$$

- ***Construction de P en utilisant l'interpolation de Lagrange***

Un moyen très simple pour construire P est l'interpolation de Lagrange [8]. Considérons un ensemble de k positions à verrouiller $\mathcal{J} = \{i_1, \ldots, i_k\}$ et calculons

$$P(\theta) = \sum_{i \in \mathcal{J}} (M(\gamma_i) - X(\gamma_i)) \cdot L_{\mathcal{J}}^{(i)}(\theta), \quad (2.33)$$

où $L_{\mathcal{J}}^{(i)}$ est l'unique polynôme de degré au plus $k-1$ définie par :

$$L_{\mathcal{J}}^{(i)}(\gamma_j) = \begin{cases} 0 & si\ j \neq i \\ 1 & si\ j = i \end{cases} \quad (2.34)$$

Ce polynôme peut être défini à l'aide de l'interpolation de Lagrange :

$$L_{\mathcal{J}}^{(i)}(\theta) = \prod_{j \in \mathcal{J} \setminus \{i\}} \frac{(\theta - \gamma_j)}{(\gamma_i - \gamma_j)}. \quad (2.35)$$

Le polynôme P, que nous obtenons, vérifiera bien les conditions $P(\gamma_i) = M(\gamma_i) - X(\gamma_i)$, pour tout $i \in \mathcal{J}$. On peut donc poser $\mathrm{e} = ev(M - X - P)$. Par conséquent, pour $i \in \mathcal{J}$, on a $e_i = 0$ ($y_i = x_i$), ainsi toutes les positions dans $\mathcal{J}$ sont verrouillées.

Cette solution proposée permet de garder inchangées les coordonnées dont les modifications sont les plus détectables.

- ***Construction de P par l'algorithme de Guruswami-Sudan***

Si le nombre de positions verrouillées est inférieur à k, l'interpolation de Lagrange n'est pas optimale car elle change $n - k$ positions, presque toujours.

Un moyen pour résoudre ce problème consiste à utiliser un algorithme de décodage afin de construire P; on essaie de décoder $ev(M - X)$. Les positions verrouillées peuvent être traitées en utilisant le décodage par syndrome avec la matrice de contrôle de parité $\mathrm{H}_{\mathcal{J}}$ (obtenue en élaguant les colonnes de H d'indice dans $\mathcal{J}$). Si le processus réussit, on obtient un P dans la boule centrée en $ev(M - X)$ et de rayon λ, où λ est le rayon de décodage de l'algorithme de décodage. Ici, l'algorithme de Guruswami-Sudan [8] offre une grande valeur de λ, c'est-à-dire une grande probabilité de réussite, et retourne une liste de polynômes dans laquelle on choisit le meilleur P respectant les contraintes. En cas d'échec du décodage, on ajoute une nouvelle

position dans $\mathcal{J}$ et on reprend le processus. Si on arrive à k positions verrouillées, on utilise l'interpolation de Lagrange. L'algorithme permet d'insérer $m = n - k$ symboles de $\mathbb{F}_q$ avec un maximum de k positions à verrouiller et au plus $n - k$ modifications.

Une autre méthode plus générale est proposée utilisant la méthode de calcul du syndrome prolongé [61]. Ce qui a permis une détermination plus efficace du vecteur de changement e.

2.6.4 Schéma de stéganographie avec les codes STC

En supposant que les modifications n'interagissent pas entre elles, l'impact total de l'insertion d'un message dans un objet de couverture est la somme de l'impact d'insertion au niveau de chaque pixel [12]. La distorsion totale est donnée par :

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \rho_i |x_i - y_i| \tag{2.36}$$

avec $0 \leq \rho_i \leq \infty$ le coût d'une modification du pixel x_i en y_i. L'objectif consiste, du côté de l'expéditeur, à insérer son message binaire $\mathrm{m} \in \{0,1\}^m$ de telle sorte que le taux de distorsion D soit minimal. La minimisation de la distorsion a conduit à des stégo-systèmes plus sûrs. Redéfinissons les deux fonctions d'insertion $Emb: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$ et d'extraction $Ext: \{0,1\}^n \rightarrow \{0,1\}^m$ en considérant un code $C(n, n-m)$ en bloc linéaire.

$$Emb(\mathrm{x},\mathrm{m}) = arg \min_{\mathrm{y} \in \mathcal{C}(\mathrm{m})} D(\mathrm{x},\mathrm{y}) \tag{2.37}$$

$$Ext(\mathrm{y}) = \mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{m} \tag{2.38}$$

où $\mathrm{H} \in \{0,1\}^{m \times n}$ est une matrice de contrôle de parité du code C, $\mathcal{C}(\mathrm{m}) = \{\mathrm{z} \in \{0,1\}^n \mid \mathrm{z}\mathrm{H}^{\mathrm{T}} = \mathrm{m}\}$ est le coset correspondant au syndrome m. Le problème résidera dans le calcul d'un quantificateur efficace du coset (2.38).

Le Code de Syndrome en Treillis

Les solutions du système $\mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{m}$ sont représentées par des chemins dans un treillis. Le vecteur y le plus proche du vecteur de couverture x est trouvé en utilisant l'algorithme de Viterbi [12]. La matrice H est écrite sous une forme particulière ; elle est formée en plaçant des sous-matrices $\widehat{\mathrm{H}}$, de taille $h \times w$, côte à côte et décalées d'une ligne vers le bas (Figure 2.5) [12]. Ce qui conduit à une matrice creuse bornée (les autres éléments de H sont nuls). La hauteur h de la sous-matrice est un paramètre de conception qui affecte la vitesse et l'efficacité de l'algorithme (dans la pratique $6 \leq h \leq 15$). La largeur de $\widehat{\mathrm{H}}$ dépend de la valeur de la charge

relative. Si $\alpha = (n-k)/n$ est égal à $1/p$ avec $p \in \mathbb{N}^*$, on choisit $w = p$. Dans le cas général, on doit chercher k tel que $1/(p+1) < \alpha < 1/p$. La matrice H contiendra un mélange de sous-matrices de largeur p et $p+1$ de telle sorte que la matrice obtenue H soit de taille $(\alpha \cdot n) \times n$. La sous-matrice $\hat{\mathrm{H}}$ doit être connue de l'expéditeur et du récepteur. Si nous considérons que $\alpha = 1/w$, alors la matrice H est de taille $b \times (b \cdot w)$, où b est le nombre de copies de $\hat{\mathrm{H}}$ dans H. Il correspond dans ce cas à la taille m du message m.

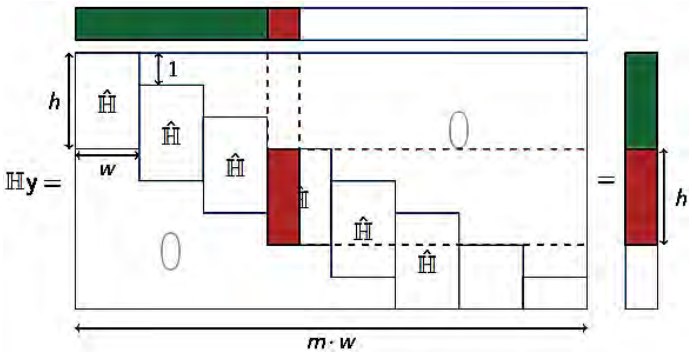

Figure 2.5 : Construction de la matrice de contrôle de parité H du STC.

Considérons la relation (2.38), en observant la forme de H sur la Figure 2.6 [12] on peut constater que seuls les w premiers bits du vecteur y interviennent dans le calcul du premier bit du message m. Par conséquent, on a besoin de trouver $(y_1, \cdots, y_w)$ tel que $(y_1, \cdots, y_w) \cdot (\mathrm{H}_{11}, \cdots, \mathrm{H}_{1w})^T = m_1$ et ainsi $(y_1, \cdots, y_w) \cdot (\hat{\mathrm{H}}_{11}, \cdots, \hat{\mathrm{H}}_{1w})^T = m_1$. De même, le second bit de m est affecté uniquement par les $2w$ premiers bits de y et ainsi de suite. En outre, aucun bit y_i du vecteur y ne peut affecter plus de h bits du message m (aucune colonne de H ne contient pas plus de h éléments non nuls). *Ces observations permettent de trouver la solution optimale de* (2.37) *avec une complexité linéaire en fonction de n.*

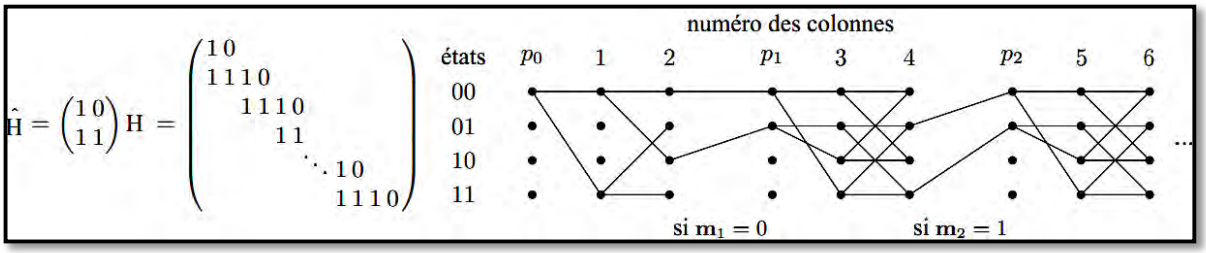

Figure 2.6 : Matrice de contrôle de parité H et son syndrome en treillis correspondant.

Chaque y *satisfaisant* $\mathrm{yH}^{\mathrm{T}} = \mathrm{m}$ *est représenté comme un chemin à travers le treillis.* Chaque chemin commence avec l'état "tous-nuls" (nœud de la première ligne et de la première colonne du treillis, qu'on appelle nœud de départ) et s'étend vers la droite. Le chemin montre le

calcul, pas à pas, du syndrome (partiel) en utilisant les bits de y. Par exemple, les deux premiers segments sur la Figure 2.6, qui relient l'état 00 de la colonne p_0 avec les états 11 et 00 de la colonne suivante (colonne 1), correspondent respectivement à un "ajout" ($y_1 = 1$) ou à un "non-ajout" ($y_1 = 0$), de la première colonne de H au syndrome. On dira qu'un état, dans le treillis, est ***accessible*** s'il existe un chemin reliant cet état au nœud de départ. A partir de chaque état accessible, deux segments s'étendent vers la droite. Ils correspondent à "l'ajout" ou "non-ajout" de la colonne suivante de la matrice H au syndrome partiel courant et sont respectivement "d'étiquette" 1 et 0. Les "étiquettes" des segments d'un chemin dans le treillis correspondent donc aux bits individuels de l'objet stégo y. En respectant ce procédé, on peut construire l'ensemble du treillis.

Pour trouver le vecteur stégo le plus proche du vecteur de couverture, on attribue des poids à tous les segments du treillis et on transforme ainsi le problème (2.37) à celui qui consiste à trouver le plus court chemin à travers le treillis. Les poids des segments entrant dans la colonne d'indice l, $l \in \{1, \ldots, n\}$, dans le treillis dépendent du $l-$ ème bit de l'objet de couverture x. Si $x_l = 0$, alors le segment horizontal (correspondant au "non-ajout" de la $l-$ ème colonne de H) a un poids de 0 et l'autre segment (correspondant à "l'ajout" de la $l-$ ème colonne de H) a un poids de ρ_l car, pour ajouter cette colonne, on aura à changer x_l de 0 à 1. Si $x_l = 1$, les rôles des segments sont inversés. Les poids correspondent au cumul des poids des segments nœud de départ. Notons que lorsque deux chemins différents aboutissent à un même nœud, celui qui a le poids plus élevé est abandonné. Tous les segments reliant les différents blocs du treillis ont un poids nul.

Algorithme de Viterbi

Même si le nombre de chemins à travers le treillis est exponentiel en n, le problème qui consiste à trouver le chemin le plus court peut efficacement être résolu par une forme de programmation dynamique appelé l'algorithme de Viterbi. Cet algorithme se compose de deux parties, la ***partie avant*** et la ***partie arrière*** [12]. La partie avant de l'algorithme comprend $n + b$ étapes. Après avoir terminé l'étape i, on connait le chemin le plus court entre l'état de départ et chaque état dans la $i-$ ème colonne du treillis (il y'à 2^h états dans chaque colonne). Ainsi, à la fin du processus (à l'étape $n + b$), on découvre le plus court chemin à travers l'ensemble du treillis. Si un état n'est pas accessible (il n'y a pas de chemin allant à cet état), on attribue le poids ∞ à ce chemin. Durant la partie arrière, le plus court chemin est remonté et l'objet stégo y le plus proche est obtenu à partir des étiquettes des segments.

Lorsqu'on termine la partie avant de l'algorithme de Viterbi (quand on atteint la fin du treillis), on revient à l'état à partir duquel on avait terminé en utilisant les segments qui n'ont pas été supprimés et on construit le stégo-objet y à partir de leurs "étiquettes".

2.6.5 Utilisation des codes LDGM et LDPC en stéganographie

Les codes à faible densité comme les codes LDGM (Low Density Generator Matrix) et les codes LDPC (Low Density Parity Check) ont été proposés en stéganographie. Ils permettent la minimisation de l'impact d'insertion des messages dans le médium de couverture.

Pour appliquer les codes LDGM dans la stéganographie, Fridrich et al. [9] ont assimilé le problème de minimisation de l'impact d'vinsertion en stéganographie à la quantification binaire généralement rencontrée dans les normes de compression. En vue d'avoir un algorithme moins complexe, ils ont utilisé deux matrices carrées qui permettent de traiter la matrice génératrice sous forme triangulaire. Ce traitement étant plus simple avec les codes LDPC, duals des codes LDGM, les codes LDPC sont exploités en stéganographie [11]. Les auteurs ont proposé une matérialisation pratique de l'approche de Filler en simplifiant certains aspects théoriques. La manipulation directe de la matrice de contrôle des codes LDPC, plus compatible au calcul de syndrome avec l'utilisation des codes correcteurs, a permis une réduction de la complexité grâce à une étape de prétraitement. La Figure 2.7 donne une représentation des différentes étapes du processus d'insertion d'un message dans un vecteur de couverture en utilisant les codes LDPC.

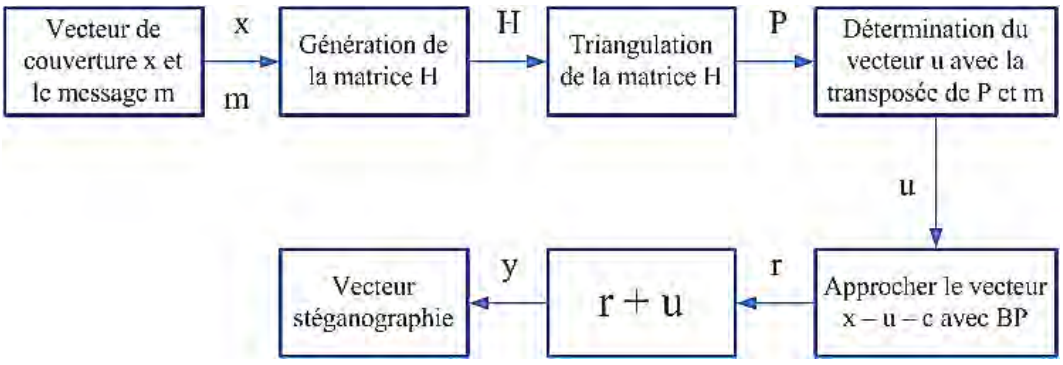

Figure 2.7 : Insertion d'un message dans un vecteur de couverture avec un code LDPC.

Cette méthode a permis la minimisation de l'impact d'insertion et offre une meilleure efficacité d'insertion en considérant que tous les éléments de couverture ont le même risque de détectabilité. Par la suite, une autre méthode adaptative est proposée dans [20].

Conclusion

Nous avons décrit dans ce chapitre les différentes techniques utilisées en stéganographie. La technique de base est celle du LSB. Les autres techniques à savoir la technique de matrice embedding et la technique du papier mouillé se basant sur la théorie des codes correcteurs d'erreurs en particulier sur le décodage. Les différents schémas proposés permettent de minimiser l'impact d'insertion. Pour produire de meilleurs schémas stéganographiques, il est possible d'utiliser soit un bon modèle soit un bon code. Nous allons nous tourner vers les codes polaires qui offrent de bonnes performances en théorie de l'information.

Chapitre 3: LES CODES POLAIRES

Introduction

Claude Shannon a montré [21], par une méthode de codage aléatoire, qu'il existe des séquences de codes permettant d'atteindre la capacité du canal. Depuis, plusieurs codes sont proposés dans l'état de l'art comme les codes LDPC [23] et les Turbo codes [24] pour approcher d'avantage cette limite. Arikan a introduit, récemment, les *codes polaires* [25] comme étant la première famille de codes qui atteignant la capacité du canal. Ces codes sont aussi optimaux pour le codage source [22]. La construction des codes polaires se base sur un phénomène appelé *polarisation de canal*. Plusieurs algorithmes de décodage sont proposés parmi lesquels nous présenterons le SC (Successive Cancellation) [25], le LP (Linear Programming) [29], ALP (Adaptive LP) [30] et SCL (SC List) [27]. Les algorithmes de décodage proposés sont soit basés LR (Likelihood Ratio) ou LLR (Log-Likelihood Ratio).

3.1 Définitions et notations usuelles

Notations utiles :

Nous allons définir les notations qui seront utilisées tout au long de ce chapitre. Soient W le canal sur lequel s'effectue la transmission ; $\mathcal{X}$ et $\mathcal{Y}$ les alphabets d'entrée et de sortie du canal W ; $W(r|c)$ les probabilités de transition telles que $c \in \mathcal{X}$ et $r \in \mathcal{Y}$; l'opérateur $\oplus$ l'addition modulo 2 $\otimes$ le produit de Kronecker défini par :

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix} \tag{3.1}$$

où A, B et $A \otimes B$ sont respectivement des matrices de tailles $m \times n$, $q \times g$ et $mq \times ng$. La puissance de Kronecker $A^{\otimes p}$ est définie par :

$$A^{\otimes p} = A \otimes A^{\otimes (p-1)} \tag{3.2}$$

pour tout $p \geq 1$, avec $A^{\otimes 0} \triangleq [1]$.

Pour les vecteurs, nous utilisons les notations suivantes : a_1^n désigne le vecteur ligne $(a_1, \ldots, a_n)$ avec $n = 2^p$, p étant un entier positif; a_i^j : le sous vecteur défini par

$(a_i, \dots, a_j)$ avec $1 \leq i \leq j \leq n$; $a_{i,e}^{j}$: le sous vecteur $(a_k : i \leq k \leq j\ ; k\ pair)$ et $a_{i,o}^{j}$: $(a_k : i \leq k \leq j\ ; k\ impair)$; soit $A \subset \{1, \dots, n\}$, a_A: le sous vecteur $(a_k : k \in A)$.

Définitions :

La capacité d'un canal de transmission discret est définie comme la valeur maximale de l'information mutuelle de ses variables aléatoires d'entrée et de sortie, par rapport à toutes les distributions de probabilité possibles. Cette information mutuelle est maximale si l'entrée est choisie suivant une distribution uniforme sur $\mathcal{X}$. La capacité symétrique (en bits/s) d'un canal B-DMC (Binary Discrete Memoryless Channel) W [25] est définie par :

$$I(W) \triangleq \sum_{r \in \mathcal{Y}} \sum_{c \in \mathcal{X}} \frac{1}{2} W(r|c) \log_2 \frac{W(r|c)}{\frac{1}{2}W(r|0) + \frac{1}{2}W(r|1)} \tag{3.3}$$

et le paramètre de Bhattacharyya (ou paramètre de fiabilité du canal) par :

$$Z(W) \triangleq \sum_{r \in \mathcal{Y}} \sqrt{W(r|0)W(r|1)}. \tag{3.4}$$

Ces deux paramètres représentent respectivement la capacité et la fiabilité du canal et constituent les expressions sur lesquelles se base le codage polaire. La capacité $I(W)$ est le plus grand débit qu'on peut atteindre pour une communication fiable à travers le canal W en utilisant ses entrées avec la même fréquence. Le paramètre $Z(W)$ constitue une limite supérieure de la décision du maximum de vraisemblance si le canal W est utilisé pour envoyer soit un 0 ou un 1.

On définit un canal symétrique par un B-DMC pour lequel il existe une permutation π de $\mathcal{Y}$ ($\pi : \mathcal{Y} \longrightarrow \mathcal{Y}$) vérifiant les deux relations suivantes $\pi^{-1} = \pi$ et $W(\pi(r)|0) = W(r|1)$. Deux types de canaux symétriques usuels sont le canal d'effacement binaire (BEC) et le canal binaire symétrique (BSC). Dans le cas où $\mathcal{Y} = \{0,1\}$ W est un BSC ; on a $W(0|0) = W(1|1)$ et $W(1|0) = W(0|1)$. En d'autres termes, la probabilité d'erreur ne dépend pas du symbole émis. Un BEC est un B-DMC tel que $W(r|0)W(r|1) = 0$ ou $W(r|0) = W(r|1)$. On écrit W^n pour désigner le canal correspondant à n utilisations de W; ainsi $W^n : \mathcal{X}^n \longrightarrow \mathcal{Y}^n$ est défini par :

$$W^n(r_1^n|c_1^n) = \prod_{i=1}^{n} W(r_i|c_i). \tag{3.5}$$

Propriétés :

Pour tout B-DMC W, nous avons [25]:

$$\log_2 \frac{2}{1+Z(W)} \leq I(W) \leq \sqrt{1-Z(W)^2} \tag{3.6}$$

$$I(W)^2 + Z(W)^2 \leq 1 \leq I(W) + Z(W) \tag{3.7}$$

Les paramètres $I(W)$ et $Z(W)$ prennent leurs valeurs dans [0,1] et de plus

$I(W) \approx 1$ est équivalent à $Z(W) \approx 0$ et (canal parfait)

$I(W) \approx 0$ équivaut à $Z(W) \approx 1$ (canal complètement bruité)

Nous pouvons facilement les vérifier avec les relations (3.6) et (3.7). *Plus $I(W)$ est grand, meilleur est le canal et inversement. Par contre le canal ayant la plus petite valeur de $Z(W)$ est le plus fiable*. Partant des deux équivalences, on peut dire que pour connaître les propriétés d'un canal B-DMC, il suffit d'étudier l'un des deux paramètres. Nous allons donc nous intéresser à $Z(W)$, plus simple et plus facile à manipuler.

L'idée principale du codage polaire est de construire à partir du canal physique W des canaux virtuels tels qu'une fraction d'entre eux tendent vers des canaux parfaits et que les restant tendent vers des canaux complètement bruités. C'est la polarisation de canal.

3.2 Polarisation de canal

La polarisation constitue le soubassement de la construction des codes polaires. Elle consiste à faire la synthèse de n copies indépendantes d'un B-DMC W donné pour construire n autres canaux $\{W_n^{(i)}: 1 \leq i \leq n\}$. La polarisation apparait dans le sens que $\{I(W_n^{(i)})\}$ tend vers 0 ou 1 suivant que $I(W_n^{(i)})$ soit plus proche de 0 ou 1 [25]. L'opération de polarisation de canal se fait en deux étapes : la combinaison de canaux et la division de canal.

3.2.1 La combinaison de canaux

Elle consiste à regrouper n copies d'un canal B-DMC W donné en un canal W_n. La combinaison pour le niveau $p = 1$ associe $n = 2 = 2^1$ copies indépendantes de $W_1 = W$ pour former le canal $W_2: \mathcal{X}^2 \longrightarrow \mathcal{Y}^2$. Les probabilités de transition sont définies comme suit :

$$W_2(r_1, r_2|u_1, u_2) = W(r_1|u_1 \oplus u_2)W(r_2|u_2) \tag{3.8}$$

Le schéma correspondant est le suivant :

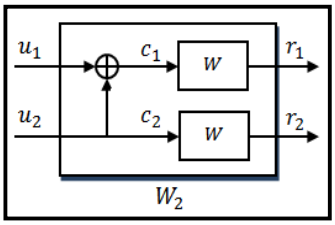

Figure 3.1 : Construction du canal W_2.

Dans ce cas on a les relations suivantes: $c_1 = u_1 \oplus u_2$ et $c_2 = u_2$. Ainsi c_1^2 et u_1^2 sont liés par la relation $c_1^2 = u_1^2 \mathrm{G}_2$ avec $\mathrm{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Le niveau suivant ($p = 2$, $n = 2^2 = 4$) de la récursivité est montré sur la Figure 3.2. Deux exemplaires indépendants du canal W_2 obtenu à partir du niveau précédent, sont combinés de la même manière pour construire $W_4 : \mathcal{X}^4 \longrightarrow \mathcal{Y}^4$ avec la probabilité de transition :

$$W_4(r_1^4|u_1^4) = W_2(r_1^2|u_1 \oplus u_2, u_3 \oplus u_4)W_2(r_3^4|u_2, u_4) \tag{3.9}$$

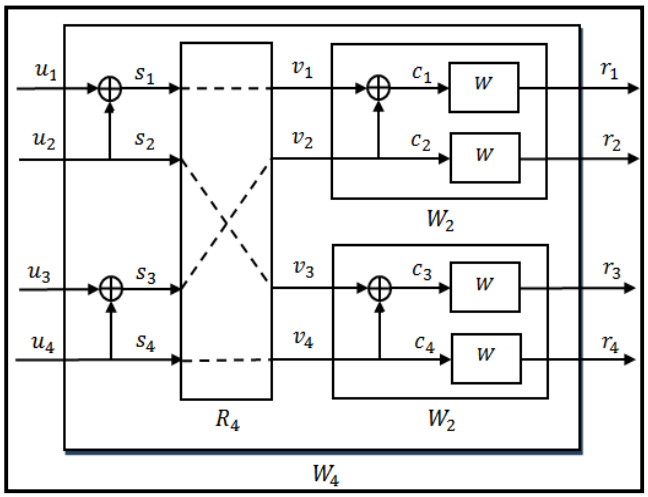

Figure 3.2 : Le canal W_4 construit à partir de deux copies de W_2.

L'opération de permutation R_4 permet de transformer le vecteur $s_1^4 = (s_1, s_2, s_3, s_4)$ en $v_1^4 = (s_1, s_3, s_2, s_4) = (s_{1,o}^4, s_{1,e}^4)$. La Figure 3.2 donne :

$$c_1^4 = (c_1, c_2, c_3, c_4) = \left(u_1 \oplus u_2 \oplus u_3 \oplus u_4, u_3 \oplus u_4, u_2 \oplus u_4, u_4\right) \tag{3.10}$$

La relation entre u_1^4 et c_1^4 est donc : $c_1^4 = u_1^4 \mathrm{G}_4$ avec $\mathrm{G}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$.

La relation liant les probabilités de transitions W_4 et W^4 est $W_4(r_1^4|u_1^4) = W^4(r_1^4|u_1^4 \mathrm{G}_4)$. On peut remarquer que $\mathrm{G}_4 = \mathrm{B}_4 \mathrm{G}_2^{\otimes 2}$, B_4 étant la matrice de permutation des lignes 2 et 3.

La généralisation de la procédure de combinaison de canaux, avec un niveau quelconque $p \geq 1$ $(n = 2^p)$, se fait par une association de deux copies indépendantes du canal $W_{n/2}$ pour réaliser W_n. La chaine de transformation débute par le passage du vecteur d'entrée u_1^n de W_n à s_1^n tel que $s_{2i-1} = u_{2i-1} \oplus u_{2i}$ et $s_{2i} = u_{2i}$ pour $1 \leq i \leq n/2$. L'opérateur R_n sur la Figure 3.3 est une permutation connue sous le nom d'opération d'inversion. Il agit sur son entrée s_1^n pour produire $v_1^n = (s_1, s_3, \dots, s_{n-1}, s_2, s_4, \dots, s_n) = (s_{1,o}^n, s_{1,e}^n)$, qui devient l'entrée pour les deux copies du canal $W_{n/2}$ comme indiqué sur la Figure 3.3.

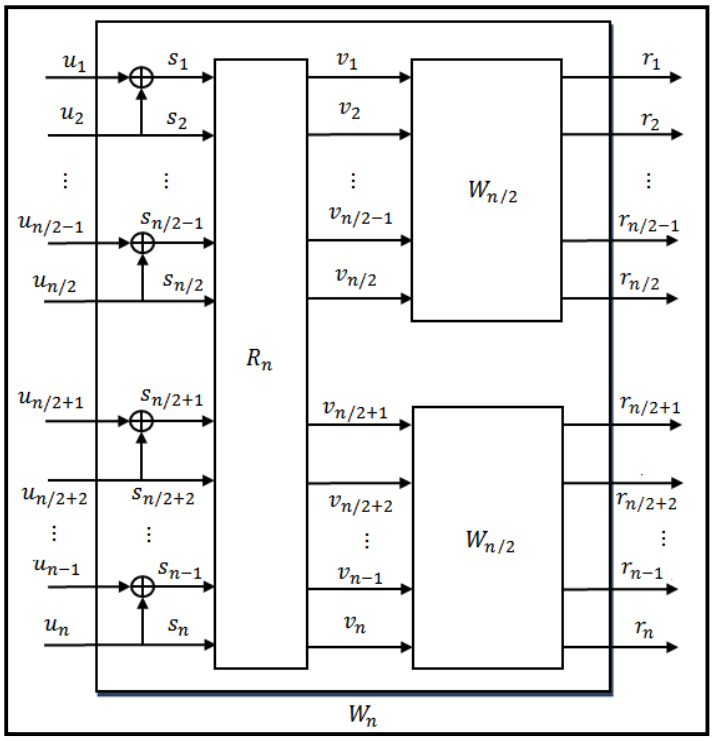

Figure 3.3 : Construction récursive du canal W_n à partir de deux copies de $W_{n/2}$.

L'application qui permet de passer de u_1^n à c_1^n est linéaire sur le corps de Galois $\mathbb{F}_2$, avec

$$c_1^n = u_1^n \mathrm{G}_n \tag{3.11}$$

Les probabilités de transition W_n et W^n sont liées par :

$$W_n(r_1^n|u_1^n) = W^n(r_1^n|u_1^n \mathrm{G}_n). \tag{3.12}$$

La matrice G_n de taille n est appelée matrice génératrice. Arikan a montré que $\mathrm{G}_n = \mathrm{B}_n \mathrm{G}_2^{\otimes p}$ avec B_n une matrice de permutation de lignes appelée bit-reversal.

3.2.2 La division de canal

Après la combinaison de canaux, l'étape suivante de la polarisation de canal consiste à subdiviser le canal W_n de taille n, obtenu avec la combinaison, en n canaux $W_n^{(i)}: \mathcal{X} \longrightarrow \mathcal{Y}^n \times \mathcal{X}^{i-1}$ définis par les probabilités de transition suivantes :

$$W_n^{(i)}\left(r_1^n, u_1^{i-1} \middle| u_i\right) = \sum_{u_{i+1}^n \in \mathcal{X}^{n-i}} \frac{1}{2^{n-1}} W_n(r_1^n|u_1^n) \tag{3.13}$$

Si u_1^n est uniforme sur $\mathcal{X}^n$ alors $W_n^{(i)}$ est le canal effectivement vu par u_i; tout se passe comme si chaque bit d'entrée u_i emprunte le canal $W_n^{(i)}$, avec $1 \leq i \leq n$, pour donner (r_1^n, u_1^{i-1}). Nous pouvons représenter cette équivalence comme suit :

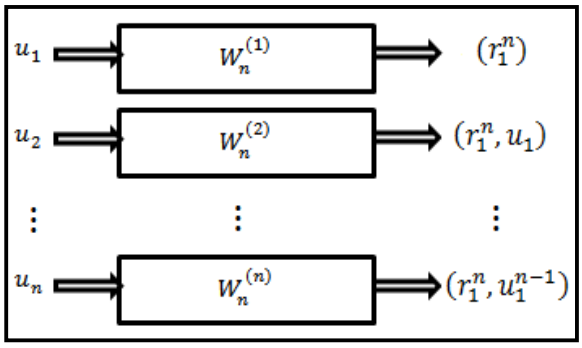

Figure 3.4 : Schéma équivalent de la transmission par codage polaire.

Ce schéma équivalent de codage polaire sera exploité lors du décodage.

3.2.3 Le phénomène de polarisation de canal

Ce phénomène de polarisation s'explique avec le théorème suivant [25] :

Pour tout canal B-DMC donné et $\delta \in [0,1]$ fixé alors les valeurs de $I\left(W_n^{(i)}\right) \in [1-\delta, 1]$ tendent vers $I(W)$ et celles appartenant à $[0, \delta]$ s'approchent de $1 - I(W)$, pour $n = 2^p$ tendant vers l'infini. En d'autres termes les deux pôles de $I\left(W_n^{(i)}\right)$ sont $I(W)$ et $1 - I(W)$.

Voici un schéma montrant l'effet de polarisation de canal [25].

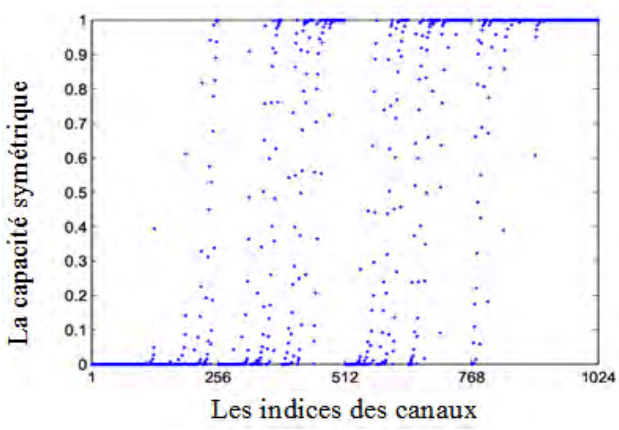

Figure 3.5 : Phénomène de polarisation de canal dans le cas d'un BEC ($\varepsilon = 0.5$).

Sur cette figure est représenté l'ensemble des $I\left(W_n^{(i)}\right)$ en fonction de $i = 2^0, \dots, n = 2^{10}$.

Puisque la construction des canaux se fait de façon récursive voyons quelle est la relation qui lie les canaux des différents niveaux de combinaison.

3.3 La transformation récursive de canal

Le processus de polarisation de canal n'est en réalité qu'une transformation de canal à partir de n exemplaires indépendants d'un B-DMC W donné. Cette transformation commence par un regroupement de deux copies de W par une technique basée sur la règle de chaine de l'information mutuelle. En effet, nous avons d'après cette règle :

d'une part

$$I(u_1, u_2 ; r_1, r_2) = I(u_1 ; r_1) + I(u_2 ; r_2) = 2I(W) \tag{3.14}$$

et d'autre part

$$I(u_1, u_2 ; r_1, r_2) = I(u_1 ; r_1, r_2) + I(u_2 ; r_1, r_2, u_1) \tag{3.15}$$

Les canaux partagés W^- et W^+ représentent ceux dont les informations mutuelles sont respectivement les deux termes de (3.15). Autrement dit $I(W^-) = I(u_1 ; r_1, r_2)$ et $I(W^+) = I(u_2 ; r_1, r_2, u_1)$; ce qui implique avec (3.14) :

$$I(W^-) + I(W^+) = 2\mathrm{I(W)} \tag{3.16}$$

$$W^{-}(r_1, r_2|u_1) = \sum_{u_2 \in \mathcal{X}} \frac{1}{2} W(r_1|u_1 \oplus u_2)\, W(r_2|u_2) \tag{3.17}$$

$$W^{+}(r_1, r_2, u_1|u_2) = \frac{1}{2} W(r_1|u_1 \oplus u_2)\, W(r_2|u_2). \tag{3.18}$$

On pose $W^{-} = W_2^{(1)}$ et $W^{+} = W_2^{(2)}$ et donc $(W, W) \rightarrow \left(W_2^{(1)}, W_2^{(2)}\right)$.

Cette transformation peut être généralisée [25] par :

$$\left(W_n^{(i)}, W_n^{(i)}\right) \xrightarrow{on\ construit} \left(W_{2n}^{(2i-1)}, W_{2n}^{(2i)}\right), \tag{3.19}$$

avec

$$W_{2n}^{(2i-1)}\left(r_1^{2n}, u_1^{2i-2}|u_{2i-1}\right) = \sum_{u_{2i} \in \mathcal{X}} \frac{1}{2} W_n^{(i)}\left(r_1^n, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}\right) W_n^{(i)}\left(r_{n+1}^{2n}, u_{1,e}^{2i-2}|u_{2i}\right) \tag{3.20}$$

$$W_{2n}^{(2i)}\left(r_1^{2n}, u_1^{2i-1}|u_{2i}\right) = \frac{1}{2} W_n^{(i)}\left(r_1^n, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}\right) W_n^{(i)}\left(r_{n+1}^{2n}, u_{1,e}^{2i-2}|u_{2i}\right) \tag{3.21}$$

Les relations entre les capacités des canaux créés à la suite de ces transformations et les canaux de synthèse d'une part et celles liant leurs fiabilités d'autre part sont les suivantes :

$$I\left(W_{2n}^{(2i-1)}\right) + I\left(W_{2n}^{(2i)}\right) = 2I\left(W_n^{(i)}\right) \quad \text{Conservation de la capacité symétrique} \tag{3.22}$$

$$Z\left(W_{2n}^{(2i-1)}\right) + Z\left(W_{2n}^{(2i)}\right) \leq 2Z\left(W_n^{(i)}\right) \qquad \text{Amélioration de la fiabilité} \tag{3.23}$$

La transformation de canal conserve la capacité symétrique et améliore la fiabilité.

$$I\left(W_{2n}^{(2i-1)}\right) \leq I\left(W_n^{(i)}\right) \leq I\left(W_{2n}^{(2i)}\right) \tag{3.24}$$

$$Z\left(W_{2n}^{(2i-1)}\right) \geq Z\left(W_n^{(i)}\right) \geq Z\left(W_{2n}^{(2i)}\right) \tag{3.25}$$

La capacité symétrique du canal de synthèse $W_n^{(i)}$ est bornée par celles des canaux obtenus après transformation. La fiabilité est bornée dans le sens inverse.

$$Z\left(W_{2n}^{(2i-1)}\right) \leq 2Z\left(W_n^{(i)}\right) + Z\left(W_n^{(i)}\right)^2, \tag{3.26}$$

$$Z\left(W_{2n}^{(2i)}\right) = Z\left(W_n^{(i)}\right)^2. \tag{3.27}$$

Toutes les inégalités se transforment en égalité dans le cas où W est un BEC.

On peut considérer la transformation schématique représentée sur la Figure 3.6. Le schéma montre que, pour construire les canaux $W_8^{(i)}$ $1 \leq i \leq 8$ (à gauche), on doit faire appel à 8 copies du canal B-DMC W (à droite). Le canal localisé à un nœud donné est obtenu à partir des deux canaux sur les nœuds situés à sa droite auxquels il est relié. Par exemple, $W_2^{(1)}$ et $W_2^{(2)}$ donnent $W_4^{(1)}$.

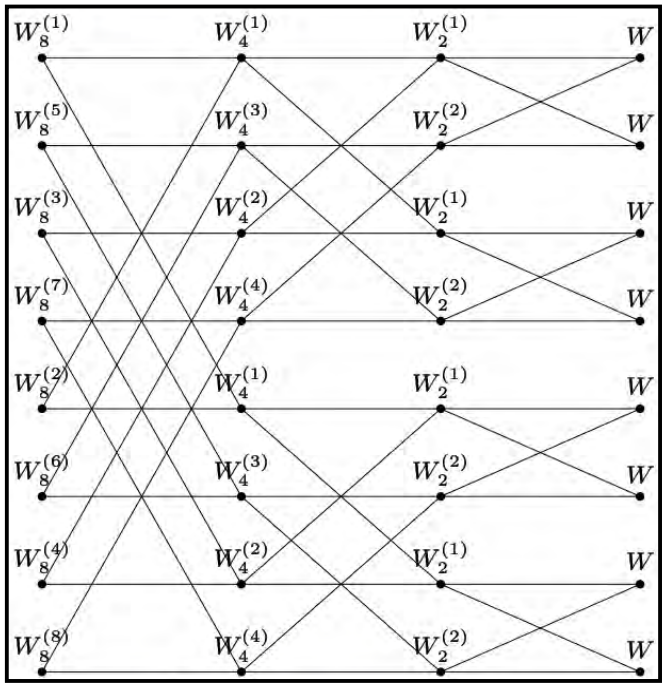

Figure 3.6 : La transformation récursive des canaux pour $n = 8$.

3.4 Le codage polaire

L'idée qui sous-tend le codage polaire est l'exploitation de l'effet de polarisation de canal pour construire des codes qui permettent d'atteindre la capacité symétrique du canal $I(W)$. Le principe du codage polaire est de créer un système de codage permettant d'accéder à chaque canal $W_n^{(i)}$ individuellement et d'envoyer les données à travers ceux qui sont les plus fiables c'est-à-dire ceux dont $Z\left(W_n^{(i)}\right)$ sont plus proches de 0.

Les codes polaires appartiennent à une classe spéciale de codes appelés G_n −coset codes. Ces codes G_n −coset sont définis par l'application qui, à la séquence d'entrée u_1^n, fait correspondre un mot de code c_1^n tel que :

$$c_1^n = u_1^n G_n \tag{3.28}$$

où $n = 2^p, p \geq 1$, représente la longueur de bloc et G_n la matrice de génératrice du code G_n −coset. Etant donné un sous-ensemble A de dimension k inclus dans $\{1, \cdots, n\}$ ($A \subset \{1, \cdots, n\}$) alors la relation (3.28) peut s'écrire sous la forme suivante :

$$c_1^n = u_A G_n(A) \oplus u_{A^c} G_n(A^c) \tag{3.29}$$

Dans cette expression, u_A est un sous vecteur de dimension k de u_1^n, composé des éléments dont les indices sont dans A, $G_n(A)$ est formée par les lignes de G_n dont les indices appartiennent à A et A^c (le complémentaire de A dans l'ensemble $\{1, \cdots, n\}$). Nous allons fixer l'ensemble A (et A^c par conséquent) et u_{A^c} en laissant u_A variable. Le vecteur u_A est appelé vecteur d'information et u_{A^c} vecteur fixé (frozen bits). En général on choisit $u_{A^c} = 0_1^{n-k}$ car le choix de u_{A^c} n'affecte pas les performances d'un canal symétrique [25]. Dans ce cas, la relation 3.29 devient $c_1^n = u_A G_n(A)$; c'est une typologie de code en bloc linéaire de vecteur d'entrée u_A, de matrice génératrice $G_n(A)$. La matrice de contrôle de parité est donnée par les colonnes de G_n du code polaire dont les indices sont dans A^c [29].

Le code polaire est un code G_n −coset de paramètre (n, k, A, u_{A^c}) dans lequel la règle de sélection du vecteur d'information est particulière. En effet, on choisit A de telle sorte que ses indices i, $1 \leq i \leq n$, correspondent aux canaux les plus fiables. En d'autres termes A est choisi tel que $Z\left(W_n^{(i)}\right) \leq Z\left(W_n^{(j)}\right)$, pour tout $i \in A$ et $j \in A^c$. Notons donc, d'après cette règle de sélection de A, qu'un code polaire est spécifique au canal pour lequel il est créé.

3.5 L'encodage polaire

Après la construction du code polaire l'étape suivante consiste à encoder un mot d'information u_1^n en un mot de code c_1^n en utilisant la relation (3.28). Pour cela, le vecteur d'information u_A est d'abord complété avec les bits fixés u_{A^c} pour obtenir le mot source u_1^n. Le mot de code obtenu est transmis à travers le canal de transmission au niveau duquel des erreurs peuvent survenir et transformer le mot de code en un mot reçu r_1^n différent. En analysant

l'opération de codage G_n, on exploite l'idée d'indexation de bits pour interpréter les différentes opérations de permutation qui composent G_n. L'expression de G_n peut s'écrire de manière récursive sous la forme suivante [25]:

$$G_n = R_n\left(G_2 \otimes G_{n/2}\right) = B_n G_2^{\otimes p} = G_2^{\otimes p} B_n \tag{3.30}$$

avec B_n une matrice de permutation définie par :

$$B_n = R_n\left(I_2 \otimes R_{n/2}\right)\left(I_4 \otimes R_{n/4}\right)\cdots\left(I_{n/2} \otimes R_2\right) = R_n\left(I_2 \otimes B_{n/2}\right) \tag{3.31}$$

Partant de ces relations essayons de définir plus explicitement les matrices de permutation B_n et R_n et le calcul de $G_2^{\otimes p}$. Pour cela nous allons utiliser la notation indicielle des vecteurs que nous appliquerons aussi aux éléments des matrices. Soit un vecteur a_1^n alors l'élément a_i sera noté par $a_{b_1 b_2 \cdots b_p}$ où $b_1 b_2 \cdots b_p$ correspond à la représentation binaire de $i-1$.

- La matrice R_n agit sur un vecteur en effectuant un décalage cyclique des bits d'indice vers la gauche. Par exemple si on donne $v_1^n = u_1^n R_n$ alors $v_{b_1 b_2 \cdots b_p} = u_{b_2 \cdots b_p b_1}$.
- La matrice de permutation B_n, constituée de p matrices de décalage, effectue p décalages cycliques vers la gauche en respectant le processus suivant : pour le $i-$ème décalage on considère $p-i+1$ premiers bits en allant de la gauche vers la droite. Ce qui revient à renverser l'ordre des bits d'indice. Pour $w_1^n = u_1^n B_n$ on se retrouve avec $w_{b_1 b_2 \cdots b_p} = u_{b_p \cdots b_2 b_1}$.

Exemple 3.1:

Prenons l'exemple de $n = 8$, $v_1^8 = u_1^8 R_8$ et $w_1^8 = u_1^8 B_8$ avec

$$u_1^8 = (u_{000}, u_{001}, u_{010}, u_{011}, u_{100}, u_{101}, u_{110}, u_{111}) = (u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8). \tag{3.32}$$

On aura ainsi

$$v_1^8 = (u_{000}, u_{010}, u_{100}, u_{110}, u_{001}, u_{011}, u_{101}, u_{111}) = (u_1, u_3, u_5, u_7, u_2, u_4, u_6, u_8) \tag{3.33}$$

$$w_1^8 = (u_{000}, u_{100}, u_{010}, u_{110}, u_{001}, u_{101}, u_{011}, u_{111}) = (u_1, u_5, u_3, u_7, u_2, u_6, u_4, u_8). \tag{3.34}$$

Le schéma de codage polaire peut se résumer avec la figure suivante :

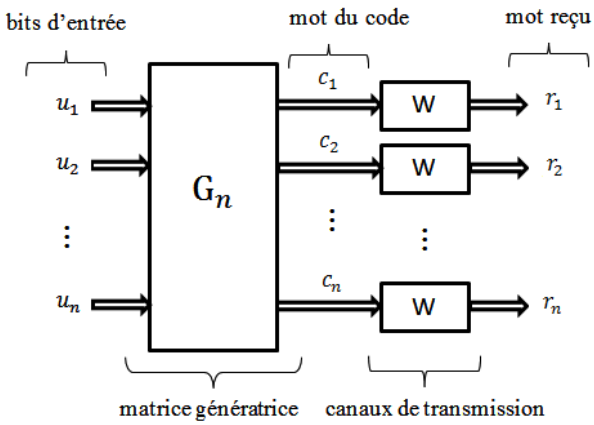

Figure 3.7 : Schéma global du codage polaire.

Le processus d'encodage peur être représenté sous forme d'un facteur de graphe équivalent à G_n ou simplement avec un graphe sans permutation B_n, donc équivalent à $G_2^{\otimes p}$. Un exemple est donné avec la figure ci-dessous. Dans cette exemple $n = 8$ et $k = 4$. D'après les valeurs des capacités de canal $I\left(W_n^{(i)}\right)$, les indices des bits d'information (free) sont $A = \{4,6,7,8\}$ et les bits fixés (frozen) $A^c = \{1,2,3,5\}$. Nous considérons que ces frozen bits sont tous égaux à 0. Le graphe représente le mappage $c_1^8 = u_1^8 G_8 = u_1^8 G_2^{\otimes 3}$.

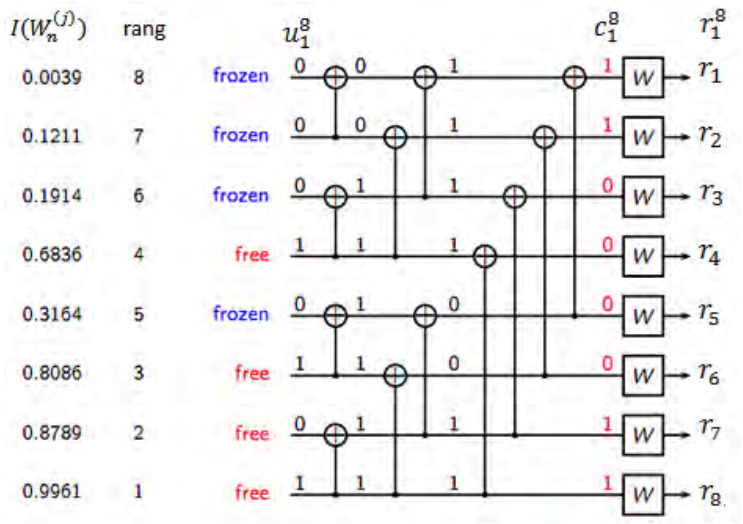

Figure 3.8 : Encodage polaire avec un facteur de graphe pour $n = 8$ et $k = 4$.

3.6 Le décodage des codes polaires

Plusieurs algorithmes de décodage des codes polaires existent à savoir l'annulation successive SC [25], le décodage par programmation linéaire LP [29], ALP et le SC avec liste

SCL (SC List) [27]. On assiste aujourd'hui de plus en plus à des propositions soit de nouvelles méthodes de décodage ou des améliorations des méthodes existantes. Les algorithmes de décodage proposés sont soit basés LR ou LLR.

3.6.1 Décodage SC

3.6.1.1 Principe du décodage SC

SC est le premier et inhérent décodage proposé par Arikan pour les codes polaires. Cette procédure de décodage est dictée par la structure récursive du code. Pour mieux comprendre cette méthode de décodage, considérons à nouveau le schéma équivalent du codage polaire (Figure 3.4) dans lequel le bit d'entrée u_i est transformé dans le canal $W_n^{(i)}$ en générant en sortie (u_1^{i-1}, r_1^n). Puisque le décodage est l'opération inverse du codage, il s'avère donc logique de considérer l'application qui, à $(\hat{u}_1^{i-1}, r_1^n)$, associe $\hat{u}_i$, avec $\hat{u}_i$ l'estimation de la valeur de u_i par la méthode de décodage. L'interprétation basique de cette application est : le $ième$ bit d'entrée u_i est décodé à partir des $i-1$ bits précédemment décodés $\hat{u}_1^{i-1}$ et du mot reçu y_1^n. Notons que pour les positions fixées, aucun calcul n'est nécessaire et $\hat{u}_{A^c} = u_{A^c}$. Cette technique de décodage, proposée par Arikan [25], [62], est connue sous le nom d'annulation successive. D'abord, une estimation $\hat{u}_1$, de u_1 est donnée à partir de r_1^n. Ensuite, $\hat{u}_2$ est décodé à partir de r_1^n et $\hat{u}_1$, etc. La tâche d'un tel décodeur SC consiste donc à déterminer une estimation $\hat{u}_1^n$ de u_1^n à partir de la connaissance de A, u_{A^c} et r_1^n. Puisque le décodeur connait les frozen bits $\hat{u}_{A^c} = u_{A^c}$, sa tâche réelle sera de déterminer une estimation $\hat{u}_A$ des bits d'information u_A. Le décodeur génère son estimation par l'algorithme de décision suivant :

$$\hat{u}_i \triangleq \begin{cases} u_i, & si\ i \in A^c \\ 0 & si\ L_n^{(i)}\left(r_1^n, \hat{u}_1^{i-1}\right) \geq 1 \\ 1, & sinon \end{cases} \qquad ,1 \ \leq i \leq n \tag{3.35}$$

avec le rapport de vraisemblance (LR) $L_n^{(i)}$ défini par :

$$L_n^{(i)}\left(r_1^n, \hat{u}_1^{i-1}\right) = \frac{W_n^{(i)}\left(r_1^n, \hat{u}_1^{i-1} \middle| 0\right)}{W_n^{(i)}\left(r_1^n, \hat{u}_1^{i-1} \middle| 1\right)}. \tag{3.36}$$

En utilisant les relations (3.20) et (3.21) on peut montrer que :

$$L_n^{(2i-1)}\left(r_1^n, \hat{u}_1^{2i-2}\right) = \frac{1 + L_{n/2}^{(i)}\left(r_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) \cdot L_{n/2}^{(i)}\left(r_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}\right)}{L_{n/2}^{(i)}\left(r_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) + L_{n/2}^{(i)}\left(r_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}\right)} \tag{3.37}$$

et

$$L_n^{(2i)}\left(r_1^n, \hat{u}_1^{2i-1}\right) = \left[L_{n/2}^{(i)}\left(r_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)\right]^{1-2\hat{u}_{2i-1}} \cdot L_{n/2}^{(i)}\left(r_{n/2+1}^{n}, \hat{u}_{1,e}^{2i-2}\right). \quad (3.38)$$

Ainsi, le calcul d'un LR de longueur n est déduit de celui de deux LRs de longueur $n/2$. On peut écrire $\left(L_{n/2}^{(i)}, L_{n/2}^{(i)}\right) \rightarrow \left(L_n^{(2i-1)}, L_n^{(2i)}\right)$. Cette récursivité continue jusqu'aux LRs de longueur 1 qui sont directement calculés par $L_1^{(1)}(r_i) = \frac{w(r_i|0)}{w(r_i|1)}$.

3.6.1.2 FFT structure (Butterfly-based architecture)

Arikan a montré que le décodage SC peut être efficacement représenté par le facteur de graphe du code qui a une structure ressemblant à la Transformée de Fourier Rapide FFT (Fast Fourier Transform). Nous allons donner deux structures : la première est la structure originale proposée par Arikan et la seconde est une autre représentation de la première.

- **Structure FFT originale**

Puisque chaque LR $L_n^{(i)}$ est associé au canal $W_n^{(i)}$, le graphe de calcul des $L_n^{(i)}$ est le même que celui de la Figure 3.6 de construction des canaux $W_n^{(i)}$. Un exemple de cette représentation est donné par la Figure 3.9 avec $n = 8$. Pour calculer le LR $L_n^{(i)}$ situé sur le nœud correspondant au canal $W_n^{(i)}$ nous devons exploiter le calcul des valeurs de rapports de vraisemblance situés sur les nœuds les canaux ayant participé à la construction du canal $W_n^{(i)}$. Chaque nœud est affecté de deux étiquettes (l'expression de l'entrée du canal et le numéro d'activation du nœud). Par exemple, le nœud ayant les étiquettes $\left(r_1^4, \hat{u}_{1,e}^6 \oplus \hat{u}_{1,o}^6\right)$ et 30 indique que la valeur du LR à calculer à ce nœud est $L_4^{(4)}\left(r_1^4, \hat{u}_{1,e}^6 \oplus \hat{u}_{1,o}^6\right)$ et que ce nœud sera le 30$^{\text{ème}}$ à être activé. Les estimations sont données dans l'ordre croissant des indices de $\hat{u}_i$ (voir (3.32)).

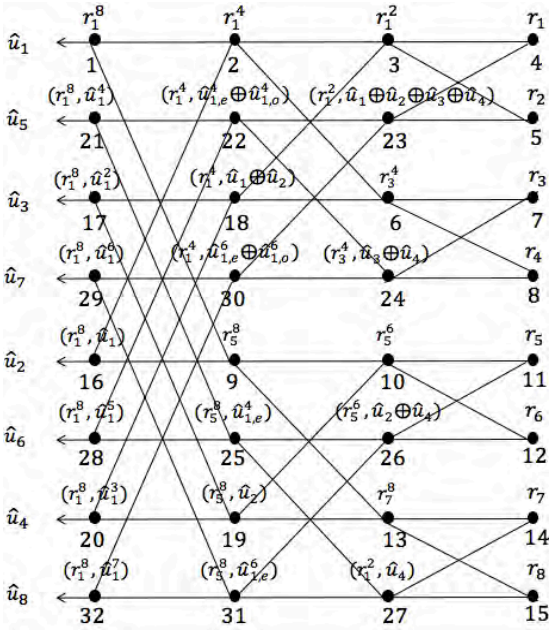

Figure 3.9 : Le décodage SC pour le codage polaire de longueur de bloc $n = 8$.

Pour générer $\hat{u}_1$ en calculant $L_8^{(1)}(r_1^8)$, le nœud 1 (decision element, DE) a besoin des LRs des nœuds 2 et 9. Il s'adresse d'abord au nœud 2 qui s'adresse à son tour aux nœuds 3 et 6 mais d'abord au nœud 3. Une fois sollicité, le nœud 3 s'adresse au nœud 4 puis au nœud 5 pour délivrer son calcul au nœud l'ayant sollicité (nœud 2). C'est après avoir reçu le LR du nœud 3 que le nœud 2 sollicite le nœud 6. Et ainsi de suite. Chaque nœud dans le schéma, après avoir calculé son LR passe le résultat aux deux nœuds de gauche auxquels il est lié (voisins de gauches) lorsqu'ils le sollicitent. Le nœud 1 calculera son LR $L_8^{(1)}(r_1^8)$ après avoir reçu les LRs des nœuds 2 et 9 et décode $\hat{u}_1$ en utilisant (3.35). Le processus continue jusqu'au nœud 32 avec le calcul de $L_8^{(1)}(r_1^8, \hat{u}_1^7)$ pour donner la dernière estimation $\hat{u}_8$. Pour plus de détaille sur le décodage SC avec cette structure voir [25].

O Seconde Structure FFT

La représentation est la même que celle précédente mais cette structure, plus adaptée pour une implémentation, est décomposée en étages et est interprétée de manière un peu différente et plus simple. On peut avoir deux configurations pour le graphe de décodage. La première est similaire à celle de la Figure 3.9 et correspond au cas où on applique la matrice de permutation B_n. Dans ce cas, les bits estimés sont obtenus dans croissant des indices (voir (3.32)). La seconde configuration représente le cas où la matrice de permutation B_n n'est pas appliquée et, dans ce cas, les bits estimés seront dans l'ordre de permutation des indices (voir

(3.34)). Notons que le processus de décodage est le même pour les deux configurations. La Figure 3.10 montre le processus du décodage SC avec $n = 8$ pour la représentation sans B_8. Ce processus commence à droite du graphe avec les LRs $L(r_i)$ des bits du mot reçu qui sont combinés par pair en progressant vers la gauche du graphe. Celui-ci est composé de $\log n = 3$ étages chacune contenant $n = 8$ nœuds. Les bits décodés sont situés à gauche du graphe. Pour chaque nœud, les deux LRs entrant, notés L_a et L_b sont combinés pour produire le LR correspondant à ce nœud. Deux fonctions de calcul de LR (f et g) sont utilisées. Ces fonctions correspondent aux relations (3.37) et (3.38). Les nœuds exécutant la fonction f sont labélisés f (blanc) et ceux exécutant la fonction g sont labélisés g (gris). De façon générale, les LRs de l'étage j sont calculés à partir des LRs de l'étage $j-1$. Ici, (3.35) est exprimée par h pour déterminer le mot décodé. A partir des relations (3.37) et (3.38) on a:

$$\begin{cases} f(L_a, L_b) = \dfrac{1 + L_a \cdot L_b}{L_a + L_b} \\ g(L_a, L_b, \hat{u}_{sum}) = g_{\hat{u}_{sum}}(L_a, L_b) = {L_a}^{(1-2\hat{u}_{sum})} \cdot L_b = \begin{cases} L_a \cdot L_b \; si \; \hat{u}_{sum} = 0 \\ ou \\ L_b / L_a \; si \; \hat{u}_{sum} = 1 \end{cases} \end{cases} \quad (3.39)$$

où $\hat{u}_{sum}$ représente la somme binaire partielle des bits précédemment estimés (voir Figure 3.9). Ce sont les sommes binaires présentes aux labels des nœuds de la Figure 3.9. La valeur de $\hat{u}_{sum}$ détermine si la fonction g est une multiplication ou une division.

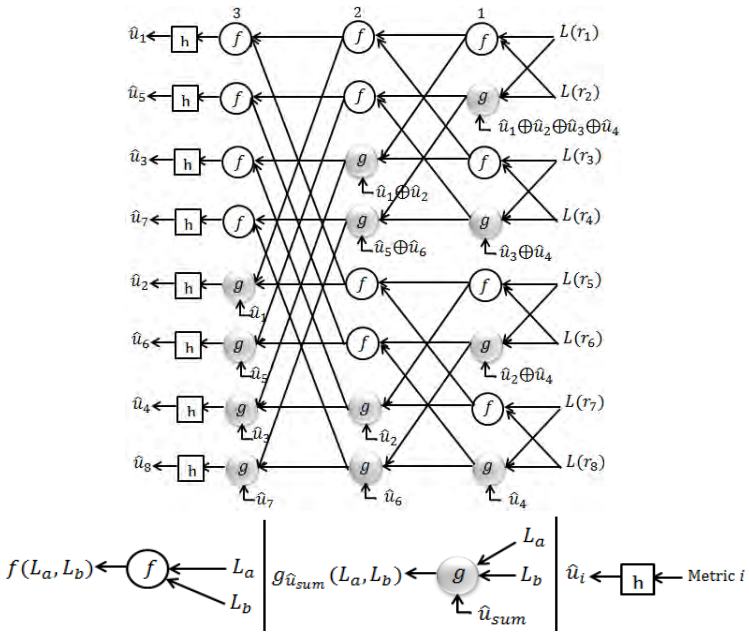

Figure 3.10 : Graphe de décodage SC de Structure FFT $n = 8$.

3.6.1.3 Version du décodeur SC basée LLR

La version du décodeur SC original, présentée dans la section précédente, est basée dans le domaine LR (likelihood ratio) dans lequel les fonctions f et g nécessitent des multiplications et des divisions qui sont difficiles à implémenter en pratique [63]. Leroux et al. ont proposé de réaliser le décodeur SC dans le domaine LLR (log-likelihood ratio) en vue de réduire la complexité des calculs de f et g. Dans le domaine LLR, (3.35) devient :

$$\hat{u}_i \triangleq \begin{cases} u_i, & si\ i \in A^c \\ 0 & si\ LL_n^{(i)}\left(r_1^n, \hat{u}_1^{i-1}\right) \geq 0 \\ 1, & sinon \end{cases} \quad ,1 \ \leq i \leq n \tag{3.40}$$

et les fonctions f et g deviennent :

$$\begin{cases} f(LL_a, LL_b) = 2\tanh^{-1}(\tanh(LL_a/2) \cdot \tanh(LL_b/2)) \\ g(LL_a, LL_b, \hat{u}_{sum}) = g_{\hat{u}_{sum}}(LL_a, LL_b) = LL_a \cdot (-1)^{\hat{u}_{sum}} + LL_b = \begin{cases} LL_a + LL_b\ si\ \hat{u}_{sum} = 0 \\ ou \\ -LL_a + LL_b\ si\ \hat{u}_{sum} = 1 \end{cases} \end{cases} \tag{3.41}$$

où les valeurs de LLR sont définies $LL_a \triangleq \ln(L_a)$ et $LL_b \triangleq \ln(L_b)$. Pour une implémentation matérielle, la fonction g peut facilement être implémentée par un additionneur/soustracteur conditionné par le bit $\hat{u}_{sum}$ alors que la fonction f est toujours complexe. Ainsi, comme pour les codes LDPC, l'approximation min-sum [64] peut aussi être exploitée pour réduire la complexité de f. Nous auront alors :

$$f(LL_a, LL_b) \approx \tilde{f}(LL_a, LL_b) \triangleq sign(LL_a) \cdot sign(LL_b) \cdot \min(|LL_a|, |LL_b|) \tag{3.42}$$

Si des LLR intermédiaires sont stockés, alors la complexité de calcul du décodeur SC est $O(n\ log\ n)$ [25].

3.6.1.4 Représentation en arbre du décodeur SC

Le processus de décodage SC peut être vu (représenté) comme un algorithme de recherche d'arrêtes dans un arbre de codes. Le décodage commence avec la racine qui a deux arrêtes (branches) avec des labelles 0 et 1 et les métriques $W_n^{(i)}(\mathrm{r}|0)$ et $W_n^{(i)}(\mathrm{r}|1)$, respectivement. Le décodage SC choisit l'arrête avec la plus grande métrique et l'autre est abandonné. L'arrête choisi donne, à son tour, deux arrêtes avec des labelles 0 et 1 avec les métriques $W_n^{(i)}(\mathrm{r}, \hat{u}_1|0)$ et $W_n^{(i)}(\mathrm{r}, \hat{u}_1|1)$, respectivement. Généralement, à chaque niveau le bit $\hat{u}_i$ est décodé (choisi

entre 0 et 1) en comparant les deux métriques $W_n^{(i)}\left(\mathrm{r}, \hat{u}_1^{i-1}|0\right)$ et $W_n^{(i)}\left(\mathrm{r}, \hat{u}_1^{i-1}|1\right)$ si i n'est pas une position frozen. Si i est une position frozen le décodeur SC donne $\hat{u}_i = u_i$. Cette procédure continue jusqu'aux nœuds feuilles où la dernière estimation est faite avec $\hat{u}_n$.

La Figure 3.11 montre un exemple de la procédure de décodage SC pour $n = 4$ et $k = 4$. Cet arbre est constitué de 4 niveaux, où chaque niveau représente un bit décodé. La valeur associée à chaque nœud est la métrique basée LR pour le chemin de décodage du nœud racine au nœud courant. Les arêtes en gras rouges dans la figure montrent le chemin de décodage SC. Le nombre écrit à côté de chacun des nœuds correspond à la métrique du chemin de décodage de la racine à ce nœud. Les nœuds qui sont étendus au cours de la procédure de décodage SC sont représentés par les cercles numérotés et les numéros correspondants indiquent l'ordre de traitement. Les cercles noirs représentent les nœuds qui sont visités (dont leur métrique de chemin est calculée) mais qui n'ont pas été retenus et les cercles gris sont ceux qui ne sont pas visités pendant le processus de recherche. Le chemin retenu par le processus de décodage n'est pas garanti d'être le plus probable. Comme le montre l'exemple, celui marqué (**1000**) a la plus grande probabilité de tous les chemins de longueur n, mais il a échoué dans la compétition au premier niveau. Le décodage est fait progressivement avec les métriques suivantes : $W_n^{(i)}(\mathrm{r}|0) = 0{,}55$ puis $W_n^{(i)}(\mathrm{r}, \hat{u}_1 = 0|0) = 0{,}3$ ensuite $W_n^{(i)}(\mathrm{r}, \hat{u}_1^2 = 00|1) = 0{,}25$ et enfin $W_n^{(i)}(\mathrm{r}, \hat{u}_1^3 = 001|1) = 0{,}2$. Ainsi, la sortie du décodeur sera $\hat{u}_1^4 = (\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4) = (\mathbf{0011})$. La sortie du décodeur correspondant à celui du décodeur ML est le chemin de longueur 4 qui a la plus grande métrique au plus bas niveau. Dans cet exemple, c'est (**1000**) avec la métrique de chemin $W_n^{(i)}(\mathrm{r}, 100|0) = 0{,}36$ qui est le chemin optimal.

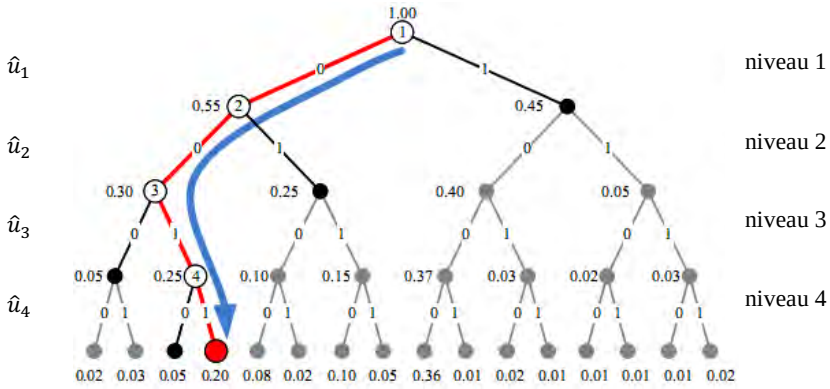

Figure 3.11 : Exemple de décodage SC sur un arbre pour $n = 4$ et $k = 4$.

On dira que le décodage a échoué si $\hat{u}_1^n \neq u_1^n$ (équivalent à $\hat{u}_A \neq u_A$) et a réussi dans le cas contraire. La complexité temporelle de l'algorithme de décodage est déterminée essentiellement par celle du calcul des LRs. En utilisant une structure d'espace efficace [65] pour implémenter le décodeur SC, les complexités temporelle et d'espace sont $O(n\,logn)$ et $O(n)$, respectivement. Bien que les codes polaires atteignent asymptotiquement la capacité du canal, des études empiriques ont montré que pour des longueurs de bloc finies (faibles et moyennes), utilisées dans la pratique, le décodage SC des codes polaires a une performance inférieure à celle des Turbo codes et des codes LDPC, en termes de probabilité d'erreurs de décodage. Pour améliorer le décodeur SC, Tal et Vardy ont proposé une variante appelée décodeur SC liste (SCL).

3.6.2 Décodage SCL

3.6.2.1 Introduction et principe

La performance de SC est limitée par la stratégie du décodage bit-par-bit. Puisque si un bit est mal décodé, il n'y a aucune chance de le corriger dans la suite du processus de décodage. Pour améliorer le décodeur SC Tal et Vardy [27] ont proposé le SCL. Le décodeur SCL est régi par un seul paramètre entier puissance de 2, L, qui désigne la taille de la liste. D'une manière générale, des valeurs plus grandes de L signifient des taux d'erreur plus faibles mais des temps d'exécution plus longs. Tout comme le SC, le SCL peut être représenté sous forme d'un arbre de codes.

3.6.2.2 Le décodeur SCL vu sous forme d'un arbre de codes

Comme SC, le SCL décode les bits d'entrée $\hat{u}_i,\ i = 1, \dots, n,$ successivement un-par-un (niveau-par-niveau). Contrairement au SC où un seul chemin est conservé après traitement à chaque niveau, SCL permet d'exploiter simultanément jusqu'à L chemins candidats pour le niveau suivant. Pour chaque niveau, le décodeur SCL double le nombre de chemins candidats pour chaque bit $\hat{u}_i$ ($\hat{u}_i = 0$ et $\hat{u}_i = 1$) comme la montre la Figure 3.12. Si $2L$ chemins candidats sont obtenus alors une procédure d'élagage est utilisée pour sélectionner les L chemins les plus probables (avec les plus grandes métriques). Ces L chemins sont stockés dans une liste pour le traitement au niveau suivant. Notons que pour un frozen bit, le nombre de chemins candidats n'est pas doublé car un tel bit est fixé et sa valeur est connue. A la fin du processus de décodage (lorsqu'on atteint les nœuds feuilles), le plus probable (qui a la plus grande métrique dans la liste) parmi les L chemins de décodage est sélectionné comme la sortie du décodeur.

La Figure 3.12 donne un exemple simple de recherche sur un arbre de code utilisant un décodeur SCL, avec $L = 2$, $n = 4$ et $k = 4$ [66]. Au niveau 1, le décodeur SCL visite les deux nœuds avec $\hat{u}_1 = 0$ et $\hat{u}_1 = 1$. Puis, au niveau 2, les 2 nœuds descendants (fils), pour chaque nœud du niveau précédent, sont explorés (4 nœuds au total). Puisque la taille de la liste du décodeur est $L = 2$, après avoir calculé toutes les $2L = 4$ nouvelles métriques de chemins associées à ces nœuds fils, le décodeur SCL sélectionne les $L = 2$ chemins qui ont les plus grandes métriques (les plus probables) comme les chemins à conserver. Ensuite, au niveau suivant 3, $2L = 4$ nœuds fils (les nœuds numérotés et en noirs dans la Figure 3.12) qui sont reliés aux L chemins retenus au niveau précédent (niveau 2) sont visités par le décodeur SCL. L'élagage est également appliqué à ce niveau pour choisir les $L = 2$ chemins les plus probables. Enfin, au niveau, $2L = 4$ nœuds fils sont encore visités dont les $L = 2$ plus probables seront retenus. Au final, le décodeur donne en sortie le chemin avec la plus grande métrique parmi les deux candidats dans la liste ; il s'agit de (0011) avec une métrique se 0,2 et (1000) avec une métrique de 0,36. Le chemin de décodage valide (**1000**), qui ne pouvait pas être trouvé par le décodeur SC avant, peut être obtenu maintenant par le décodeur SCL.

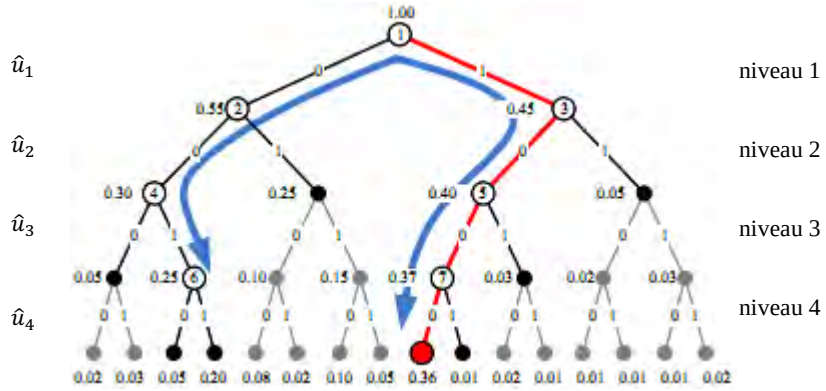

Figure 3.12 : Processus de recherche du décodeur SCL avec $n = 4$, $k = 4$ et $L = 2$.

Basé sur SCL, une autre variante de SC appelée SC pile (SCS, SC stack) est également proposé par Chen et al. [66]. Les auteurs ont proposé une pile à la place d'une liste. Ils ont montré que SCS donne des résultats similaires à la place du SCL.

3.6.2.3 Performances du décodeur SCL

Il est évident que, une taille de la liste L plus grande entraine de meilleures performances du décodeur SCL. Les résultats de simulations dans [65] ont montré que pour une longueur de bloc

$n = 2048$ bits et un taux $R = 1/2$, une taille relativement petite de la liste de $L = 32$ est suffisante pour avoir des performances très proches du décodeur ML.

3.6.2.4 Complexité

Puisque L chemins de décodage sont maintenus simultanément et chaque chemin à une complexité de $O(n)$, la complexité du décodeur SCL est alors $O(Ln)$. En plus, puisque l'arbre de code a n niveaux, une implémentation directe du décodeur SCL effectuerait $O(Ln^2)$ calculs. Cependant, un choix intelligent des structures de données, et la nature récursive des calculs permettent aux auteurs de [65] d'utiliser la technique appelée "lazy copy" (mécanisme de copie sur écriture) basée sur la structure de partage de la mémoire entre les chemins candidats pour réduire cette complexité. Par conséquent, ils ont montré que le décodeur SCL peut être implémenté avec une complexité de calcul de $O(Ln \log n)$.

Comme pour le décodeur SC, le décodeur SCL peut être aussi réalisé en utilisant les métriques basées LLR (Log-Likelihood-Ratio).

3.6.2.5 Décodage SCL base LLR

L'algorithme SCL original [27] est décrit dans le domaine LR alors que la plupart des modules de traitement et de stockage dans les systèmes de transmission numériques modernes sont basés LLR [67]. Balatsoukas et al. [68] ont montré que l'algorithme de décodage SCL pouvait être formulé exclusivement dans le domaine LLR. Ils ont utilisé quelques propriétés utiles de la formulation basée LLR afin de réduire la complexité de l'étape de tri du décodeur SCL. En outre, le décodeur SCL basée LLR peut facilement être incorporé dans les systèmes de communication existants tandis que le décodeur basé LR nécessiterait des étapes de traitement supplémentaires pour convertir les LLRs du canal en LR. En outre, le décodage SCL basé LLR permet une implémentation efficace en espace et numériquement stable, une réduction significative de la taille de l'architecture matérielle précédente ainsi qu'une augmentation de la fréquence de fonctionnement maximale [68]. Yuan et Parhi [67] ont proposé un algorithme SCL basé LLR en redéfinissant la mise à jour des métriques des chemins sous forme LLR. L'approche proposée, peut atteindre à la fois la réduction de la latence et de la complexité matérielle en même temps [67].

3.6.3 Décodeur SCL concaténé avec les CRC (Cyclic Redundancy Check)

Tal et Vardy [27] ont observé dans leurs simulations que, dans la plupart des cas, lorsque le décodeur SCL échoue, le mot de code transmis (chemin correct) se trouve parmi les L chemins de la liste, mais il n'est pas le plus probable et n'est donc pas choisi comme sortie du décodeur. Une erreur de décodage se produit car il y a un autre chemin plus probable qui est sélectionné comme sortie du décodeur. Notons que dans une telle situation, le décodeur ML allait également échouer. Tal et Vardy ont conclu que les performances des codes polaires seraient encore considérablement améliorées avec un outil appelé "genie aided" (assistance) capable d'identifier le mot de code transmis s'il se trouve dans la liste. Ce qui peut facilement être implémenté en utilisant le pré-codage de contrôle de redondance cyclique CRC [65]. Elle consiste à ajouter des bits de plus non-frozen au code polaire. Le décodeur SCL, élimine d'abord les chemins parmi les L candidats qui ne passent pas le CRC et ensuite choisit le chemin le plus probable parmi ceux qui restent. Cependant, il existe un compromis entre la longueur du CRC et le gain de performance. En effet, plus le CRC est long, plus il peut rejeter de mots de code incorrects et plus il dégrade les performances du code polaire initial du fait de l'augmentation de son taux. Le choix de la longueur du CRC est guidé par la taille L de la liste du décodeur SCL et la valeur de SNR considérée. Nous donnons trois exemples de CRCs différents de longueurs 4, 8 et 16 dont leurs polynômes générateurs sont, respectivement :

$$\begin{cases} g(x) = x^4 + x + 1 & pour\ \ CRC\ \ 4\ bits \\ g(x) = x^8 + x^7 + x^6 + x^4 + x^2 + 1 & pour\ \ CRC\ \ 8\ bits \\ g(x) = x^{16} + x^{15} + x^2 + 1 & pour\ \ CRC\ \ 16\ bits \end{cases} \quad (3.43)$$

Les résultats empiriques de [65] ont montré qu'un code polaire de longueur $n = 2048$ et le taux $R = 1/2$, décodé par un SCL avec $L = 32$ et assisté par un CRC de 16 bits, réalisent une meilleure performance en terme de taux d'erreur binaire BER (Bit Error Rate) comparés aux turbo-codes et codes LDPC de l'état de l'art avec $n = 2304$ utilisé dans la norme WiMAX avec un même taux, lorsqu'on considère une probabilité d'erreur de 10^{-4}.

3.6.4 Décodage par programmation linéaire LP

3.6.4.1 Notations et définitions

- Les ensemble d'indices $J = \{1, \ldots, m\}$ et $I = \{1, \ldots, n\}$ désignent respectivement les lignes et colonnes de H. La matrice H est souvent représentée par un *graphe de Tanner* $Gph = (V, E)$. L'ensemble des nœuds V de Gph est composé de deux ensembles disjoints d'indices

dans I (appelés *nœuds de contrôle*) et J (appelés *nœuds de variables*), respectivement. Une arête $[i, j] \in E$ relie les nœuds i et j si et seulement si $H_{ji} = 1$.

- L'*enveloppe convexe* d'un ensemble S de points dans l'espace euclidien est le plus petit ensemble convexe contenant S. L'enveloppe d'un ensemble fini de points $S \in \mathbb{R}^n$ est l'ensemble de toutes les combinaisons convexes de ses points :

$$Conv(S) = \left\{ \sum_{i=1}^{|S|} \lambda_i s_i : \lambda_i \geq 0, \sum_{i=1}^{|S|} \lambda_i = 1 \right\} \tag{3.44}$$

Si $n = 2$, il forme un polygone convexe et, plus généralement, un *polytope* dans $\mathbb{R}^n$. Chaque point s_i de S qui n'appartient pas à l'enveloppe convexe des autres points ($s_i \notin Conv(S \backslash \{s_i\})$ est appelé *sommet* de $Conv(S)$.

- La *Programmation Linéaire (Linear Programming, LP)* consiste à trouver une solution à un ensemble d'inégalités qui optimise une fonction objectif linéaire.
- Une solution *intégrale* dans un polytope à un LP est un point dans le polytope dont les composants sont tous des entiers.
- Un problème *ILP* (*Integer LP)* est un LP où les variables sont contraintes à être des entiers. Les problèmes *ILP* sont NP-difficile dans plusieurs situations pratiques.
- Un problème *BILP (Binary ILP)* est un cas spécial d'ILP où les variables sont contraintes d'être binaires (0 ou 1). Un tel problème est aussi classé comme NP-difficile.
- On appelle *LP Relaxation* l'utilisation d'un LP pour approximer la solution d'un problème ILP [69]. Le LP relaxation d'un BILP est le problème qu'on obtient en remplaçant chaque contrainte de la forme $x_i \in \{0, 1\}$ du problème BILP initial par une paire de contraintes linéaires $0 \leq x_i \leq 1$. Cette technique de relaxation transforme un problème d'optimisation NP-difficile en un problème connexe qui peut être résolu en un temps polynomial.
- Un *cutting-plane* est une méthode d'optimisation qui affine itérativement un ensemble réalisable au moyen d'inégalités linéaires, appelés *cuts*.

3.6.4.2 Le décodage ML (ou MAP)

Théoriquement l'algorithme de décodage optimal des codes polaires est le décodeur ML (Maximum-Likelihood) ou MAP (Maximum A posterori). Cependant, étant trop complexe pour être implémenté en pratique, cet algorithme est considéré comme référence pour la comparaison des performances des autres algorithmes de décodage. La probabilité qu'un mot de code $\mathrm{c} \in C$ soit transmis à travers le canal binaire symétrique sans mémoire et que r soit

reçu est Pr[r|c]. En supposant que les mots de code sont équiprobables, le décodage ML est équivalent à résoudre le problème d'optimisation ci-dessous [69] :

$$\arg\max_{c\in C}\ (\Pr[r|c]) = \arg\max_{c\in C}\left(\prod_{i=1}^{n}\Pr[r_i|c_i]\right) \tag{3.45}$$

où $c_i \in \{0,1\}$ et $r_i \in \{0,1\}$ désignent, respectivement, le $i^{ème}$ symbole des mots c et r. Ce problème d'optimisation peut être réécrit comme suit (voir annexe pour la démonstration) :

$$\arg\min_{c\in C}\left(\sum_{i=1}^{n}\gamma_i c_i\right) = \arg\min_{c\in C}(\langle c,\gamma\rangle) = \arg\min_{c\in C}\ c\,\gamma^T \tag{3.46}$$

avec $\gamma = (\gamma_1, \dots, \gamma_n)$ le vecteur des rapports de vraisemblance LLR (Log-Likelihood Ratios) $\gamma_i = \log\frac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i=1]}$. La solution donnée par ce problème d'optimisation correspond au mot de code ML. Le signe de γ_i determine si le bit transmis c_i est plus probable d'être un 0 ou 1. Si c_i est plus susceptible d'être un 1 alors γ_i sera négatif et si c_i est plus probable d'être un 0 alors γ_i sera positif. On appelle $\sum_{i=1}^{n}\gamma_i c_i$ le coût d'un mot de code particulier c, où γ_i représente le coût en mettant un bit particulier c_i à 1. Nous allons souvent exploiter le fait que le vecteur coût γ peut être uniformément redimensionné sans modifier la solution du problème ML. Par exemple, étant donné un canal binaire symétrique (BSC) avec une probabilité d'erreurs $p_e = \Pr[0|1] = \Pr[1|0]$, on a $\gamma_i = \log[(1-p_e)/p_e]$ si le bit reçu $r_i = 0$, et $\gamma_i = \log[p_e/(1-p_e)] = -\log[(1-p_e)/p_e]$ si $r_i = 1$. On peut considérer $\gamma_i = +1$ si $r_i = 0$, et $\gamma_i = -1$ si $r_i = 1$ dans la résolution du problème.

3.6.4.3 Relaxation du décodage ML (ou décodage LP) des codes polaires

Le décodage ML peut être formulé sous forme d'un LP équivalent. Notons qu'il existe une correspondance un à un entre les mots de codes et les solutions intégrales du décodage LP [69]. Pour un code linéaire $\mathcal{C}(n,k)$ donné, le *polytope des mots de code* est défini par l'enveloppe convexe de tous les mots de code possibles [69] :

$$poly(\mathcal{C}) = Conv(\mathcal{C}) = \left\{\sum_{c\in\mathcal{C}}\lambda_c c : \lambda_c \geq 0, \sum_{c\in\mathcal{C}}\lambda_c = 1\right\} \tag{3.47}$$

Le décodage LP a été introduit par Feldman et al. [69] comme une approximation du décodage ML. Le polytope des mots de code a été réduit à un autre polytope $\mathcal{P}(\mathrm{H})$ appelé *fundamental polytope* par Vontobel et Koetter [70]

$$\mathcal{P}(\mathrm{H}) = \bigcap_{j=1}^{m} conv(\mathcal{C}_j), \tag{3.48}$$

où $\mathcal{C}_j \triangleq \{\mathrm{c} \in \{0,1\}^n : \langle \mathrm{c}, h_j \rangle = 0\}$ est un code local et h_j désigne la $j^{ème}$ ligne de H. Notons que $conv(\mathcal{C}) \subseteq \mathcal{P}(\mathrm{H})$. Le problème LP est donc décrit par :

$$\begin{aligned} \arg\min_{\mathrm{c}} \quad & \langle \mathrm{c}, \gamma \rangle \\ s.t. \quad & \mathrm{c} \in \mathcal{P}(\mathrm{H}) \end{aligned}, \tag{3.49}$$

Le polytope $\mathcal{P}(\mathrm{H})$ peut être aussi décrit par un ensemble d'inégalités linéaires, appelées *contraintes de parité* [69] :

$$\sum_{i \in V} (1 - c_i) + \sum_{i \in \mathcal{N}(j) \setminus V} c_i \geq 1 \ \forall \ V \subseteq \mathcal{N}(j) : |V| \ est \ impair, \tag{3.50}$$

où l'ensemble des voisins $\mathcal{N}(j) \subseteq \{1, \dots, n\}$ représente les indices des nœuds de variables qui sont directement connectés au nœud de contrôle j, $\mathcal{N}(j) = \{i : h_{ji} = 1\}$. Notons que les contraintes de parité et les *contraintes box linéaires* $0 \leq c_i \leq 1$ décrivent exactement le polytope fondamentale $\mathcal{P}(\mathrm{H})$ [66, Théorème 4]. Ce qui se traduit par

$$\mathcal{P}(\mathrm{H}) = \{\mathrm{c} \in \mathbb{R}^n | \ \mathrm{cA}^{\mathrm{T}} \leq \mathrm{b}, 0 \leq c_i \leq 1\}, \tag{3.51}$$

où la matrice A et le vecteur b sont tels que $\mathrm{cA}^{\mathrm{T}} \leq \mathrm{b}$ décrit les contraintes de parité. Ainsi, le décodage LP peut également être formulé comme suit

$$\begin{aligned} \arg\min_{\mathrm{c}} \quad & \langle \mathrm{c}, \gamma \rangle \\ s.t. \quad & 0 \leq c_i \leq 1, \qquad \forall \ i = 1, \dots, n \\ & \sum_{i \in V} (1 - c_i) + \sum_{i \in \mathcal{N}(j) \setminus V} c_i \geq 1, \\ & \forall \ j = 1, \dots, m, \ V \subseteq \mathcal{N}(j) \ : \ |V| \ est \ impair \end{aligned} \tag{3.52}$$

Le polytope $\mathcal{P}(\mathrm{H})$ contient des sommets *intégraux* et *non-intégraux*. Les sommets intégraux correspondent exactement aux mots de code de $\mathcal{C}$ [69]. Lorsque le solveur LP fournit une

solution intégrale, celle-ci est assurée d'être le mot de code ML (c'est la propriété de ML-certificat). Cependant, le nombre de contraintes introduites dans $\mathcal{P}(\mathrm{H})$ reste toujours exponentiel en fonction du degré[2] maximal de la matrice de contrôle de parité H $\mathrm{O}\left(2^{\max_j |N(j)|}\right)$. Par conséquent, pour les codes avec une matrice de contrôle de parité H à forte densité, comme les codes polaires, la description de $\mathcal{P}(\mathrm{H})$ est difficile à appliquer dans la pratique. En outre, le Lemme 2 dans [29] stipule que le décodeur LP échoue sur le polytope fondamental $\mathcal{P}(\mathrm{H})$ pour les codes à forte densité. Pour résoudre ces problèmes, un autre polytope convexe $\mathcal{P}(\mathrm{H}_{\mathcal{P}})$, basé sur le facteur de graphe Gph_S [25], a été proposée, où $\mathrm{H}_{\mathcal{P}}$ représente la matrice découlant de Gph_S avec $n\,logn = |J|$ lignes et $n(1 + logn) = |I|$ colonnes qui représentent respectivement les nœuds de contrôle et de variable. Un exemple de facteur de graphe est représenté sur la Figure 3.13 pour une longueur de bloc $n = 2^3$ [25]. Les nœuds de variable u, auxiliaires $a = \{a_{l,t}\}$, $l = \{1, 2, \dots, 8\}, t = \{1, 2\}$ et c sont binaires mais, pour des raisons de relaxation, ils sont considérés comme étant dans l'intervalle $[0, 1]$. Les cercles dans le graphe représentent les nœuds de variable et les carrés sont des nœuds de contrôle. Notons que chaque nœud de contrôle est connecté à deux ou trois nœuds de variable ; on parle de nœuds de contrôle de degré 2 et degré 3.

$$\mathcal{P}(\mathrm{H}_{\mathcal{P}}) = \left(\bigcap_{j \in J} \mathcal{P}_j(\mathrm{H}_{\mathcal{P}})\right) \cap T, \tag{3.53}$$

où $\mathcal{P}_j(\mathrm{H}_{\mathcal{P}})$ est un polytope local minimal et T est le plan de coupure défini en mettant toutes les frozen variables dans u à 0 ($\forall\, i \in A, u_i = 0$). Pour chaque nœud $j \in J$ avec les voisins $\mathcal{N}(j) = \{s_1, s_2, s_3\}$, $\mathcal{P}_j(\mathrm{H}_{\mathcal{P}})$ est défini par les inégalités :

$$\begin{aligned} &0 \le s_l \le s_q + s_t, \qquad l, q, t \in \{1,2,3\} : \ l \ne q \ne t \\ &s_1 + s_2 + s_3 \le 2 \end{aligned} \tag{3.54}$$

Pour les nœuds de contrôle $j \in J$ de degré 2 avec $\mathcal{N}(j) = \{s_1, s_2\}$, $\mathcal{P}_j(\mathrm{H}_{\mathcal{P}})$ est défini par :

$$\begin{aligned} &0 \le s_l \le 1, \qquad f \in \{1,2\} \\ &s_1 = s_2 \end{aligned} \tag{3.55}$$

[2] On appelle degré d'un nœud de contrôle le nombre de nœuds de variable auxquels il est connecté.

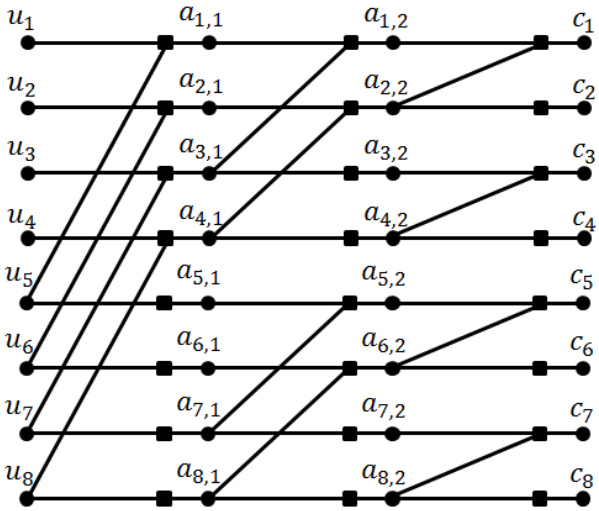

Figure 3.13 : Facteur graphe Gph_S du code polaire avec $n = 2^3$.

Le décodeur polaire LP devient :

$$\begin{aligned} &\arg\min_{\mathrm{c}} \quad \langle \mathrm{c}, \gamma \rangle \\ &s.t. \quad \mathrm{v} \in \mathcal{P}(\mathrm{H}_{\mathcal{P}}) \subseteq [0,1]^{n(1+logn)} \end{aligned} \quad , \tag{3.56}$$

où $\mathrm{v} = (\mathrm{c}, a, \mathrm{u})$, c est défini par les $c_i = v_i, i \in \{1, \dots, n\}$, et correspond au nœuds de variable mots de code. La projection $\bar{\mathcal{P}}(\mathrm{H}_{\mathcal{P}})$ de $\mathcal{P}(\mathrm{H}_{\mathcal{P}})$ est défini par

$$\bar{\mathcal{P}}(\mathrm{H}_{\mathcal{P}}) = \{\mathrm{c} \in \{0,1\}^n | \exists\, (a, \mathrm{u}) : (\mathrm{c}, a, \mathrm{u}) = \mathrm{v} \in \mathcal{P}(\mathrm{H}_{\mathcal{P}})\}. \tag{3.57}$$

Si la solution du décodeur LP de $\mathcal{P}(\mathrm{H}_{\mathcal{P}})$ projeté sur $\bar{\mathcal{P}}(\mathrm{H}_{\mathcal{P}})$ est intégral, elle est assurée d'être le mot de code ML [29, Lemme 3]. En plus $\bar{\mathcal{P}}(\mathrm{H}_{\mathcal{P}}) \subseteq \mathcal{P}(\mathrm{H})$. Les polytopes $\mathcal{P}(\mathrm{H})$ et $\mathcal{P}(\mathrm{H}_{\mathcal{P}})$ sont définis par un grand nombre de contraintes. D'où, la complexité du problème LP croit de façon exponentielle avec le degré maximum des nœuds de contrôle, δ_{max} [71]. Cela devient plus significatif dans le cas d'un code de forte densité où δ_{max} augmente avec la longueur du code n. Taghavi et Siegel ont montré que la relaxation LP de Feldman possède quelques propriétés qui leur ont permis de résoudre le problème d'optimisation en utilisant un nombre de contraintes beaucoup plus petit.

3.6.5 Décodage Adaptative LP (ALP) et codes polaires

3.6.5.1 Principe du décodage adaptative ALP

Un *cut* ou *contrainte violée* d'un vecteur $\mathrm{c} \in [0,1]^n$ sur j est défini par un ensemble $V \subseteq \mathcal{N}(j)$ de cardinalité impaire tel que les contraintes d'inégalités de relaxation ne sont pas

vérifiées [71]. Une *contrainte active* est une contrainte satisfaite avec égalité. Par exemple, une contrainte qui génère une violation de (3.50) signifierait que

$$\sum_{i \in V} c_i - \sum_{i \in \mathcal{N}(j) \setminus V} c_i > |V| - 1 \tag{3.59}$$

Dans [71], un algorithme efficace pour trouver des violations a été proposé. Basé sur cet algorithme, un autre algorithme ACG-ALP (Adaptive Linear Programming with Adaptive Cut-Generation) (Algorithme 2 dans [72]) a été proposé. Des modifications apportées au décodeur ACG-ALP, qui le rend approprié au décodage des codes polaires, ont été proposées dans [30]. Pour ALP toutes les contraintes de contrôle de parité ne sont pas ajoutées en même temps au début, comme dans LP [69], mais plutôt de façon itérative. Le solveur ALP fonctionne avec un nombre minimal de contraintes, et en fonction de la solution, seules les contraintes « utiles », qui génèrent une violation pour la solution courante, sont ajoutées de manière adaptative. Cette procédure est répétée jusqu'à l'obtention d'une solution intégrale ou qu'aucune autre violation ne se présente. Le décodeur ALP commence spécifiquement en résolvant le LP avec ces premières contraintes :

$$\begin{array}{lll} c_i \geq 0 & si & \gamma_i \geq 0 \\ & ou & \\ c_i \leq 1 & si & \gamma_i < 0 \end{array} \text{, pour tout } i \in \{1, 2, \ldots, n\} \tag{3.60}$$

La solution de ce problème LP initial coïncide avec la sortie ferme d'un décodage des valeurs de LLR reçues. Si la solution optimale est intégrale, alors elle correspond au mot de code ML sinon une erreur est générée en sortie. Le décodeur ALP converge avec moins de contraintes que le décodeur LP.

3.6.5.2 Décodage ALP dans le contexte des codes polaires

Un code polaire peut être défini en utilisant la matrice $H_{\mathcal{P}}$ correspondante au facteur de graphe avec les bits sources fixés ou en utilisant la matrice de contrôle de parité H. La disponibilité de ces deux représentations motive l'idée de modifier le décodeur ACG-ALP pour l'appliquer aux codes polaires et améliorer ainsi les performances par rapport au décodeur LP. Deux méthodes sont utilisées dans [30] pour appliquer ALP aux codes polaires. Taranalli et al. ont étudié quatre manières d'utiliser les représentations du facteur de graphe et de la matrice de contrôle de parité dans le décodeur ACG-ALP [30] et ont proposé quatre variantes. Les décodeurs 1, 3, 4, ont les mêmes performances en termes de taux d'erreur par trame (FER) et

sont plus performants que le décodeur 2. En outre, le décodeur ACG-ALP 4 possède la plus faible complexité temporelle. Par conséquent, les auteurs ont appliqué ce décodeur aux codes polaires [30, Algorithme 1]. Pour réduire la complexité du décodage, les auteurs ont proposé une réduction du facteur de graphe original qu'ils ont nommé RFG (Reduced Factor Graph) $G_{\mathcal{R}}$. La matrice de contrôle de parité $H_{\mathcal{R}}$ correspondante à ce graphe est composée uniquement de nœuds de contrôle de degré 3 [30, Lemme 2]. Le décodeur LP a la propriété de ML-certificat (c'est-à-dire lorsque le décodage réussit la solution trouvée est le mot de code que donnerait le décodage ML) sur les polytopes $\mathcal{P}(H_{\mathcal{P}})$ [29] et $\mathcal{R}(H_{\mathcal{R}})$ [30].

3.7 Codes polaires systématiques

Un code est sous forme systématique si le mot d'information encodé se retrouve entièrement et explicitement dans le mot de code correspondant. Selon la théorie des codes, tout code en bloc linéaire peut être transformé en un code systématique équivalent. Ainsi, Arikan a introduit les codes polaires, qu'il avait initialement défini comme étant non-systématiques [25], sous une forme systématique [73]. Les processus d'encodage et de décodage associés permettent de conserver la faible complexité temporelle. Les codes polaires systématiques peuvent être encodés en utilisant la matrice génératrice G_n, soit avec ou sans la matrice de permutation bit-reversal B_n devant la puissance de Kronecker $G_2^{\otimes p}$. Cependant, le processus reste le même.

3.7.1 Processus d'encodage polaire systématique

Nous examinons comment les bits d'information peuvent être présentés à la sortie du décodeur dans l'ordre dans lequel ils ont été présentés à la source. L'idée est de définir un codage polaire qui, à un mot source u, fait correspondre un mot de code $c = (c_A, c_{A^c})$, $c_A = \{c_i | i \in A\}$ tel que $c_A = u_A$ les bits d'information. Nous devrons donc chercher une matrice génératrice systématique G_{n-syst} qui permettrait ce mappage. La détermination d'une telle matrice n'étant pas chose aisée, une forme explicite de G_{n-syst} n'est pas encore proposée, à notre connaissance. Nous savons, par ailleurs, que la matrice génératrice G_n est inversible et égale à son propre inverse. Ce qui implique que $c = uG_n$ est équivalente à $u = cG_n$. Par conséquent $uG_nG_n = u = (u_A, u_{A^c})$. Partant de ce constat, nous pouvons dire qu'un code systématique est obtenu si nous parvenons à intercaler entre les matrices une opération qui empêche d'avoir u_{A^c} dans l'expression finale mais plutôt d'avoir $c_{A^c} = f(u)$. Nous verrons, que le fait de forcer les frozen bits à prendre les valeurs fixées à 0 entre les deux matrices,

permet d'obtenir ce résultat. Ainsi, le codage polaire systématique sera composé de trois étapes résumées comme suit :

$$\mathrm{u} = (\mathrm{u}_A, 0_{A^c}) \xrightarrow{\times \mathbf{G_n}} \mathrm{c}' = (\mathrm{c}'_A, \mathrm{c}'_{A^c}) \xrightarrow{\boldsymbol{forçage}} \mathrm{u}' = (\mathrm{c}'_A, \mathbf{0}_{A^c}) \xrightarrow{\times \mathbf{G_n}} \mathrm{c} = (\mathrm{u}_A, \mathrm{c}_{A^c}) \qquad (3.61)$$

Pour illustrer le codage systématique, on donne deux exemples. On considère d'abord le cas d'un code polaire $PC(4,3)$ sans la matrice de permutation B_4 devant $\mathrm{G}_2^{\otimes 2}$ (donc $\mathrm{G}_4 = \mathrm{G}_2^{\otimes 2}$) (Figure 3.14) [74]. Le processus de codage systématique est décrit comme suite. Tout d'abord le mot u est codé classiquement. Le mot de code intermédiaire obtenu est noté c'. Ensuite, les frozen bits c'_A de c' (ici c'est la première ligne c'_1) sont forcés à 0. Le message intermédiaire obtenu, noté u', est ensuite codé classiquement. Le mot de code final c est bien composé des bits d'informations originaux et d'un bit de redondance. On obtient donc bien un mot de code systématique.

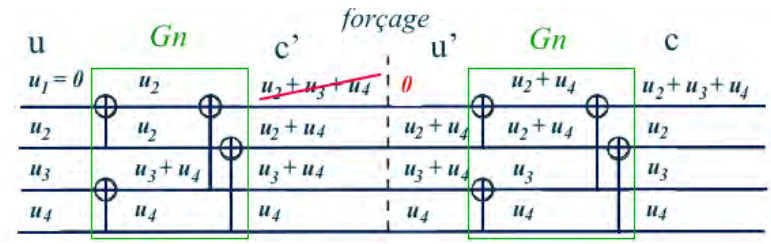

Figure 3.14 : Encodage polaire systématique pour $n = 4$ et $R = 3/4$.

Nous donnons un autre exemple pour $PC(8,5)$ représentant l'encodage polaire systématique avec la matrice de permutation [73] (Figure 3.15) B_8 devant $\mathrm{G}_2^{\otimes 3}$ (donc $\mathrm{G}_8 = \mathrm{B}_8\, \mathrm{G}_2^{\otimes 3}$).

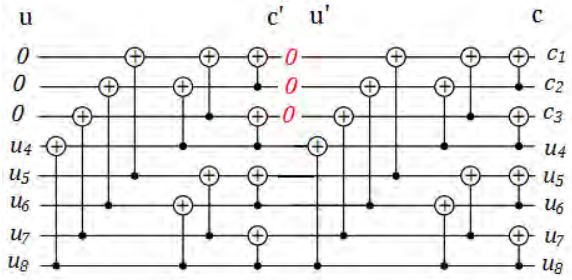

Figure 3.15 : Encodage systématique de $PC(8,5)$ avec la matrice de permutation B_8.

Le mot source, composé des bits d'information et de redondance (frozen bits), est encodé par la relation $\mathrm{c}' = \mathrm{u}\mathrm{G}_2^{\otimes 3}$ (sans la matrice de permutation B_8). L'étape suivante consiste à forcer à 0 les bits de c' correspondant aux frozen positions A^c pour obtenir u'. Et, dans la troisième et

dernière étape, on effectue une nouvelle l'encodage de u′; ce qui donne finalement le mot de code systématique c contenant explicitement les bits d'information u_A.

3.7.2 Décodage des codes polaires systématiques

Le décodeur polaire systématique est exactement le même que le décodeur non systématique, sauf que, à la fin il retourne $\hat{c}_A$ du codage non systématique au lieu de $\hat{u}_A$ donné par le cas systématique. Le décodage peut être vu comme se déroulant en deux étapes successives. La première étape correspond à un décodage classique utilisant une des méthodes de décodage existantes (SC, SCL, LP, ALP, BP, SCAN, …) et la seconde consiste à encoder le mot $\hat{c}_A$ pour obtenir $\hat{u}_A$.

$$r \xrightarrow{décodage\ classique} \hat{c}_A \xrightarrow{encodage\ polaire} \hat{u}_A \tag{3.62}$$

3.7.3 Performances des codes polaires systématiques

Le principal avantage des codes polaires systématiques est qu'ils sont plus robustes contre les erreurs de propagation lorsque le décodage SC est appliqué. Les résultats expérimentaux ont montré que, par rapport au code polaire original (non systématique), le code polaire systématique a le même taux d'erreur bloc ou trame BLER (BLoc Error Rate) ou FER (Frame Error Rate) et améliore le Taux d'Erreur Binaire BER (Bit Error Rate) pour un code polaire $PC(256,128)$. Ce résultat est surprenant car les deux décodeurs estime $\hat{c}$ de façon similaire. Si des erreurs sont commises lors de l'estimation de $\hat{u}$, ces erreurs se propagent de gauche à droite à travers le décodeur, de sorte que l'on puisse s'attendre à obtenir encore plus d'erreurs à droite où $\hat{c}$ est estimé. On constate plutôt que les erreurs s'annulent, ainsi $\hat{c}$ contiendra moins d'erreurs. Cependant, théoriquement le mécanisme derrière les performances des codes polaires systématiques par rapport au cas non-systématique n'est pas toujours clair.

Conclusion

La construction des codes polaires se base sur la polarisation de canal. Les codes polaires constituent des codes correcteurs d'erreurs en bloc linéaire qui présentent de très bonnes performances en termes de capacité, de probabilité d'erreurs et de complexité. L'algorithme SC est la première méthode de décodage proposée pour les codes polaires. Ce type de décodage utilise les propriétés spécifiques des codes polaires. Cependant ses performances en termes de

probabilité d'erreurs de décodage pour des codes polaires à faible et moyenne longueur de bloc ne sont pas optimales. Ainsi, d'autres types de décodage sont proposés par la suite pour améliorer les performances. Parmi ces décodages les plus connus sont BP, LP, ALP et SCL. La théorie et la pratique des codes polaires sont à leur début mais, du fait de leurs performances, ces codes sont de plus en plus explorés dans les domaines dans lesquels les codes correcteurs d'erreurs sont utilisés. Nous allons nous intéresser particulièrement à leur application dans le domaine de la stéganographie numérique.

Chapitre 4: APPLICATION DES CODES POLAIRES EN STEGANOGRAPHIE POUR MINIMISER L'IMPACT D'INSERTION

Introduction

L'introduction des codes en stéganographie a été une grande avancée dans la stéganographie avec l'application de plusieurs codes comme Hamming, BCH, RS, LDGM, LDPC, STC, … Dans la continuité de ces travaux et suite à l'optimalité des codes polaires pour le problème d'insertion PLS soulignée par Filler et al. dans [19], nous présentons dans ce chapitre deux méthodes pour la définition d'un schéma de stéganographie basée sur les codes polaires. Nous définissons, dans un premier temps, un schéma appelé PCS (Polar Coding Steganography) pour le cas où tous les pixels de l'image ont la même sensibilité à la modification (profil constant) qui sera ensuite adapté au cas où certains pixels devront rester inchangés après insertion (papier mouillé). Ce schéma est composé de deux phases. La première phase donne un vecteur stégo solution du problème de stéganographie. Dans le cas où cette solution n'est pas optimale, nous utiliserons la programmation linéaire pour trouver la solution optimale. Dans la deuxième partie du chapitre, nous proposons une autre méthode permettant de réduire la complexité comparée au premier. Cette méthode comporte deux approches ; la première est basée sur les tables de correspondances et la seconde exploite la forme du syndrome en établissant une connexion entre sa valeur décimale et la position des modifications sur le médium de couverture pour trouver plus facilement le stégo médium. Nous avons également appliqué les schémas proposés sur des images numériques dont les pixels sont permutés avant insertion en exploitant la matrice de permutation bit-reversal utilisée dans la construction des codes polaires. Cela permet d'étendre les changements sur des pixels isolés de l'image et rendre ainsi plus sûre le stégo-système.

4.1 Construction des codes polaires pour la stéganographie

Considérons un canal B-DMC W et les canaux $W_n^{(i)}$ obtenus par polarisation de canal. La construction des codes polaires peut se résumer en trois étapes comme le représente la figure ci-dessous [75]:

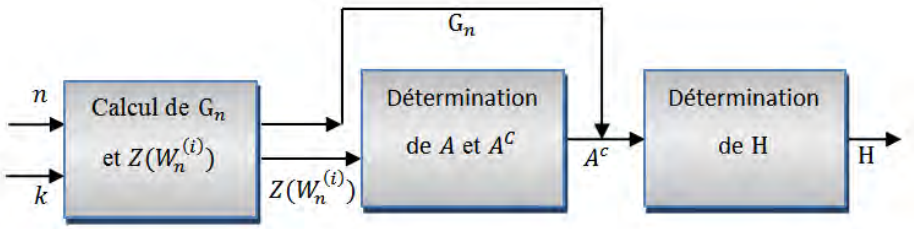

Figure 4.1 : Construction des codes polaires pour la stéganographie.

Etape 1 : *Calcul des paramètres de fiabilité des canaux (ou paramètres de Bhattacharyya).*

Nous avons vu dans le chapitre précédent que le choix de l'ensemble des bits d'information A et de celui des bits de redondance A^c dépend des valeurs des paramètres de fiabilité $Z(W_n^{(i)})$. Plus précisément nous choisissons les canaux les plus fiables (ayant les plus petites valeurs de paramètres de fiabilité) pour transporter notre information. Les indices de ces canaux forment l'ensemble A et celles des autres canaux constituent l'ensemble A^c. Rappelons que pour un canal binaire symétrique (BSC) nous avons les relations suivantes :

$$Z\left(W_{2n}^{(2i-1)}\right) \leq 2Z\left(W_n^{(i)}\right) - Z\left(W_n^{(i)}\right)^2, \tag{4.1}$$

$$Z\left(W_{2n}^{(2i)}\right) = Z\left(W_n^{(i)}\right)^2, \tag{4.2}$$

avec une égalité dans (4.1) pour le cas d'un canal BEC (canal d'effacement binaire).

Le canal stéganographique est considéré comme étant un BSC. Nous considérons que notre canal de synthèse W engendre des canaux $W_n^{(j)}$ tels que nous ayons l'égalité dans la relation (4.1). Ainsi nous pouvons calculer tous les paramètres de fiabilité de façon récursive. Les relations (4.1) et (4.2) deviennent [75] :

$$Z\left(W_n^{(j)}\right) = 2Z\left(W_{n/2}^{((j+1)/2)}\right) - Z\left(W_{n/2}^{((j+1)/2)}\right)^2 \quad si\ j\ est\ impair \tag{4.3}$$

$$Z\left(W_n^{(j)}\right) = Z\left(W_{n/2}^{(j/2)}\right)^2 \quad si\ j\ est\ pair \tag{4.4}$$

La valeur initiale de ces relations récursives est donnée par

$$Z\left(W_1^{(1)}\right) = Z(W) = \sum_{y\epsilon\{0,1\}} \sqrt{\mathrm{W}(r|0)\mathrm{W}(r|1)} = 2\sqrt{\mathrm{W}(0|0)\mathrm{W}(0|1)} = 2\sqrt{p_e(1-p_e)} \tag{4.5}$$

où p_e est la probabilité d'erreur du canal W, $p_e = W(0|1) = W(1|0)$ et $1 - p_e = W(0|0) = W(1|1)$.

Etape 2 : *Détermination des ensembles des bits d'information et des bits de redondances.*

Après avoir calculé les paramètres de Bhattacharyya l'étape suivante consiste à choisir les canaux ayant les paramètres de fiabilités les plus faibles (les canaux les plus fiables) pour les bits d'information. Les indices de ces canaux forment l'ensemble des bits d'information A. Son cardinal est égal à la dimension k du code polaire considéré et les autres canaux (au nombre de $n-k$) transportent les bits de redondance (souvent considérés comme égaux à 0). Leurs indices constituent l'ensemble fixé A^c.

Etape 3 : *Génération de la matrice de contrôle de parité*

Arikan n'a pas parlé de matrice de contrôle de parité dans la construction des codes polaires car le décodage se fait directement sans vérification de la présence d'erreurs (correction automatique des erreurs). Un élément commun dans la construction de tout schéma stéganographique est la matrice de contrôle de parité H. En effet, elle est utilisée aussi bien à l'insertion qu'à l'extraction du message. Pour déterminer la matrice de contrôle de parité d'un code polaire, nous pouvons utiliser le lemme de Goela et al. [29, lemme 1] : si les bits du vecteur de redondance $\mathrm{u}_{\mathrm{A}^c}$ sont fixés à 0 alors la matrice de contrôle de parité est donnée par les colonnes de la matrice génératrice G_n du code polaire dont les indices sont dans A^c.

Rappelons qu'un code polaire de dimension k et de longueur $n = 2^p$ a une matrice génératrice de taille (n, n) et se présente sous la forme $\mathrm{G}_n = \mathrm{B}_n \mathrm{G}_2^{\otimes p}$.

Exemple 4.1 : Pour $p = 2$ nous avons :

$$\mathrm{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathrm{G}_2^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{et} \quad \mathrm{G}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{4.6}$$

avec B_4 la matrice de permutation des lignes 2 et 3.

Le calcul des paramètres de fiabilité donne :

$$Z\left(W_4^{(1)}\right) = 0.9744 \geq Z\left(W_4^{(2)}\right) = 0.7056 \geq Z\left(W_4^{(3)}\right) = 0.5904 \geq Z\left(W_4^{(4)}\right) = 0.1296.$$

Pour $k = 1$ nous choisissons un canal (le plus fiable) pour les informations donc $A = \{4\}$ et $A^c = \{1, 2, 3\}$. La matrice de contrôle de parité correspondante sera définie comme suit :

$$\mathrm{H}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ et donc } \mathrm{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}. \quad (4.7)$$

Nous utiliserons dans la définition de notre schéma les outils d'optimisation par programmation linéaire.

4.2 Premier schéma de stéganographie basé sur les codes polaires

4.2.1 Première étape du schéma

Dans cette partie nous considérons que la modification de n'importe quel pixel produit la même distorsion (profil constant). Nous nous intéressons ainsi au nombre de bits modifiés pendant l'insertion ce qui correspondra à la distorsion totale produite par l'insertion du message secret dans l'objet de couverture. Nous allons nous inspirer du travail de Filler et al. [12] dans lequel le schéma stéganographique proposé est fortement lié à la forme de la matrice de contrôle de parité du code STC utilisé. En effet, en observant la matrice de contrôle de parité H des codes polaires et sa transposée H^{T}, nous pouvons faire les remarques suivantes [75] :

- les colonnes de H sont deux à deux indépendantes ;
- si on parcourt les colonnes de H^{T}, la position à laquelle on rencontre le premier coefficient non nul (égal à 1) diffère de celle des autres colonnes de H^{T} ;
- de plus en partant de la dernière colonne et en commençant par la première ligne, la position à laquelle on rencontre le premier coefficient non nul (égal à 1) est la première rencontrée sur cette ligne (donc la dernière position égale à 1 sur cette ligne si on part de la gauche).

Partant de ces remarques et du produit matriciel $\mathrm{yH}^{\mathrm{T}} = \mathrm{m}$, nous aurons :

$$y_1 H_{1j}^T + y_2 H_{2j}^T + \cdots + y_n H_{nj}^T = m_j \; ; \quad \text{pour } j = n-k, \cdots, 1. \quad (4.8)$$

Soit i la position du premier 1 rencontré sur la colonne j ($H_{ij}^T = 1$), le système d'équations binaires précédent devient :

$$y_i = y_{i+1} H_{(i+1),j}^T + \cdots + y_n H_{nj}^T + m_j \; . \quad (4.9)$$

Pour déterminer y_i dans l'équation précédente, nous devons auparavant trouver les y_{i+l} tels que $H^T_{(i+l),j} = 1$. Nous allons supposer que ces positions correspondent aux positions verrouillées de x. Dans ce cas $x_{i+l} = y_{i+l}$. De ce fait, avant de calculer les éléments y_k du vecteur stégo initial y, nous allons d'abord affecter à y, comme valeur initiale, le vecteur de couverture (y = x). Les modifications de certaines positions y_i de y se feront au fur et à mesure du parcours des colonnes. A la suite de ce processus, nous nous retrouvons avec un vecteur stégo y découlant de t modifications du vecteur de couverture x. Pour mieux illustrer, nous allons utiliser un exemple.

Exemple 4.2 :

Considérons le vecteur de couverture $x = (x_1, \cdots, x_8)$ qui doit contenir le message $m = (m_1, \cdots, m_4)$ pour donner le vecteur stégo $y = (y_1, \cdots, y_8)$. Pour appliquer la première méthode ci-dessus nous utiliserons un code polaire de longueur $n = 2^p = 8\,(p = 3)$ et de dimension $k = 4$. L'ensemble des bits d'information et celui des bits fixés sont donnés par $A = \{4, 6, 7, 8\}$ et $A^c = \{1, 2, 3, 5\}$. La matrice de contrôle de parité est définie par :

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \text{ et sa transposée } H^T = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 1 & 0 & 0 & \mathbf{1} \\ 1 & 0 & \mathbf{1} & 0 \\ 1 & 0 & 1 & 1 \\ 1 & \mathbf{1} & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \quad (4.10)$$

A partir de la relation (4.9), nous avons le système suivant :

$$\begin{cases} y_2 = y_4 + y_6 + y_8 + m_4 \\ y_3 = y_4 + y_7 + y_8 + m_3 \\ y_5 = y_6 + y_7 + y_8 + m_2 \\ y_1 = y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + m_1 \end{cases} \quad (4.11)$$

Le vecteur y doit être initialisé au vecteur de couverture x. Prenons la première équation, le calcul du coefficient y_2 nécessite la connaissance des coefficients y_4, y_6 et y_8. Nous considérons que ces positions sont fixées donc $y_4 = x_4$, $y_6 = x_6$ et $y_8 = x_8$. De manière identique, nous fixons le coefficient $y_7 = x_7$ pour calculer y_3 en utilisant y_4 et y_8 qui ont été fixés à l'étape précédente. Le calcul de y_5 se fait en utilisant y_6, y_7 et y_8 fixés précédemment. Le dernier coefficient y_1 est déterminé à partir de coefficients qui sont soit fixés soit déjà

calculés. D'où la nécessité de commencer à partir de la dernière colonne de H^T sinon nous ne pourrions pas calculer y_1 sans fixer tous les autres coefficients $y_i,\ 2 \leq i \leq 8$; ce qui serait aberrant.

La matrice de contrôle de parité est la même que celle donnée dans l'exemple précédent. Pour un vecteur de couverture $x = (0,1,1,0,1,0,1,0)$ et un message $m = (1,0,0,1)$ nous avons le vecteur stégo suivant $y = (1,1,1,0,1,0,1,0)$ et le vecteur de modification correspondant $e = (1,0,0,0,0,0,0,0)$. Nous avons inséré 4 bits de message pour n'en modifier qu'un seul. D'où une efficacité d'insertion de 4.

Notons que le schéma décrit ci-dessus présente un taux d'insertion pouvant aller jusqu'à 100% (nous pouvons insérer un nombre de bits égal à la taille du support) mais sa densité de changement serait aussi grande dans ce cas. L'application de la méthode nous donne une solution vérifiant $yH^T = m$ mais elle n'est pas forcément la meilleure. Pour tester l'optimalité de la solution obtenue, nous comparons le nombre de modifications avec la valeur $(n-k)/2$. Si c'est inférieur à $(n-k)/2$ alors la solution trouvée avec la première méthode est optimale. Afin de s'assurer de trouver la solution optimale (celle qui donne le vecteur stégo le plus proche du vecteur de couverture x vérifiant $yH^T = m$) nous définissons, à partir de la solution déjà obtenue, une méthode qui permet de converger vers la cette solution optimale.

4.2.2 Deuxième étape (optimisation de la première solution)

L'objectif avec cette étape est de trouver le vecteur de couverture y le plus proche de x qu'on puisse avoir en utilisant le code polaire $C(n,k)$. Soient y_p le vecteur stégo trouvé en appliquant la première étape. La question est de savoir comment trouver, à partir de y_p, le vecteur optimal y_{opt} vérifiant toujours $yH^T = m$. Nous devons donc définir un algorithme qui, initialisé à y_p, converge vers y_{opt}. Autrement dit en considérant les vecteurs d'erreurs, l'algorithme devra, à partir de e_p, donner e_{opt} le vecteur d'erreurs correspondant à y_{opt}. Nous cherchons ainsi le vecteur d'erreur e de poids minimal vérifiant $eH^T = m - xH^T = s$.

Récapitulation :

- nous avons une ***solution de départ*** $\mathbf{e_p} \longmapsto$ solution initiale,
- nous cherchons un vecteur **e** ***de poids minimal*** $\longmapsto$ problème de minimisation,
- vérifiant le système $\mathbf{eH^T = m - xH^T = s} \longmapsto$ contraintes.

En considérant ces trois points nous avons un problème d'optimisation, notamment un problème de minimisation, sous contraintes d'égalité avec comme solution initiale $\mathrm{e_p}$. Notre problème d'optimisation est défini pour des vecteurs binaires et sans contraintes d'inégalités. Puisque nous cherchons à minimiser le poids du vecteur e, donc pour définir le produit scalaire de la fonction objectif nous considérons le vecteur coût unitaire ($\gamma = \{1\}^n$). Chercher le vecteur e de poids minimal revient à chercher le vecteur réalisant le minimum du produit scalaire avec le vecteur γ. Ce qui nous donne comme fonction objectif $f(\mathrm{e}) = \langle \gamma, \mathrm{e} \rangle$ (produit scalaire entre γ et e); c'est la fonction à minimiser. A partir de là, nous pouvons définir notre problème d'optimisation de la manière suivante :

$$\begin{aligned} \arg\min_{\mathrm{e}} \quad & \underset{s.c}{f}(\mathrm{e}) = \langle \gamma, \mathrm{e} \rangle = \gamma^{\mathrm{T}}\mathrm{e} \\ & \begin{cases} \mathrm{e} \in \{0,1\}^n \ \text{un vecteur binaire} \\ \mathrm{eH}^{\mathrm{T}} = \mathrm{m} - \mathrm{xH}^{\mathrm{T}} = \mathrm{s} \\ \mathrm{e}_p \ \textit{solution initile} \iff \mathrm{e}_p\mathrm{H}^{\mathrm{T}} = \mathrm{s} \end{cases} \end{aligned} \tag{4.12}$$

Ce problème est celui d'une optimisation linéaire sous contraintes d'égalités écrit sous forme standard avec une contrainte supplémentaire ; le vecteur e est constitué d'éléments binaires. Nous pouvons la résoudre par plusieurs méthodes de résolution des problèmes de programmation linéaire telle que la méthode du simplexe et celle des points intérieurs. Ainsi, l'application de la méthode d'optimisation fournit la solution optimale de notre problème stéganographique. Ce que nous allons illustrer avec deux exemples.

Exemple 4.3:

Soit le message $\mathrm{m} = (0,0,1,0)$ et le vecteur de couverture $\mathrm{x} = (0,1,0,1,0,0,0,1)$. La première méthode donne un vecteur d'erreurs $\mathrm{e}_p = (1,1,0,0,1,0,0,0)$ et le vecteur stégo correspondant $\mathrm{y}_p = (1,0,0,1,1,0,0,1)$. L'optimisation de la solution donnée par la méthode fournit les résultats suivants : le vecteur d'erreurs optimal $\mathrm{e}_{opt} = (0,0,0,0,0,1,0,0)$ et vecteur stégo optimal $\mathrm{y}_{opt} = (0,1,0,1,0,1,0,1)$. La première étape donne une efficacité d'insertion $eft_{inst} = (n-k)/D = 4/3$, avec D la distorsion, tandis que la seconde offre une efficacité d'insertion $eft_{inst} = 4/1$.

Exemple 4.4:

Considérons le vecteur de couverture $x = (1, 0, 1, 1, 1, 0, 0, 1)$, le message $m = (0, 0, 1, 1)$. Les résultats de la première solution sont $e_p = (0, 1, 0, 0, 0, 0, 0, 0)$ et $y_p = (1, 1, 1, 1, 1, 1, 0, 1)$. La solution optimisée est $e_{opt} = (0, 1, 0, 0, 0, 0, 0,0)$ et $y_{opt} = (1, 1, 1, 1, 1, 1, 0, 1)$. Les deux étapes donnent la même solution avec une efficacité d'insertion égale à 4.

Le schéma peut être globalement résumé par la figure ci-dessous [75].

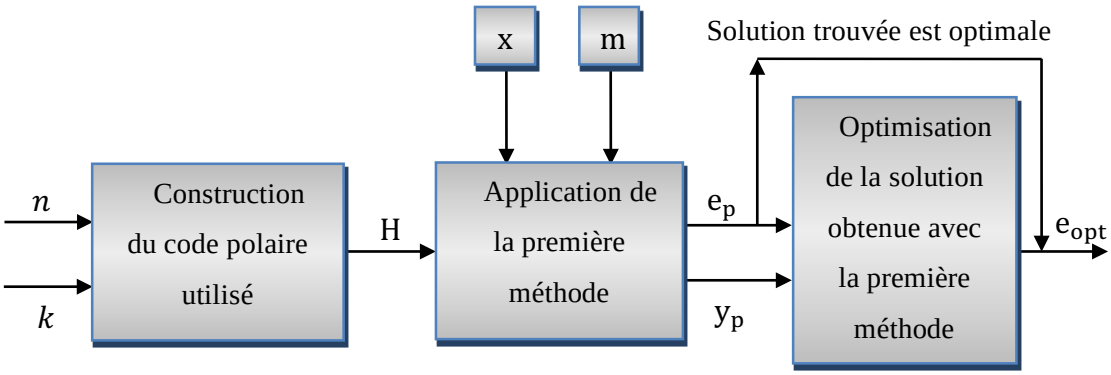

Figure 4.2 : Représentation schématique du schéma stéganographique proposé.

4.2.3 Calcul de l'efficacité d'insertion

Exemple 4.5 : ($p = 3$, $n = 2^p = 8$ et $k = 4$)

$$H=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (4.13)$$

En examinant les colonnes de H nous avons les égalités suivantes sur les colonnes H_j:

- $H_{.1} + H_{.2} = H_{.3} + H_{.4} = H_{.5} + H_{.6} = H_{.7} + H_{.8} = (0\ 0\ 0\ 1)^T$
- $H_{.1} + H_{.3} = H_{.2} + H_{.4} = H_{.5} + H_{.7} = H_{.6} + H_{.8} = (0\ 0\ 1\ 0)^T$
- $H_{.1} + H_{.4} = H_{.2} + H_{.3} = H_{.5} + H_{.8} = H_{.6} + H_{.7} = (0\ 0\ 1\ 1)^T$
- $H_{.1} + H_{.5} = H_{.2} + H_{.6} = H_{.3} + H_{.7} = H_{.4} + H_{.8} = (0\ 1\ 0\ 0)^T$ (4.14)
- $H_{.1} + H_{.6} = H_{.2} + H_{.5} = H_{.3} + H_{.8} = H_{.4} + H_{.7} = (0\ 1\ 0\ 1)^T$
- $H_{.1} + H_{.7} = H_{.2} + H_{.8} = H_{.3} + H_{.5} = H_{.4} + H_{.6} = (0\ 1\ 1\ 0)^T$
- $H_{.1} + H_{.8} = H_{.2} + H_{.7} = H_{.3} + H_{.6} = H_{.4} + H_{.5} = (0\ 1\ 1\ 1)^T$

Le syndrome $s = m - xH^T$ de taille 4 est égal :

- ✓ au vecteur nul ($s = (0,0,0,0)$) avec une probabilité de $\frac{1}{2^4} = \frac{1}{16}$
  ↦ aucune modification (0) du vecteur de couverture ;
- ✓ à une colonne de H avec une probabilité de $\frac{8}{16}$ (8 vecteurs différents représentant les colonnes de H sur les $2^4 = 16$ possibles)
  ↦ une (1) modification du support ;
- ✓ à la somme de deux colonnes de H avec une probabilité de $\frac{7}{16}$ (7 vecteurs obtenus par sommation deux à deux des différentes colonnes de H)
  ↦ deux (2) modifications du support.

Soit Z la variable aléatoire représentant le nombre de modifications apportées par l'insertion du message. Son support est $S = \{z_i = i\}_{i=\{0,1,2\}}$. La loi de probabilité est la suivante : $P(Z = z_1) = \frac{1}{16}$, $P(Z = z_2) = \frac{1}{2}$ et $P(Z = z_3) = \frac{7}{16}$. Par conséquent le nombre moyen de modifications est égal à l'espérance de Z notée par $E(Z)$:

$$nbre_{modif} = E(Z) = 0\times\frac{1}{16} + 1\times\frac{8}{16} + 2\times\frac{7}{16} = \frac{11}{8} = 1.375. \quad (4.15)$$

L'efficacité d'insertion est donc :

$$eft_{inst} = \frac{taille(\mathrm{m})}{nbre_{modif}} = \frac{4}{11/8} = \frac{32}{11} \cong 2{,}91. \quad (4.16)$$

Elle correspond à la distance moyenne du code polaire pour une charge relative $\alpha = \frac{m}{n} = \frac{4}{8} = \frac{1}{2}$. Cette valeur de l'efficacité d'insertion est bien supérieure à $\frac{n-k}{2} = 2$ et constitue la plus grande possible en utilisant un code binaire de longueur $n = 8$ et de dimension $k = 4$, pour un profil constant.

4.2.4 Condition d'optimalité du schéma proposé

Le schéma proposé n'est pas adapté pour des messages de taille inférieure ou égale à $p = \log_2(n)$ ($taille\ (\mathrm{m}) = m \leq p$). Par conséquent, les messages dont nous voulons que le traitement par ce schéma donne un nombre de modifications minimal doivent vérifier $taille\ (\mathrm{m}) = m > p$. Pour répondre à ce critère nous pouvons ajuster soit la taille du message soit celle du vecteur de couverture. Puisque le message est donné à l'avance, il serait plus

simple de choisir la deuxième option qui consiste à choisir le vecteur de couverture de telle sorte que sa taille n vérifie le critère $\log_2(n) > m$. Ce critère imposé est logique dans la mesure où pour $m \leq p$ certaines colonnes de la matrice de contrôle de parité H seront liées (elles seront identiques puisque nous sommes dans le cas binaire). Ce qui favorise davantage de modifications. Précisons que même des messages de taille $m \leq p$ peuvent être insérés et extraits à la réception mais le nombre minimal de modifications n'est plus garanti dans ce cas.

Exemple 4.6 :

Pour illustrer ce que nous venons de dire nous choisissons $p = 3$ ($n = 2^p = 8$). Si $k = 5$ ($m = n - k = 3 = p$) alors $A^c = \{1, 2, 3\}$, $A = \{4, 5, 6, 7, 8\}$ et

$$H=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (4.17)$$

Les colonnes successives de H sont 2 à 2 identiques (1 et 2, 3 et 4, etc.). Donc le nombre moyen de modifications est égal à :

$$nbre_{modif} = 0\times\frac{1}{8} + 1\times\frac{4}{8} + 2\times\frac{3}{8} = \frac{5}{8} = 1.25 \quad (4.18)$$

Alors que le code de Hamming donne un nombre moyen de modifications égal à $0\times\frac{1}{8} + 1\times\frac{7}{8} = \frac{7}{8} = 0{,}875$. Ce qui est plus petit que $1{,}25$ pour un même nombre de bits insérés.

4.2.5 Schéma de stéganographie à papier mouillé

Dans le schéma que nous venons de définir, nous avons considéré le cas d'un profil constant. Nous allons montrer, dans cette partie, que le schéma peut être adapté au cas du papier mouillé.

Dans le cas où les coûts de modifications $\rho_i = 1$ (profil constant), la minimisation de la distorsion D se résume à minimiser le nombre de modifications du vecteur de couverture. Par contre si les ρ_i sont quelconques dans $[0, \infty]$ nous devrons définir le schéma pour la minimisation du nombre de positions modifiées en tenant compte des contraintes de coûts ρ_i. Nous devons donc modifier les pixels aux niveaux desquels les modifications sont moins perceptibles (ayant les plus petites valeurs de ρ_i). Filler et al. ont défini trois types de profils

pour leur schéma de stéganographie [12] : le profil constant ($\rho_i(x) = 1$), le profil linéaire ($\rho_i(x) = 2x$) , et le profil carré ($\rho_i(x) = 3x^2$). Nous allons développer une technique de stéganographie afin de minimiser l'impact d'insertion pour une distorsion donnée.

Considérons un profil quelconque défini par $\rho = \{\rho_i\}_{1 \le i \le n}$[3] [13]. Notre objectif est d'adapter le schéma proposé à ce cas général des profils de distorsion. Filler et al. [12] ont proposé deux méthodes pour appliquer leur schéma aux cas des papiers mouillés. La première consiste à verrouiller un certain nombre $|\mathcal{J}|$ de positions (les éléments mouillés) de l'objet stégo et de ne modifier que les bits correspondant aux autres positions (éléments secs). La réussite d'une telle méthode dépend, bien entendu, du nombre de positions verrouillées et du type de codage utilisé. La valeur de $|\mathcal{J}|$ ne doit pas dépasser la dimension du code k [7], [8]. Dans le cas contraire la recherche du vecteur stégo ne donnerait aucune solution. La deuxième technique consiste à verrouiller les positions au niveau desquelles les changements seront les plus visibles. Filler et al. ont montré que cette méthode est bien adaptée dans la pratique. Si le nombre d'éléments mouillés est supérieur à k nous nous autorisons à en modifier certaines[4] . Pour proposer une méthode de stéganographie des codes polaires à papier mouillé nous allons utiliser cette dernière approche qui est plus pratique et plus adaptée à notre schéma.

Notre schéma est composé de deux parties dont la seconde améliore la première. Il s'avère donc nécessaire de voir comment chacune de ces deux parties est appliquée au cas du papier mouillé. La première méthode se base sur la relation $\mathrm{yH^T} = \mathrm{m}$ en exploitant la forme de la matrice de contrôle de parité du code polaire. Par conséquent elle est indépendante du type de profil considéré et s'applique ainsi de façon identique au cas du profil constant. Et dans ce cas si la solution offerte par cette méthode correspond à la celle optimale il ne sera pas nécessaire d'appliquer la seconde. Concernant la deuxième méthode dont l'implémentation dépend du type de profil utilisé, des modifications devront être apportées pour définir un schéma de stéganographie à papier mouillé. Le problème est le même que dans le cas du profil constant (problème d'optimisation, minimisation plus précisément). Nous utilisons le même principe de la programmation linéaire pour trouver notre solution optimale. La solution initiale et les contraintes n'ont pas changées. Ce qui change ici c'est la fonction objectif. En effet, dans le cas

[3] Les trois types de profils sus cités sont des cas particuliers de cette considération.

[4] Si nous considérons que le nombre d'éléments mouillés est égal à $|\mathcal{J}| > k$, nous pourrons en modifier $|\mathcal{J}| - k$.

du profil constant, l'objectif était de minimiser le poids de Hamming du vecteur de modifications e alors que pour un profil quelconque, le but est de minimiser la fonction de distorsion. Réécrivons la relation de distorsion et les fonctions d'insertion et d'extraction associées en fonction du vecteur de modifications e :

$$D(\mathrm{e}) = \sum_{i=1}^{n} \rho_i e_i, \tag{4.19}$$

où $|x_i - y_i| = e_i$ et $0 \leq \rho_i \leq \infty$ le coût de modification du LSB x_i d'un pixel en y_i.

$$\begin{aligned} Emb(\mathrm{x}, \mathrm{m}) &= \arg \min_{\mathrm{e} \in C(\mathrm{s})} D(\mathrm{e}) \\ Ext(\mathrm{y}) &= \mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{m} \Leftrightarrow \mathrm{e}\mathrm{H}^{\mathrm{T}} = \mathrm{s} = \mathrm{m} - \mathrm{x}\mathrm{H}^{\mathrm{T}} \end{aligned} \tag{4.20}$$

Notre fonction objectif $f(\mathrm{e}) = \langle \gamma, \mathrm{e} \rangle$ (produit scalaire entre le vecteur coût de la programmation linéaire et la variable e) apparait clairement dans l'expression de $D(\mathrm{e})$. Par conséquent les éléments du vecteur coût sont représentés par les coûts de modifications des pixels. Posons $\gamma = \rho = \{\rho_i\}_{1 \leq i \leq n}$ et nous retrouvons la même forme de programmation linéaire. Et la résolution peut se faire de la même manière que dans le cas du profil constant.

4.2.6 Résultats des tests sur des images numériques

Pour vérifier l'efficacité de notre schéma et l'invisibilité des messages cachés en utilisant ce schéma, nous l'avons testé sur différentes images en niveaux de gris. Les images, tirés de la base d'images du $BOSS$[5], sont de format pgm et de taille $(\mathbf{512 \times 512})$ $\boldsymbol{pixels}$.

Nous insérons un message de taille $\mathbf{3}$ $\boldsymbol{ko}$, soit $\mathbf{4576}$ $\boldsymbol{bits}$, dans deux images en niveaux de gris (10.pgm et 1000.pgm) tirées de la base d'images BOSS. Les caractères du message sont convertis en leur code ASCII puis en binaire. Le vecteur binaire obtenu est ensuite subdivisé en sous vecteurs de taille égale à $n - k$ (n et k représentent respectivement la longueur et la dimension du code polaire utilisé). Pour ce qui concerne les images, nous récupérons la matrice des LSBs que nous subdivisons en sous vecteurs de taille n dans lesquels nous insérons les segments de messages.

[5] $BOSS$ (Break Our Stego-System) : compétition pour les attaques des schémas de stéganographie, http://bows2.gipsa-lab.inpg.fr/BOWS20rigEp3.tgz.

Les données nécessaires pour l'extraction du message sont insérées dans les premiers pixels de l'image de couverture. Nous réservons le premier octet de la matrice des LSBs de l'image à la taille du message inséré. Le deuxième et le troisième octet contiennent respectivement la longueur n et la dimension k du code polaire utilisé. Ce choix des pixels qui contiennent ces informations est un exemple parmi tant d'autres et constitue une clé partagée entre les deux entités qui communiquent.

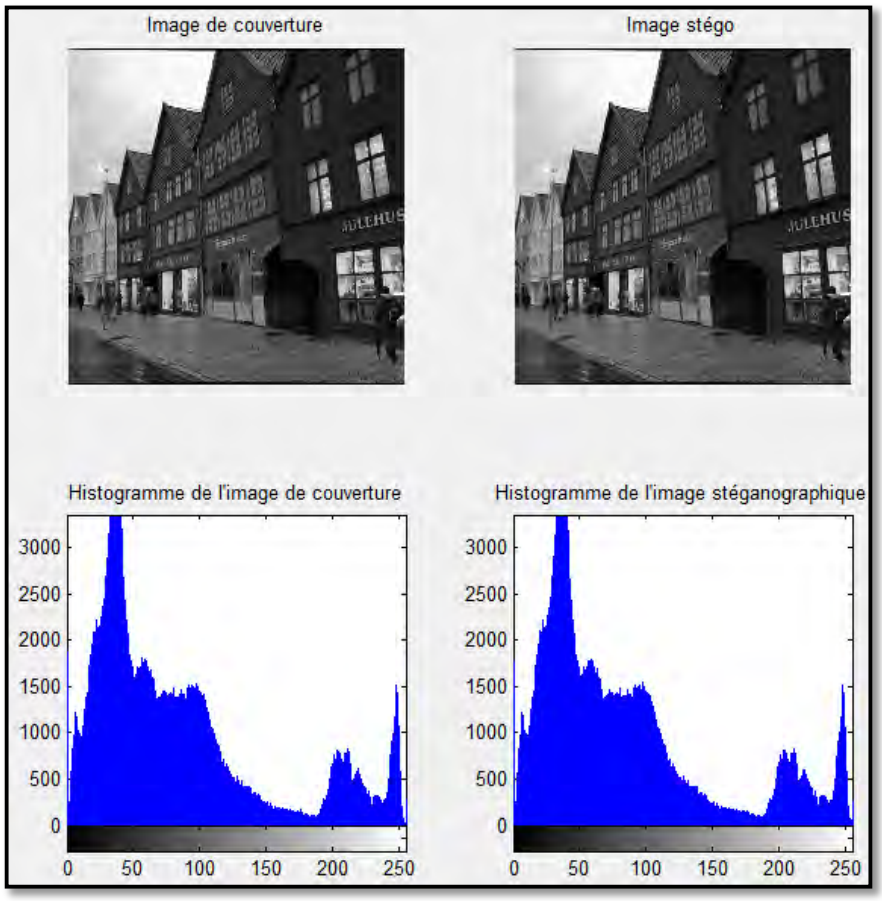

Figure 4.3 : Images de couverture 10.pgm et 10_stégo.pgm et leur histogramme.

Une fois le message inséré dans les images, nous pouvons procéder à l'évaluation de notre schéma :

✓ La première et la plus simple évaluation à faire concerne l'imperceptibilité visuelle. Aussi bien pour l'image 10.pgm que pour le 1000.pgm, les changements dans les images stégos 10_stégo.pgm et 1000_stégo.pgm sont invisibles pour l'œil humain, comme le montrent les Figures 4.3 et 4.4. D'où le premier et principal objectif de la stéganographie est atteint.

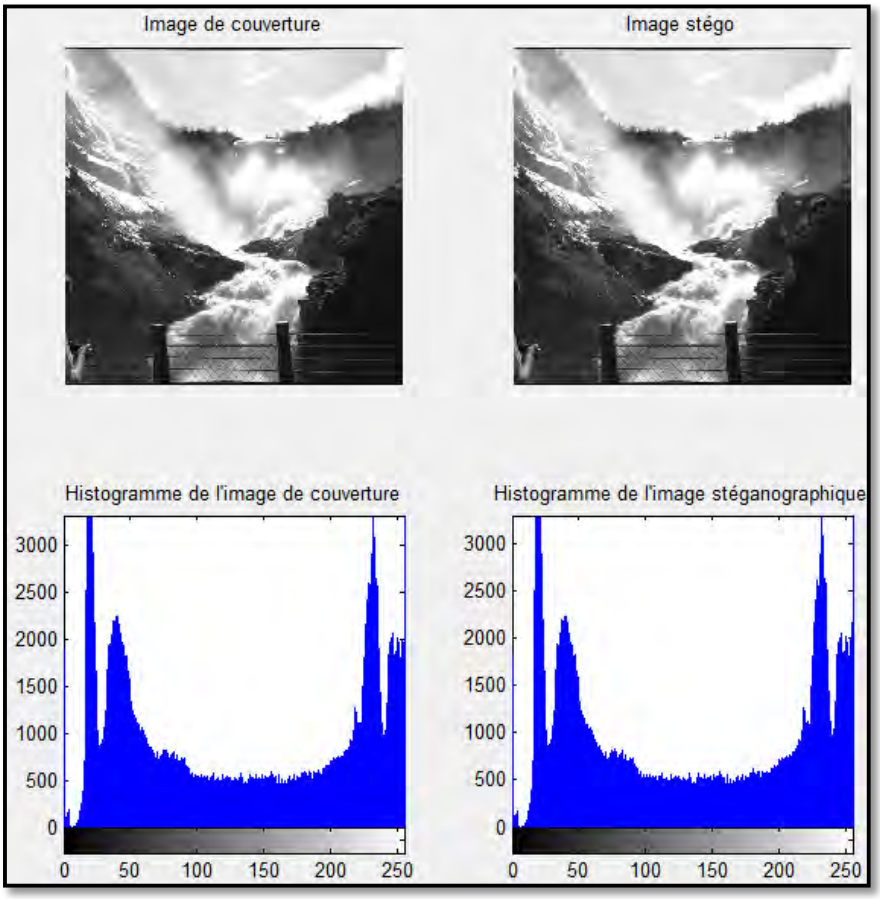

Figure 4.4 : Images de couverture 1000.pgm et 1000_stégo.pgm et leur histogramme.

✓ En comparant les histogrammes des images de couverture et stégo (pour un attaquant "semi-actif"), il est très difficile de voir la différence entre les deux images. Car, il existe une très faible différence entre les histogrammes des images hôte et stégo, qui est très difficile à percevoir. Cette différence entre les deux histogrammes est plus perceptible si la taille du message à insérer augmente.

Notons aussi que nous pouvons prendre des précautions de telle sorte à n'utiliser l'image de couverture une seule fois et la supprimer tout juste après insertion du message secret. Ce qui permettra de garantir que l'attaquant ne dispose que de l'image stégo pour vérifier si elle contient ou non un message secret. Il faudrait donc utiliser des moyens beaucoup plus sophistiqués pour arriver à détecter la présence du message. Par exemple, la proposition d'une méthode de stéganalyse ciblée pour pouvoir attaquer ce schéma. Comme tout bon schéma de stéganographie, le nôtre permet, au destinataire, d'extraire le message secret intégralement et sans aucune altération.

Pour faire la vérification, nous extrairons le message que nous avions inséré dans les images des exemples précédents. Tous d'abord, nous récupérons la longueur n et la dimension k du code ; ce qui nous permet de construire la matrice de contrôle de parité. Cette matrice de contrôle de parité du code polaire est utilisée pour le calcul des syndromes (avec la relation $yH^T = m$) qui correspondent aux segments de messages insérés. La récupération de la taille du message nous donne le nombre d'octets (nombre de pixels) concernés par l'insertion. Le regroupement des segments de messages (par concaténation) permet de reconstruire le message en entier.

Nous pouvons faire la comparaison entre le message inséré et celui extrait de l'image de couverture.

```
La stéganographie est ancien mais n'a connu un véritable développement
qu'à partir de l'année 1998 avec la technique de matrix embedding
introduite par Crandall dans [ME]. Cette technique a permis d'améliorer
l'efficacité d'insertion des schémas de stéganographie. Vue les similarités
qui existent entre le matrix embedding et le principe de décodage des
codes correcteurs d'erreurs, ces derniers sont exploités à des fins
stéganographiques. La première implémentation fut proposée par Westfeld
dans l'algorithme F5[F5] avec un code de Hamming C(n,k,1) et permet de
réduire les modifications des coefficients DCT quantifiés. Les codes BCH
structurés ont été introduits en stéganographie par Schönfeld et Winkler
[BCH]. Les auteurs ont proposé deux manières de calcul de syndrome : une
approche classique pour trouver le vecteur d'erreur de poids minimal en
utilisant une matrice de contrôle de parité H et une autre moins complexe
utilisant un polynôme générateur g(x) pour la recherche des racines du
polynôme localisateur d'erreurs. Une amélioration est fournie par
R. Zhang, V. Sachnev, H. Joong Kim dans [FastBCH] avec l'utilisation des
tables de correspondance en vue de réduire la complexité. L'efficacité
d'insertion est supérieure à celle des codes de Hamming mais reste
toujours faible. L'utilisation des codes RS dans la stéganographie est
rendue possible grâce au travail de Galand et Fontaine [RS]  dans lequel
la version des codes GRS (Reed-Solomon Généralisé) est utilisée pour
définir un schéma de stéganographie à papier mouillé. Plus récemment, les
codes convolutifs, plus particulièrement les codes STC [STC] ont été
exploités pour faire de la stéganographie. Leurs performances sont avérées
être meilleures. Friedrich et al. ont utilisé ces codes dans leur schéma
HUGO [HUGO]. La combinaison des techniques de LSB, matrice embedding et
du papier mouillé a permis de réaliser des schémas stéganographiques plus
efficaces. Friedrich et al. ont précisé dans [HUGO] que pour produire de
meilleurs schémas stéganographiques il est possible d'utiliser soit un bon
modèle soit un bon code. Nous portons notre choix sur les codes polaires
qui ont été récemment découverts par Arikan [PC] et dont les performances
sont prouvées.
```

Figure 4.5 : Message inséré.

Le message inséré dans l'image de couverture (Figure 4.5) est identique à celui extrait à partir de l'image stégo (Figure 4.6). D'où, l'objectif de récupération du message en entier, est atteint.

```
La stéganographie est ancien mais n'a connu un véritable développement
qu'à partir de l'année 1998 avec la technique de matrix embedding
introduite par Crandall dans [ME]. Cette technique a permis d'améliorer
l'efficacité d'insertion des schémas de stéganographie. Vue les similarités
qui existent entre le matrix embedding et le principe de décodage des
codes correcteurs d'erreurs, ces derniers sont exploités à des fins
stéganographiques. La première implémentation fut proposée par Westfeld
dans l'algorithme F5[F5] avec un code de Hamming C(n,k,1) et permet de
réduire les modifications des coefficients DCT quantifiés. Les codes BCH
structurés ont été introduits en stéganographie par Schönfeld et Winkler
[BCH]. Les auteurs ont proposé deux manières de calcul de syndrome : une
approche classique pour trouver le vecteur d'erreur de poids minimal en
utilisant une matrice de contrôle de parité H et une autre moins complexe
utilisant un polynôme générateur g(x) pour la recherche des racines du
polynôme localisateur d'erreurs. Une amélioration est fournie par
R. Zhang, V. Sachnev, H. Joong Kim dans [FastBCH] avec l'utilisation des
tables de correspondance en vue de réduire la complexité. L'efficacité
d'insertion est supérieure à celle des codes de Hamming mais reste
toujours faible. L'utilisation des codes RS dans la stéganographie est
rendue possible grâce au travail de Galand et Fontaine [RS]  dans lequel
la version des codes GRS (Reed-Solomon Généralisé) est utilisée pour
définir un schéma de stéganographie à papier mouillé. Plus récemment, les
codes convolutifs, plus particulièrement les codes STC [STC] ont été
exploités pour faire de la stéganographie. Leurs performances sont avérées
être meilleures. Friedrich et al. ont utilisé ces codes dans leur schéma
HUGO [HUGO]. La combinaison des techniques de LSB, matrice embedding et
du papier mouillé a permis de réaliser des schémas stéganographiques plus
efficaces. Friedrich et al. ont précisé dans [HUGO] que pour produire de
meilleurs schémas stéganographiques il est possible d'utiliser soit un bon
modèle soit un bon code. Nous portons notre choix sur les codes polaires
qui ont été récemment découverts par Arikan [PC] et dont les performances
sont prouvées.
```

Figure 4.6 : Message extrait.

Pour évaluer les performances de notre schéma, nous avons également calculé le MSE et le PNSR avec les relations (2.1) et (2.2). Nous générons de façon aléatoire 10 messages de tailles différentes que nous insérons dans 5 images (1.pgm, 10.pgm, 100.pgm, 1000.pgm et 10000.pgm). Nous avons fait la moyenne des MSE et PNSR et les résultats sont représentés sur la Figure 4.7.

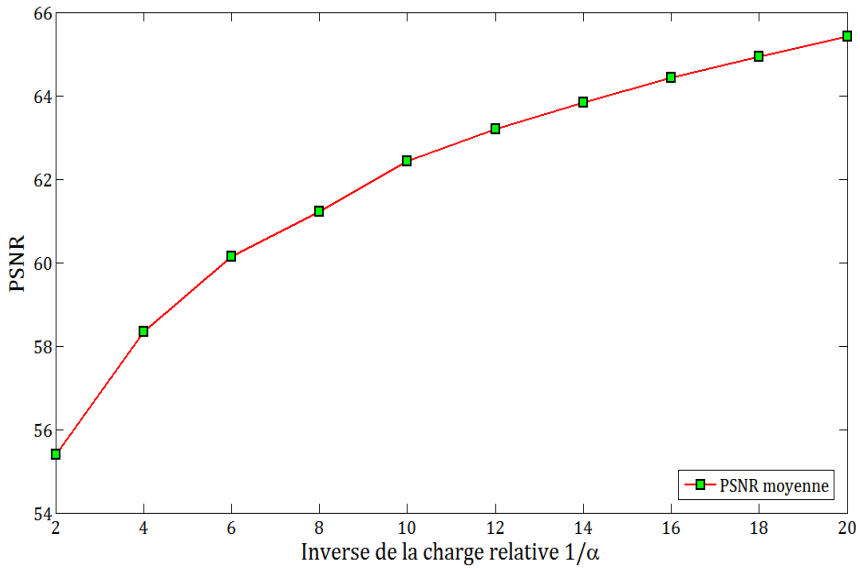

Figure 4.7 : Variation du PSNR en fonction de la charge relative.

Ces valeurs sont bien supérieures à $\mathbf{35}\ \boldsymbol{dB}$, valeur au-delà de laquelle la différence entre les deux images (couverture et stégo dans notre cas) est très faible [40]. Cela montre que le schéma proposé présente une bonne performance en termes d'efficacité d'insertion.

4.3 Nouveaux algorithmes de stéganographie par codage polaire

Nous proposons deux méthodes l'une basée sur les tables de correspondance [45] et l'autre exploite la forme du syndrome pour évaluer la position des modifications [46].

4.3.1 Méthode basée sur les tables de correspondance

Rappelons qu'une table de correspondance est une structure de données utilisée pour remplacer un calcul par une opération de consultation afin de gagner en vitesse, car rechercher une valeur en mémoire est souvent plus rapide qu'effectuer un calcul important. La méthode proposée dans cette sous-section utilise ces tables de correspondance pour donner le vecteur de modifications à partir du syndrome. Celui-ci est donné par $\mathrm{s} = \mathrm{eH}^{\mathrm{T}} = \mathrm{m} - \mathrm{xH}^{\mathrm{T}}$. Les cas de profil constant et du papier mouillé sont considérés. La structure particulière de la matrice de contrôle de parité des codes polaires va être exploitée pour faciliter les calculs.

4.3.1.1 Cas du profil constant

La table de correspondance est un tableau composé de 2 colonnes de 2^{n-k} lignes chacune. La première colonne contient les différentes configurations de syndrome possibles et la seconde représente les leaders de cosets (ou vecteurs de modifications). Ainsi, sur une ligne i, avec $1 \leq i \leq 2^{n-k}$, nous avons le syndrome $\mathrm{s} = \mathrm{S}[i]$ sur la première colonne et le vecteur de modifications $\mathrm{e} = \mathrm{E}[i]$ sur la deuxième colonne. Après avoir calculé un syndrome s alors pour trouver e on procède comme suit [45] :

- on parcourt les lignes de la colonne 1;
- si on trouve la position i à laquelle on a $\mathrm{S}[i] = \mathrm{s}$ alors
- le coset correspondant au syndrome s se trouve à la même position i dans la colonne 2.

L'opération de calcul de leaders de coset est ainsi remplacée par une consultation dans la table stockée. En guise d'exemples, nous utilisons un code polaire $PC(4,1)$ pour la stéganographie $\mathcal{S}_{PC}(4,3)$ et $PC(8,4)$ pour $\mathcal{S}_{PC}(8,4)$.

Exemple 4.7: Une matrice de contrôle de parité de $PC(4,1)$

$$\mathrm{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ et } \mathrm{H}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.21}$$

Les colonnes $H_{.j}$ de H vérifient les égalités[6] suivantes :

$$\begin{aligned} &\blacksquare \quad H_{.1} + H_{.2} = H_{.3} + H_{.4} = (0\ 0\ 1)^{\mathrm{T}} \\ &\blacksquare \quad H_{.1} + H_{.3} = H_{.2} + H_{.4} = (0\ 1\ 0)^{\mathrm{T}} \\ &\blacksquare \quad H_{.1} + H_{.4} = H_{.2} + H_{.3} = (0\ 1\ 1)^{\mathrm{T}} \end{aligned} \tag{4.22}$$

Le syndrome peut avoir une des trois configurations suivantes [45]:

- **Config 1** : le syndrome est égal à une colonne de H;
- **Config 2** : il est égal au vecteur nul;
- **Config 3** : c'est la somme de 2 colonnes de H avec le système (4.22).

Ce qui nous permet de dresser le tableau suivant :

Tableau 4-1: Table de correspondance pour $\mathcal{S}_{PC}(4,3)$.

colonne 1: syndromes			colonne 2: modifications	
S [1]	1 0 0		E [1]	1 0 0 0
S [2]	1 0 1		E [2]	0 1 0 0
S [3]	1 1 0		E [3]	0 0 1 0
S [4]	1 1 1	⇔	E [4]	0 0 0 1
S [5]	0 0 0		E [5]	0 0 0 0
S [6]	0 0 1		E [6]	1 1 0 0
S [7]	0 1 0		E [7]	1 0 1 0
S [8]	0 1 1		E [8]	1 0 0 1

Considérons un message $\mathrm{m} = (0\ 1\ 0)$ et un vecteur de couverture $\mathrm{x} = (1\ 0\ 0\ 1)$. Le calcul du syndrome donne $s = \mathrm{eH}^{\mathrm{T}} = \mathrm{m} - \mathrm{xH}^{\mathrm{T}} = (0\ 0\ 1) = \mathrm{S}[6]$. Donc le leader de coset correspondant est $\mathrm{E}[6] = (1\ 1\ 0\ 0) = \mathrm{e}$.

[6] Ici, on effectue une addition binaire des colonnes.

Exemple 4.8: Une matrice de contrôle de parité de $PC(8,4)$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{et} \quad H^{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.23}$$

Comme à l'exemple précédent, nous avons le système :

- $H_{.1} + H_{.2} = H_{.3} + H_{.4} = H_{.5} + H_{.6} = H_{.7} + H_{.8} = (0\ 0\ 0\ 1)^{T}$
- $H_{.1} + H_{.3} = H_{.2} + H_{.4} = H_{.5} + H_{.7} = H_{.6} + H_{.8} = (0\ 0\ 1\ 0)^{T}$
- $H_{.1} + H_{.4} = H_{.2} + H_{.3} = H_{.5} + H_{.8} = H_{.6} + H_{.7} = (0\ 0\ 1\ 1)^{T}$
- $H_{.1} + H_{.5} = H_{.2} + H_{.6} = H_{.3} + H_{.7} = H_{.4} + H_{.8} = (0\ 1\ 0\ 0)^{T}$ (4.24)
- $H_{.1} + H_{.6} = H_{.2} + H_{.5} = H_{.3} + H_{.8} = H_{.4} + H_{.7} = (0\ 1\ 0\ 1)^{T}$
- $H_{.1} + H_{.7} = H_{.2} + H_{.8} = H_{.3} + H_{.5} = H_{.4} + H_{.6} = (0\ 1\ 1\ 0)^{T}$
- $H_{.1} + H_{.8} = H_{.2} + H_{.7} = H_{.3} + H_{.6} = H_{.4} + H_{.5} = (0\ 1\ 1\ 1)^{T}$

Nous avons **Config 1, 2, 3** de l'exemple précédant et le **Tableau 4.2**.

Tableau 4-2: Table de correspondance pour $\mathcal{S}_{PC}(8,4)$.

colonne 1: syndromes			colonne 2: modifications	
S [1]	1 0 0 0		E [1]	1 0 0 0 0 0 0 0
S [2]	1 0 0 1		E [2]	0 1 0 0 0 0 0 0
S [3]	1 0 1 0		E [3]	0 0 1 0 0 0 0 0
S [4]	1 0 1 1		E [4]	0 0 0 1 0 0 0 0
S [5]	1 1 0 0		E [5]	0 0 0 0 1 0 0 0
S [6]	1 1 0 1		E [6]	0 0 0 0 0 1 0 0
S [7]	1 1 1 0		E [7]	0 0 0 0 0 0 1 0
S [8]	1 1 1 1	⇔	E [8]	0 0 0 0 0 0 0 1
S [9]	0 0 0 0		E [9]	0 0 0 0 0 0 0 0
S [10]	0 0 0 1		E [10]	1 1 0 0 0 0 0 0
S [11]	0 0 1 0		E [11]	1 0 1 0 0 0 0 0
S [12]	0 0 1 1		E [12]	1 0 0 1 0 0 0 0
S [13]	0 1 0 0		E [13]	1 0 0 0 1 0 0 0
S [14]	0 1 0 1		E [14]	1 0 0 0 0 1 0 0
S [15]	0 1 1 0		E [15]	1 0 0 0 0 0 1 0
S [16]	0 1 1 1		E [16]	1 0 0 0 0 0 0 1

Soient le vecteur de couverture $x = (1\ 0\ 1\ 0\ 1\ 0\ 0\ 1)$ et le message $m = (1\ 1\ 0\ 1)$. Le syndrome est $s = (1\ 1\ 0\ 0) = S[5]$ et le leader de coset est $E[6] = (0\ 0\ 0\ 0\ 1\ 0\ 0\ 0) = e$.

4.3.1.2 Cas du papier mouillé

Soit $\mathcal{J}$ l'ensemble des positions à verrouiller (éléments mouillés). Si on trouve un vecteur de modifications e avec un bit égal à 1 (un 1-bit) à une position pos, devant être verrouillée, alors une des trois configurations se présente [45].

- Le syndrome calculé s se trouve dans la **Config 1** (c'est-à-dire le vecteur e a un seul 1-bit à une position à verrouiller) et on procède comme suit :

On subdivise la première moitié de la table de correspondance composée de n éléments en $n/4$ sous-ensembles de 4 éléments chacun. Dans chaque sous-ensemble, un syndrome est égal à la somme des 3 autres. Par conséquent, pour verrouiller la position pos, on considère le vecteur de modifications (dans le sous-ensemble du syndrome calculé) correspondant à la somme des 3 vecteurs dont la somme de leurs syndromes est égale au syndrome du vecteur de modifications e. Ce qui nous donne un autre vecteur ayant 3 positions i, j et l égales à 1 et différentes de pos. Si une de ces 3 positions est dans $\mathcal{J}$ alors on cherche un autre vecteur avec des 1-bits à des positions non-verrouillées. Pour ce faire, on remplace le couple du triplet (i, j, l) appartenant à $\mathcal{J}$ par un autre qui ne soit pas dans $\mathcal{J}$ avec les égalités d'un équivalent[7] du système (4.24).

- Le syndrome est dans la **Config 2**, alors :

Aucun problème ne se pose puisque le vecteur des modifications a tous ses composants nuls.

- Le syndrome est dans la **Config 3** (c'est-à-dire e a deux 1-bits aux positions 1 et pos dont au moins l'une doit être verrouillée), alors :

On cherche le couple (i, j) n'appartenant pas à $\mathcal{J}$ avec les égalités du système. Le vecteur de modifications aura ainsi deux 1-bits aux positions i et j.

[7] Equivalent désigne le système obtenu dans le cas d'une stéganographie avec un paramètre différent de 8. Par exemple, pour la stéganographie $\mathcal{S}_{PC}(4,3)$, le système est (4.22).

Exemple 4.9 :

Considérons toujours le vecteur de couverture $x = (1\ 0\ 1\ 0\ 1\ 0\ 0\ 1)$ et le message $m = (1\ 1\ 0\ 1)$. Le syndrome est $s = (1\ 1\ 0\ 0) = S[5]$ et le premier vecteur de modifications trouvé est $E[6] = (0\ 0\ 0\ 0\ 1\ 0\ 0\ 0) = e$. Si $\mathcal{J} = (5,6,7)$ alors le seul 1-bit est à la 5-ème position qui doit être verrouillée. Ainsi, on subdivise la première moitié du **Tableau 4.2** en 2 sous-ensembles de 4 éléments. Le syndrome s se trouve dans le deuxième sous-ensemble. Ainsi, $s = (1\ 1\ 0\ 0) = S[5] = S[6] + S[7] + S[8]$ et $e = E[5] = E[6] + E[7] + E[8] = (0\ 0\ 0\ 0\ 0\ 1\ 1\ 1)$. Ce vecteur e a trois 1-bits aux positions 6, 7 et 8. Le couple $(6,7)$ étant dans $\mathcal{J}$; nous devons donc le remplacer par autre couple avec les égalités du système 4.24. D'après la troisième ligne, nous pouvons remplacer le couple $(6,7)$ soit par $(1,4)$ ou par $(2,3)$. Nous pouvons donc choisir entre $e = (1\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$ ou $e = (0\ 1\ 1\ 0\ 0\ 0\ 0\ 1)$ pour le vecteur de modifications.

Cet exemple que nous venons de donner est le pire cas de figure qui puisse se présenter puisque les deux autres configurations sont plus faciles à résoudre.

4.3.2 Méthode de calcul direct du vecteur de modifications

Cette méthode exploite la forme du syndrome calculé à partir du médium de couverture et le message secret. Nous proposons deux approches pour cette méthode de stéganographie. La première approche est définie en se basant sur les tables de correspondance [45]. Il faut noter, cependant, que le stockage de ces tables ne sera pas nécessaire. La deuxième approche [46] sera une version améliorée de la première. Sa particularité est qu'elle est définie sans les tables de correspondance. Nous allons considérer le cas de profil constant et du papier mouillé comme nous l'avions fait avec la méthode précédente.

4.3.2.1 Première approche de la méthode de calcul direct

- **Profil constant**

La méthode que nous allons proposer exploite une correspondance uniforme entre la valeur du syndrome et la position des coefficients non-nuls du vecteur de modifications correspondant. Par exemple, sur le **Tableau 4.2**, les observations suivantes peuvent être faites :

- **Cas 1** : sur la première moitié du tableau (de l'élément 1 à l'élément $8 = n$), les syndromes ont leur premier bit (le bit de poids fort) égal à 1 et leur valeur décimale

varie entre $8 = n$ et $15 = 2n - 1$. La position du seul 1-bit du vecteur de modifications va de 1 à $8 = n$ (on compte de la gauche vers la droite) ;

- **Cas 2** : à la $9 = (n + 1)$-ème ligne, le syndrome et le vecteur de modifications sont tous des vecteurs nuls;
- **Cas 3** : sur la partie restante du tableau (du $10 = (n + 2)$-ème au $16 = 2n$-ème et dernier élément), les syndromes ont comme premier bit 0 et leur valeur décimale varie entre 1 et $7 = n - 1$. Les vecteurs de modifications ont deux 1-bits le premier à la position 1 et le second à une position qui va de 2 à $8 = n$.

Les trois parties du **Tableau 4.2** sont séparées par des traits épais. Ces parties et les remarques associées sont aussi valables pour le **Tableau 4.1 (**avec $n = 4$) et pour les autres tableaux correspondants aux systèmes équivalents. D'après ces remarques, il existe une relation entre la valeur décimale du syndrome s et la position des éléments non-nuls du vecteur de modifications e. Cela est dû au fait que, sur les colonnes de la matrice de contrôle de parité H, figurent les valeurs binaires des nombres de n à $2n - 1$. Cependant, une condition nécessaire est que le nombre de syndromes doit être égal au double de la taille du vecteur de couverture.

$$2^{n-k} = 2n \tag{4.25}$$

Or n est une puissance de 2, soit $n = 2^p$, alors

$$2^{n-k} = 2 \cdot 2^p = 2^{p+1}. \tag{4.26}$$

Donc

$$n - k = p + 1. \tag{4.27}$$

Ainsi

$$k = n - 1 - p = n - 1 - \log_2 n = 2^p - 1 - p \tag{4.28}$$

Les codes polaires $PC(4,1)$ et $PC(8,4)$, donnés en exemples avec la méthode des tables de correspondance sont tels que (4.28) est vérifiée. En effet, pour $PC(4,1)$ on a $k = 1 = 4 - 1 - \log_2 4 = 2^2 - 1 - 2$ et $k = 4 = 8 - 1 - \log_2 8 = 2^3 - 1 - 3$ pour $PC(8,4)$.

La validité des observations concernent les valeurs de [75]

$$\begin{matrix} p \in \{2,3,4,5,6,7\} = \mathcal{P} \\ n \in \{4,8,\dots,128\} = \mathcal{N} \\ k \in \{1,4,\dots,120\} = \mathcal{K} \end{matrix} \quad (4.29)$$

avec $n = 2^p$, $k = 2^p - 1 - p$ et $p \in \mathcal{P}$.

Pour un code polaire $PC(n,k)$, la longueur n est une puissance de 2 et la dimension k est un entier positif dans $\{1,2,\cdots,n-1\}$. Concernant un schéma de stéganographie par codage polaire, la relation d'optimalité [75] est $m = n - k > p = \log_2 n$. Les paramètres de notre code polaire dans l'approche proposée vérifient cette condition car on a $n - k = p + 1 > p$. Soit un code polaire $PC(N = 2^P, K)$ pour la stéganographie $\mathcal{S}_{PC}(N, N-K)$. Si $N \notin \mathcal{N}$ ($P \in \{8,9,\dots\} = \mathbb{N}\backslash(\mathcal{P} \cup \{0,1\})$ et/ou $K \notin \mathcal{K}$, on peut toujours se ramener à un cas de validité.

- Pour $N \in \mathcal{N}$, si $K \in \mathcal{K}$ alors on applique directement la méthode avec $\mathcal{S}_{PC}(N, N-K)$ sinon on normalise K.
- Pour $N \notin \mathcal{N}$, on normalise N puis K.

Normalisation N :

On subdivise N en plusieurs parties n tels $n \in \mathcal{N}$. Puisque N et n sont des puissances de 2 et $N > n$, N est divisible par tout $n \in \mathcal{N}$. Le rapport $N/n = 2^P/2^p = 2^{P-p}$ est aussi une puissance de 2. On obtient 2^{P-p} segments de taille n.

Normalisation de K :

On doit ramener K à un entier $k \in \mathcal{K}$. Mais, puisque c'est la taille du message qui nous intéresse en stéganographie, nous allons donc chercher à subdiviser $N - K$ en $n - k = p + 1$ parties telles que $n = 2^p \in \mathcal{N}$ et $k = (n - 1 - \log_2 n) \in \mathcal{K}$. Puisque nous connaissons n, nous pouvons déterminer k. $N - K$ n'est pas toujours divisible par $n - k$. Soit $N - K = (n-k) \cdot q + r$, avec $0 \leq r < n - k$. Si $r = 0$ alors on subdivise $N - K$ en q segments de taille $n - k$. Dans le cas contraire ($0 < r < n - k$), on a q segments de taille $n - k$ et un autre restant de taille r. On complète le segment restant avec des 0 pour avoir une taille égale à $n - k$. Ainsi on aura finalement $q + 1$ segments de taille $n - k$.

L'insertion se fait par couple d'un segment du médium de couverture et d'un segment de message avec le code polaire $PC(n,k)$ répondant au critère de validité. Pour cela le nombre de

segments de couverture doit être supérieur ou égal au nombre de segments de message. Nous proposons l'algorithme suivant qui calcule un leader de coset pour un syndrome donné :

Algorithme 4.1 Calcul d'un leader de coset d'un syndrome

Entrées: le vecteur de couverture **x**, le message secret **m** et la matrice de contrôle de parité **H**.

Sorties: le leader de coset **e**.

1: Initialisation : $p \leftarrow$ *un élément de* $\mathcal{P}$; $n \leftarrow 2^p$; $k \leftarrow n - 1 - p$;

2: $\mathbf{e} \leftarrow (0, \dots, 0)$; $\mathbf{y} \leftarrow \mathbf{x}$;

3: **Si** ($\mathbf{xH^T} \neq \mathbf{m}$) **alors**

4: $\mathbf{s} \leftarrow \mathbf{m} - \mathbf{xH^T}$;

5: calculer la valeur décimale du syndrome binaire ($dec \leftarrow ConversionDecimale$ (**s**))

6: **si** le 1^{er} coefficient du syndrome **s** est égal à 1 **alors**

7: on met le $(dec + 1 - n)$-ème coefficient du vecteur **e** à 1 ($\mathbf{e}[dec + 1 - n] \leftarrow 1$);

8: **sinon**

9: on met le 1^{er} et le $(dec + 1)$-ème coefficient de **e** à 1 ($\mathbf{e}[1] \leftarrow 1$ et $\mathbf{e}[dec + 1] \leftarrow 1$);

10: **fin (si)**

11: **Fin (Si)**

La fonction $conversionDecimale(\mathrm{s})$ permet de faire la conversion d'un vecteur binaire s à sa valeur décimale.

O Papier mouillé

Dans cette section, nous allons expliquer comment les codes polaires peuvent être utilisés pour un canal à papier mouillé. Donnons dans un premier temps deux théorèmes qui conditionnent l'applicabilité des codes polaires dans le cas du papier mouillé.

Théorème 4.1 (Rang de la matrice de contrôle de parité) [76]: Si une matrice de contrôle de parité H d'un code polaire de longueur de bloc n et de dimension k est obtenue en utilisant le lemme de Goela et al. [29, lemme 1] alors son rang est :

$$rang(\mathrm{H}) = n - k \tag{4.30}$$

Preuve : La matrice génératrice d'un code polaire G_n est inversible [25] c'est-à-dire ses colonnes sont linéairement indépendantes (aucune colonne n'est combinaison linéaire des

autres). Cela est équivalent à $rang(\mathrm{G}_n) = n$. La matrice H^{T} étant obtenue en retranchant à G_n les k colonnes d'indices dans l'ensemble des bits d'information A [29]. Donc G_n est la matrice H^{T} à laquelle on ajoute k autres colonnes dont aucune n'est combinaison linéaire des autres colonnes de H^{T}. Ainsi $rang(\mathrm{H}^{\mathrm{T}}) + k = rang(\mathrm{G}_n) = n$. Or $rang(\mathrm{H}^{\mathrm{T}}) = rang(\mathrm{H})$. Donc $rang(\mathrm{H}) + k = n$. Par suite $rang(\mathrm{H}) = n - k$.

Considérons toujours l'ensemble des éléments mouillés $\mathcal{J}$. Le nombre maximum de positions qu'on peut verrouiller pour la stéganographie à papier mouillé est $n - rang(\mathrm{H}) = k$.

Théorème 4.2 (Nombre maximum de positions qu'on peut verrouiller) [76] : Soit $\mathcal{S}_{PC}(n, m = n - k)$ la stéganographie appliquée à un code polaire tel que $n \in \mathcal{N}$ et $k \in \mathcal{K}$. Le nombre maximal ℓ_{max} pour lequel on est toujours capable de verrouiller n'importe quelle combinaison de ℓ_{max} positions est :

$$\ell_{max} = \frac{n}{2} - 1 \tag{4.31}$$

Preuve : Considérons, par exemple, le verrouillage des $n/2$ dernières positions du vecteur de couverture. Cela revient à élaguer les $n/2$ dernières colonnes de la matrice H pour le produit matriciel $\mathrm{yH}^{\mathrm{T}} = \mathrm{m}$. Dans ce cas, la deuxième ligne de la matrice H a tous ses éléments égaux à 0 et peut donc s'écrire comme combinaison des autres lignes (voir par exemple (4.21) et (4.23)). Ce qui signifie qu'on n'est pas toujours capable de verrouiller $n/2$ ou plus. Le nombre maximal de positions qu'on peut toujours verrouiller, pour toute combinaison, est donc inférieur à $n/2$. Il est compris entre 1 et $n/2 - 1$. Soit ℓ le nombre de positions verrouillées. Donc on a $1 \leq \ell \leq n/2 - 1$. Dans notre problème de stéganographie, nous devons verrouiller un nombre de positions conduisant à un système ayant au moins une solution. Nous devons donc avoir un système ayant un nombre d'inconnues supérieur ou égal au nombre d'équations. Le nombre d'inconnus du système après verrouillage est égal à $n - \ell$ et le nombre d'équations est égal à $n - k = 1 + \log_2 n = 1 + p$ (le rang de H). Donc, nous devons avoir $n - \ell \geq n - k$, d'où $\ell \leq k$. Par conséquent, deux contraintes s'impose à la valeur de ℓ ($\ell \leq n/2 - 1$ et $\ell \leq k$). Par suite $\ell_{max} = min\,\{k, n/2 - 1\} = min\,\{n - 1 - \log_2 n\,, n/2 - 1\} = min\,\{2^p - 1 - p, 2^{p-1} - 1\}$.
Montrons que $\ell_{max} = n/2 - 1$. Cela revient à démontrer que $n/2 - 1 \leq n - 1 - \log_2 n$. Ce qui équivaut à démontrer que la différence $diff = n - 1 - \log_2 n - (n/2 - 1) = n/2 - \log_2 n = 2^{p-1} - p$ est positive. Considérons, pour cela, la fonction $f: \mathbb{N}\backslash\{0,1\} \leftarrow \mathbb{Z}$ telle que $f(p) = 2^{p-1} - p$. Cette fonction est continue et dérivable. Sa dérivée est $f'(p) = (p - 1) \cdot$

$2^{p-2} - 1 \geq 0$, pour $p \geq 2$. Ce qui implique que f est croissante. En plus, $f(2) = 2^{2-1} - 2 = 0$. Donc $f(p) \geq 0$, pour tout $p \geq 2$. Ce qui implique que $diff \geq 0$. Ainsi $n - 1 - \log_2 n \geq n/2 - 1$. Finalement, on a $\ell_{max} = n/2 - 1$.

Ainsi, nous pouvons, d'une part, verrouiller k positions mais pas n'importe lesquelles. Et, d'autre part, toute combinaison de $n/2 - 1$ positions peut être choisie pour le verrouillage. Donc pour s'assurer de pouvoir toujours réussir le verrouillage, nous n'allons pas dépasser la valeur ℓ_{max}. On considère toujours l'ensemble des positions à verrouiller $\mathcal{J}$. Soit un vecteur de modifications e, on distingue trois cas :

- Le syndrome calculé s se trouve dans le **Cas 1** (e a un seul 1-bit à la position pos à verrouiller) alors :

On considère, par quadruplet, les valeurs décimales des syndromes correspondant à des vecteurs de modifications avec une seule position à 1-bit. Les syndromes considérés (au nombre de n) représentent la moitié du nombre de configurations de syndromes possibles (qui est de $2n$) et le nombre de quadruplets est de $n/4$. Les valeurs décimales vont de n à $2n - 1$ (voir par exemple **Tableau 4.2** avec $n = 8$). Les valeurs décimales des syndromes et leurs quadruplets Q_i se présentent comme suit [76] :

$$\begin{array}{rcccl} de & n = n + 0 \cdot 4 & \text{à} & n + 3 = n + 1 \cdot 4 - 1 & Q_1 \\ & n + 1 \cdot 4 & \rightarrow & n + 2 \cdot 4 - 1 & Q_2 \\ & n + 2 \cdot 4 & \rightarrow & n + 3 \cdot 4 - 1 & Q_3 \\ & & \vdots & & \\ & 2n - 4 = n + (n/4 - 1) \cdot 4 & \rightarrow & 2n - 1 = n + (n/4) \cdot 4 - 1 & Q_{n/4} \end{array} \quad (4.32)$$

Une généralisation de cette représentation donne

$$Q_i\colon\ n + (i-1) \cdot 4 \ \rightarrow\ n + (i) \cdot 4 - 1, \ \text{avec } i = 1, \ldots, n/4 \quad (4.33)$$

Pour connaître le quadruplet Q_i auquel appartient un syndrome s, on calcule son indice i par

$$i = \left\lceil \frac{dec(\mathrm{s}) - n + 1}{4} \right\rceil \quad (4.34)$$

avec $\lceil . \rceil$ l'opération d'arrondi supérieur et $dec(\mathrm{s})$ la valeur décimale du syndrome s.

Preuve : D'après (4.33), la valeur décimale d'un syndrome s $dec(\mathrm{s})$ varie entre $n + (i-1) \cdot 4$ et $n + (i) \cdot 4 - 1$. On peut écrire $dec(\mathrm{s})$ varie entre $n + 4 \cdot i - 4$ et $n + 4 \cdot i - 1$ pour Q_i.

Par conséquent $(dec(\mathrm{s}) - n + 1)/4$ varie entre $i - 3/4$ et i. Ainsi, si on arrondit à la borne entière supérieure alors $\lceil (dec(\mathrm{s}) - n + 1)/4 \rceil$ est compris entre $\lceil i - 3/4 \rceil = i$ et $\lceil i \rceil = i$. Ce qui donne (4.34).

Par exemple pour $\mathcal{S}_{PC}(4,3)$, nous avons un seul quadruplet $(4, 5, 6, 7)$ et pour $\mathcal{S}_{PC}(8,4)$, nous avons deux quadruplets $(8, 9, 10, 11)$ et $(12, 13, 14, 15)$. Ce que nous pouvons vérifier avec les **Tableaux 4.1** et **4.2**. Dans chaque quadruplet, un syndrome est égal à la somme bit-à-bit des 3 autres syndromes. Par conséquent, pour verrouiller la position pos, on considère le vecteur de modifications (dans le quadruplet) correspondant à la somme des 3 leaders de cosets dont la somme de leurs syndromes est égale au syndrome de notre vecteur e. Ce qui nous donne un autre vecteur ayant 3 positions (i, j et l) non-nuls différentes de pos. Si une de ces 3 positions est dans $\mathcal{J}$ alors on procède de la même façon qu'avec la méthode du papier mouillé avec les tables de correspondance (sous-section 4.3.1.2.). On cherche ainsi un autre vecteur de modifications avec des 1-bits à des positions non-verrouillées. Si un couple du triplet (i, j, l) est dans $\mathcal{J}$, alors on choisit ce couple sinon, si un seul des trois éléments est dans $\mathcal{J}$, on prend un couple contenant cet élément. Le couple choisi est ensuite remplacé par un autre couple qui n'est pas dans $\mathcal{J}$ avec les égalités d'un équivalent du système (4.24).

- Si le syndrome est dans le **Cas 2**, alors

Aucun problème ne se pose car le vecteur de modifications e a tous ses composants nuls.

- Si le syndrome est dans le **Cas 3** (le vecteur e a deux 1-bits aux positions 1 et pos dont au moins l'une est à verrouiller), alors

On cherche le couple d'indices (i, j) n'appartenant pas à $\mathcal{J}$ avec les égalités du système. Le vecteur de modifications aura ainsi deux 1-bits aux positions i et j.

Pour le remplacement de positions, l'algorithme est le suivant :

Algorithme 4.2. Remplacement d'un couple par un autre non-inclus dans $\mathcal{J}$.

Entrées : couple à remplacer (i, j) et l'ensemble des éléments mouillés $\mathcal{J}$.

Sortie : le nouveau couple obtenu (l, t).

Déroulement :

1: **Tant que** (on n'a pas trouvé et qu'il reste des éléments à parcourir) **on continue**

2: on cherche une première position (de 1 à n) n'appartenant pas à $\boldsymbol{\mathcal{J}}$ (soit l_1).
3: on cherche une seconde position qui n'est pas dans $\boldsymbol{\mathcal{J}}$ (soit t_1, avec $t_1 > l_1$).
4: **Si** (l_1,t_1) vérifie avec ($\boldsymbol{i}, \boldsymbol{j}$) une des relations de (4.24) alors
// i.e. $(bin(n + \boldsymbol{i} - 1) + bin(n + \boldsymbol{j} - 1) = bin(n + l_1 - 1) + bin(n + t_1 - 1))$
5: $\boldsymbol{l} \leftarrow l_1$; $\boldsymbol{t} \leftarrow t_1$; **retourner** $(\boldsymbol{l}, \boldsymbol{t})$.
6: **Sinon**
7: on revient à la ligne 3.
8: **Fin (Si)**
9: **si** on n'a pas un bon couple (l_1,t_1)
10: on revient à la ligne 2.
11: **fin (si)**
12: on répète le processus jusqu'à ce qu'on ait un bon couple (l_1,t_1) et on fait
13: $\boldsymbol{l} \leftarrow l_1$; $\boldsymbol{t} \leftarrow t_1$; **retourner** $(\boldsymbol{l}, \boldsymbol{t})$.
14: **Fin (Tant que)**

avec $bin(a)$ la représentation binaire du nombre a.

4.3.2.2 Deuxième approche de la méthode de calcul direct

- **Profil constant**

Considérons le code polaire $PC(8,4)$ appliqué à la stéganographie. D'après (4.23), les colonnes H_j, $1 \leq j \leq 8$, de la matrice H vérifient les égalités (4.24). Calculons d'abord le syndrome $s = m - xH^T$. S'il est égal à :

- **Synd. 1** : vecteur nul alors le vecteur de changement est aussi égal au vecteur nul ;
- **Synd. 2** : une colonne de H, soit H_j, alors il a comme premier élément 1 et le vecteur de changement e n'a qu'un seul 1 à la position j. Les colonnes H_j ($j = 1$ à n) de H représentent les valeurs binaires des nombres compris entre n et $2n - 1$ (voir, par exemple, (4.21) et (4.23)). Ainsi, à la colonne j, nous avons la représentation binaire de $n + j - 1$. Soit $s = H_j$, ainsi $dec(H_j) = n + j - 1$. D'où $\boldsymbol{j = dec(s) - n + 1}$;
- **Synd. 3** : la somme de deux colonnes (H_1 et H_j) de H alors il a comme premier coefficient 0, (4.21) et (4.23). Sa valeur décimale varie entre 1 et $n - 1$. Le vecteur de changement a deux 1; le premier à la première position et le second à la position j. La

valeur décimale de la somme de H_1 et H_j donne $((n) + (n + j - 1))$ mod $2n = j - 1$. Donc $dec(H_1 + H_j) = j - 1$. D'où $\boldsymbol{j = dec(s) + 1}$.

Ces trois cas sont aussi valables pour les systèmes équivalents. Fort de ces observations, on peut établir une relation entre la valeur décimale du syndrome s et la position des 1 du vecteur de changement e. Comme dans la première approche, une condition nécessaire est $2^{n-k} = 2 \cdot n = 2^{p+1}$. Ce qui est équivaut à $k = n - 1 - \log_2 n = 2^p - 1 - p$. Le domaine de validité est toujours le même et correspond à l'ensemble donné dans (4.29).

- **Cas du papier mouillé**

Soit $\mathcal{J}$ l'ensemble des positions à verrouiller (éléments mouillés). Ainsi :

- si le syndrome s est dans le **Synd. 1** alors on ne fait rien car e est aussi un vecteur nul.
- si s est dans **Synd. 2** alors on considère, par quadruplets, les valeurs décimales des n syndromes dont les vecteurs de changement ont un seul 1. Ces syndromes correspondent aux n colonnes de H et constituent la moitié de tous les $2^{n-k} = 2n$ syndromes possibles. On a $n/4$ quadruplets et chacun est constitué de quatre syndromes consécutifs [76]:

$$Q_i\colon \{n + 4 \cdot i - 4;\ n + 4 \cdot i - 3;\ n + 4 \cdot i - 2;\ n + 4 \cdot i - 1\}, \tag{4.35}$$

avec $i = 1, \dots, n/4$. L'indice i de chaque quadruplet Q_i d'un syndrome donné s se calcule par $i = \lceil (dec(\mathrm{s}) - n + 1)/4 \rceil$.

Soient s, s_1, s_2 et s_3 les syndromes constituant un quadruplet Q et e, e_1, e_2 et e_3 leurs vecteurs de changement correspondant, $\mathrm{e}_i \cdot \mathrm{H}^{\mathrm{T}} = \mathrm{s}_i$, $i = 1, 2, 3$. Dans chaque quadruplet, un syndrome est égal à la somme des trois autres donc $\mathrm{s} = \mathrm{s}_1 + \mathrm{s}_2 + \mathrm{s}_3$. Soit $\mathrm{e}_4 = \mathrm{e}_1 + \mathrm{e}_2 + \mathrm{e}_3$, $\mathrm{e}_4 \cdot \mathrm{H}^{\mathrm{T}} = \mathrm{e}_1 \cdot \mathrm{H}^{\mathrm{T}} + \mathrm{e}_2 \cdot \mathrm{H}^{\mathrm{T}} + \mathrm{e}_3 \cdot \mathrm{H}^{\mathrm{T}} = \mathrm{s}_1 + \mathrm{s}_2 + \mathrm{s}_3 = \mathrm{s}$. Ainsi e_4 est dans le coset de s. Par conséquent, pour verrouiller la position j, on prend comme vecteur de changement e_4. Les vecteurs e, e_1, e_2 et e_3 ont chacun un seul 1 respectivement aux positions j, h, l et t qui sont toutes différentes. Donc e_4 a trois 1 aux positions h, l et t. Si au moins une de ces trois positions est dans $\mathcal{J}$ alors on cherche un autre vecteur de changement avec des 1 à des positions n'appartenant pas à $\mathcal{J}$. On choisit un couple du triplet contenant le ou les deux éléments appartenant à $\mathcal{J}$. Le couple choisi, soit (h,l), est ensuite remplacé par un autre couple (f, g) qui ne soit pas inclus dans $\mathcal{J}$ avec les égalités d'un équivalent du système (4.24) (voir **Algorithme**

4.2). Le nouveau vecteur de changement aura des 1 au niveau des positions f, g et t n'appartenant pas à $\mathcal{J}$ et des 0 aux positions remplacées h et l.

- si s est dans le **Synd. 3** (i.e. e a deux 1 aux positions 1 et j dont, au moins, une est dans $\mathcal{J}$), alors nous cherchons, avec (4.24) ou un équivalent, le couple (h,l) n'appartenant pas à $\mathcal{J}$. Le vecteur de changement aura ainsi deux 1 aux positions h et l.

Le nouveau schéma proposé peut se résumer comme suit [76]:

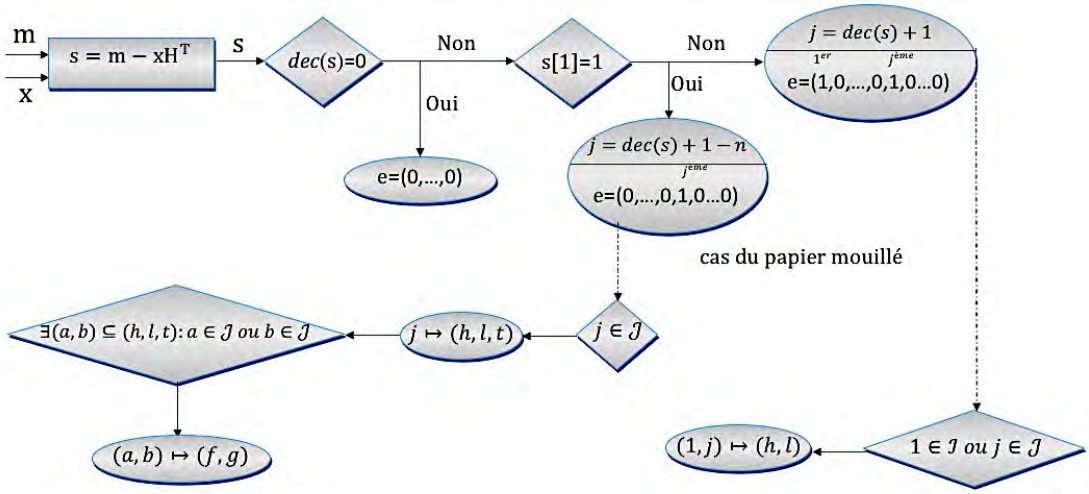

Figure 4.8 : Nouveau schéma de stéganographie par codage polaire.

La partie supérieure du schéma concerne le cas du profil constant avec les trois cas possibles. Dans le cas du papier mouillé, on continue avec la partie inférieure en procédant au remplacement des positions mouillées. Après le remplacement, les nouvelles positions sont mises à 1 et les anciennes remises à 0 dans le vecteur de changement e. Après avoir calculé e, on peut obtenir le vecteur stégo y par $y = x + e$.

4.3.3 Comparaison des complexités des deux schémas

Nous avons représenté dans la Figure 4.9 l'efficacité d'insertion $emb_{effc} = m/D(x,y)$ de la méthode proposée dans le cas d'un canal à papier mouillé en fonction de l'humidité relative $\tau = |\{i \mid \rho_i = \infty\}|/n$. Nous avons verrouillé $n/2 - 1$ éléments et donc $\tau = (n/2 - 1)/n$.

Pour une humidité relative variant entre $0,25$ et $0,5$, l'efficacité d'insertion augmente de $2,4$ à $5,9$. L'augmentation est d'autant plus rapide que l'humidité relative est grande. Cela montre les bonnes performances en termes d'efficacité d'insertion.

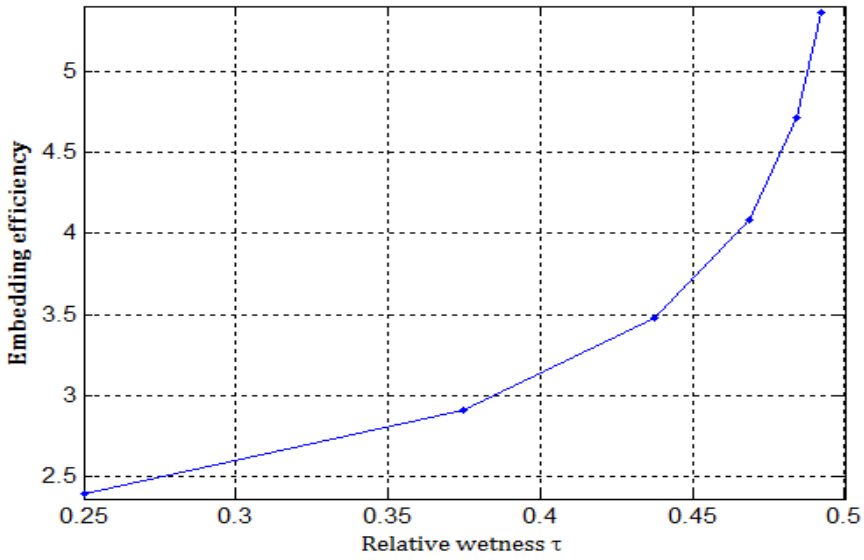

Figure 4.9 : Efficacité d'insertion du schéma pour les codes à papier mouillé.

Nous avons également observé la variation de complexité du premier schéma PCS et celle de l'algorithme proposé dans cette sous-section. Afin de comparer leur complexité nous mesurons la quantité de ressources (en temps) nécessaire pour la résolution du problème de minimisation de l'impact d'insertion (ici recherche du vecteur de modification). Pour ce faire, nous avons observé leur temps d'exécution sur un ordinateur. Nous avons fait plusieurs tests à partir d'un ordinateur équipé d'un processeur Intel Pentium Dual CPU 3.46GHz et doté d'une mémoire physique totale de $2Go$.

Nous avons choisi un code polaire de longueur de bloc $n \in \mathcal{N}$ et de dimension $k \in \mathcal{K}$ car notre algorithme s'applique avec ces valeurs (cf. relation (4.29)). Pour chaque couple $(n, k) \in (\mathcal{N}, \mathcal{K})$, nous avons généré de façon aléatoire 20 vecteurs de couverture et 20 messages. Nous avons calculé ensuite la valeur moyenne des temps d'exécution (en seconde) de l'insertion des messages dans les vecteurs de couverture. Ce calcul est fait pour les deux algorithmes.

Les résultats obtenus pour le profile constant et papier mouillé sont représentés respectivement par les courbes Figure 4.10 et Figure 4.11. Chaque courbe représente spécifiquement la durée moyenne du temps d'exécution de l'algorithme de recherche du vecteur de modifications correspondant au syndrome calculé à partir du vecteur de couverture et du message qui sont générés de façon aléatoire. La courbe du temps d'exécution de l'algorithme de PCS est en bleu et celle en rouge représente celui du nouvel algorithme proposé.

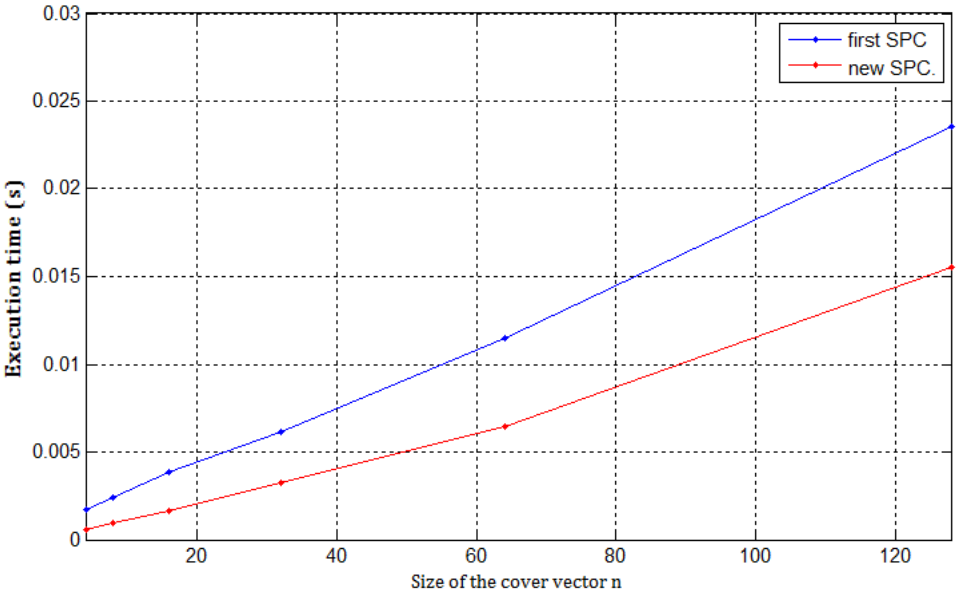

Figure 4.10 : Temps d'exécution des deux schémas pour le profil constant.

Nous voyons que le temps d'exécution du nouvel algorithme est plus faible que celui du schéma PCS [75] aussi bien pour le profil constant (Figure 4.10) que pour le cas du papier mouillé (Figure 4.11). La différence entre les temps d'exécution augmente avec la taille du vecteur de couverture n. Cette réduction du temps d'exécution nous permet de nous prononcer sur celle de la complexité. Par conséquent, le schéma proposé dans ce chapitre, permet la minimisation de l'impact d'insertion avec une complexité plus faible par rapport au schéma précédent PCS.

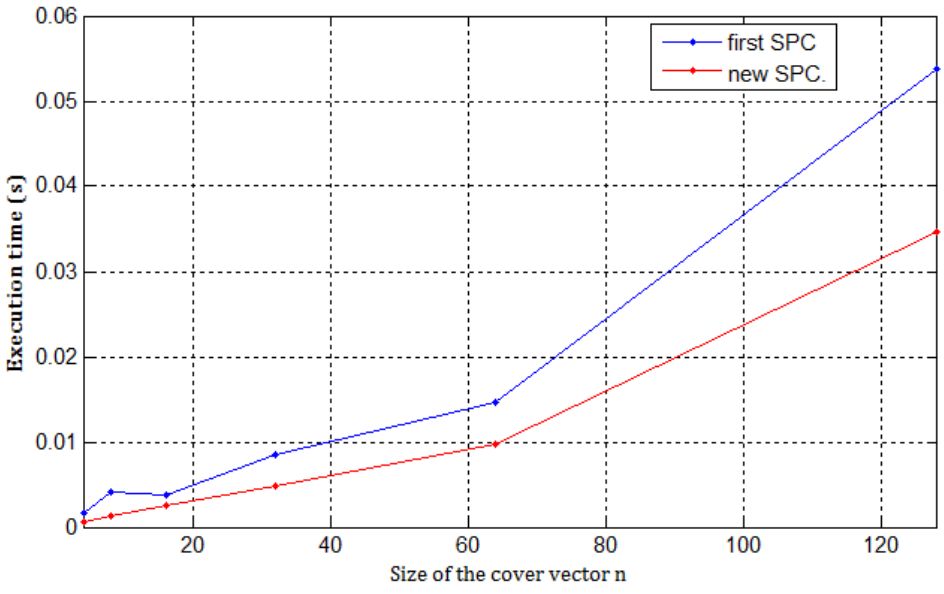

Figure 4.11 : Temps d'exécution des deux schémas pour le canal à papier mouillé.

Nous avons testé le schéma proposé sur des images numériques en niveaux de gris représentées dans le domaine spatial.

4.3.4 Application sur des images avec une permutation des pixels

Nous pouvons tester le schéma stéganographique sur des images de couverture provenant de la base de données BOSSbase (Break Our Stego System) version 1.01 [39] contenant 10000 images en niveaux de gris codées sur 8-bits de taille 512×512 pixels et de format pgm obtenues à partir d'images naturelles de diverses tailles, pris par huit caméras différents, puis retaillées et rognées.

Pour rendre moins indétectable le message, nous choisissons de permuter les pixels de l'image de couverture avant l'insertion. Puisque les images ont une taille fixe de 512×512 pixels et 512 est une puissance de 2, nous pouvons utiliser la matrice de permutation-bits B_{512}, décrit dans le chapitre 3 ($\mathrm{G}_n = \mathrm{B}_n\mathrm{G}_2^{\otimes p}$), pour la permutation. Cette matrice de permutation B_{512} peut être utilisée pour permuter aussi bien les lignes mais aussi les colonnes de l'image de couverture avant d'effectuer l'insertion du message secret comme l'illustre la Figure 4.12. Après permutation, l'image obtenue est subdivisée en $512/n = 2^{9-p}$ blocs car la longueur de bloc du code polaire utilisé est aussi une puissance de 2, $n = 2^p$. En plus, nous pouvons permuter les lignes et les colonnes de ces blocs d'images de $n\times n$ pixels en utilisant la matrice de permutation B_n. Ainsi, on répète le même processus, comme montré par la Figure 4.12, avec des blocs d'images I_{RC}^i et B_n [76]. Ce choix de permutation est un exemple parmi tant d'autres (nous pouvions utiliser la matrice R_n par exemple) et peut être partagé, comme clé secrète, entre l'émetteur et le destinataire du message.

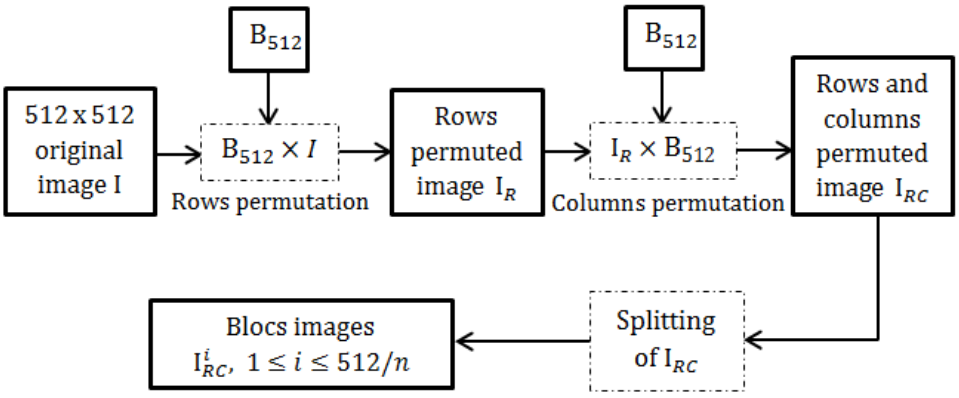

Figure 4.12 : Permutation et division des images.

Ainsi, les modifications vont s'éparpiller à travers des pixels isolés de l'image de couverture rendant moins détectable le message secret et permettant une insertion plus sécurisée.

Après l'insertion, il est nécessaire de retrouver l'ordre initial des pixels de l'image. Pour se faire, nous allons encore utiliser la matrice B_n puisqu'elle est inversible et égale à son propre inverse. Notons que la technique de permutation a été utilisée dans le passé mais son application dépendait d'une clé obtenue à partir d'un mot de passe. Dans un tel cas le récepteur avait besoin de la bonne clé secrète pour pouvoir répéter la permutation qui avait une complexité linéaire $O(n)$ dans [4]. Notre technique de permutation dépend uniquement de la matrice B_n qui est déjà utilisée dans la construction du code polaire donc connu de l'émetteur et du destinataire du message.

De cette manière, nous avons quatre images au choix pour faire l'insertion du message secret. Nous pouvons utiliser l'image de couverture original (initiale) I, ou l'image dont les lignes sont permutées I_R, ou l'image dont les colonnes sont permutées $I_{C,}$ ou encore l'image dont les lignes et les colonnes sont permutées I_{RC}. Ce choix secret peut être partagé avec le destinataire et reste inconnu de toute tierce personne. L'image '28.pgm' de BOSSbase est utilisée pour illustrer les effets de la permutation. L'image originale et les trois images permutées sont montrées par la Figure 4.13 (en haut à gauche l'image originale, en haut à droite l'image avec des lignes permutées, en bas à gauche l'image avec des colonnes permutées et en bas à droite l'image avec des lignes et des colonnes permutées).

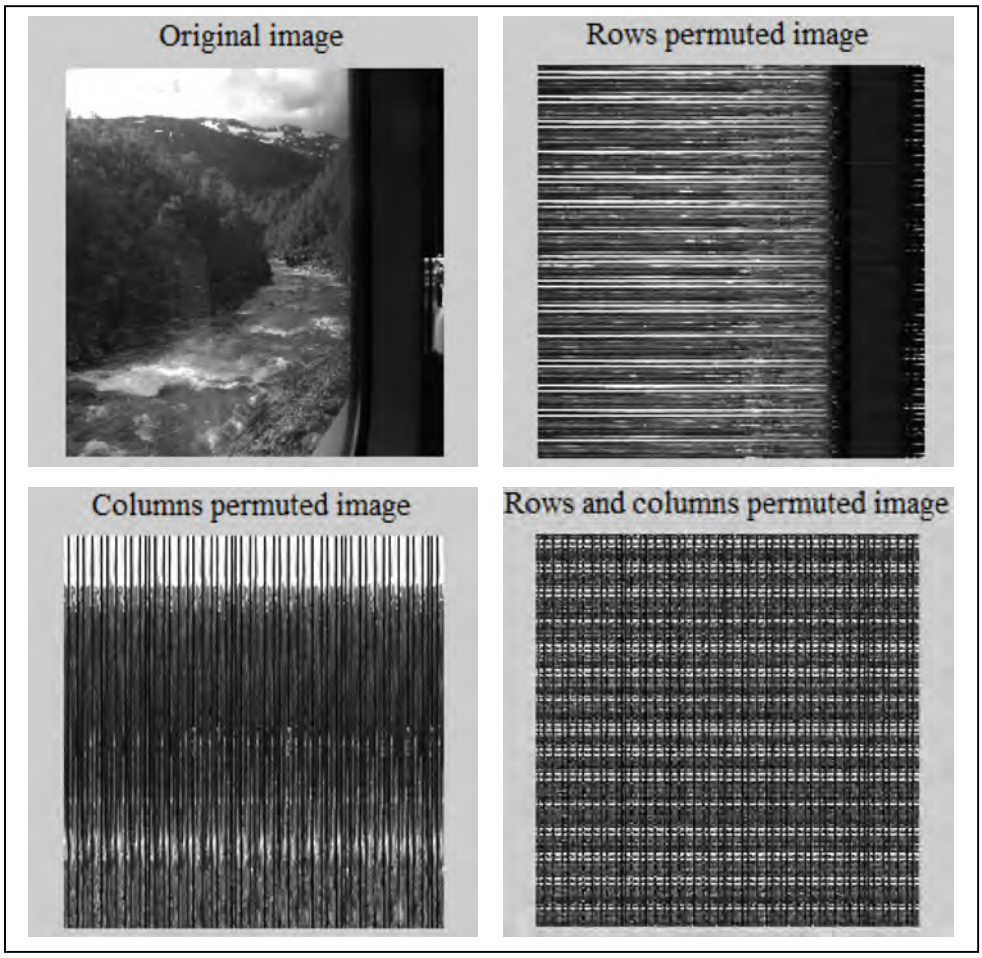

Figure 4.13 : L'image originale et les différentes images permutées.

Comme nous pouvons le constater, la bande noire sur les colonnes situées à droite de l'image originale est aussi visible sur l'image avec des lignes permutées. De même, les pixels en blanc sur la partie supérieure (en haut) restent en haut de l'image avec des colonnes permutées. Par contre, pour l'image avec des lignes et des colonnes permutées les pixels sont uniformément distribués à travers l'ensemble des positions.

Pour comparer la position des changements pour une insertion avec ou sans permutation, nous avons fait la différence entre l'image originale '28.pgm' et les images stégo. Dans la Figure 4.14, les pixels blancs (niveau de gris égal à 1) correspondent aux modifications par ± 1 et les pixels noirs (niveau de gris 0) correspondent aux pixels qui n'ont pas changé. Avec l'image permutée lignes et colonnes, les changements sont uniformément distribués sur l'ensemble de l'image (à droite) contrairement à l'image sans permutation (à gauche) dans laquelle les changements sont situés en haut de l'image.

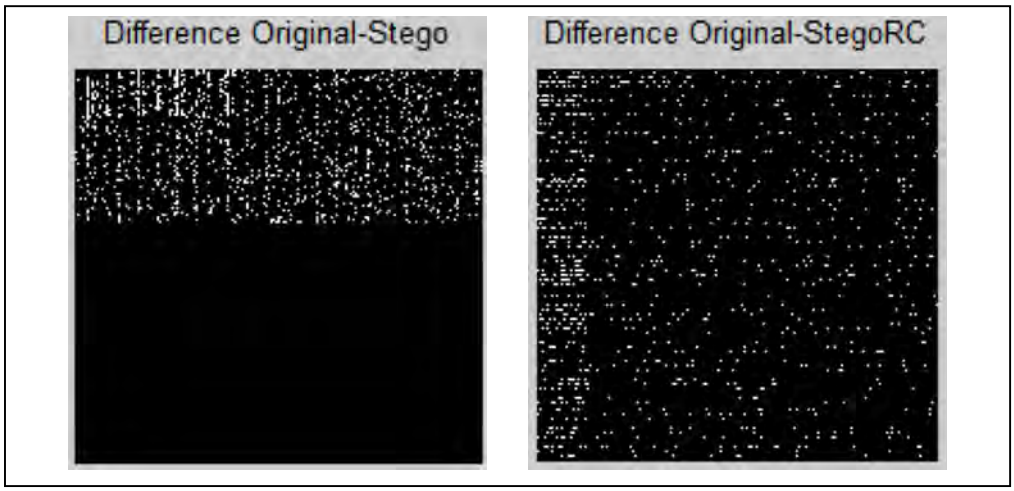

Figure 4.14 : Positions des modifications sur l'image '28.pgm' non-permutée (à gauche) et dont les lignes et les colonnes sont permutées (à droite) pour un taux de 0,2 bpp est inséré.

Conclusion

Nous venons de définir un schéma de stéganographie a profil constant et à papier mouillé basé sur les codes polaires et ayant une bonne efficacité d'insertion. Le schéma proposé est composé de deux parties : une première partie donnant une solution de départ et une seconde partie offrant une convergence vers la solution optimale en utilisant la programmation linéaire. Nous avons pu montrer, en l'appliquant sur des images différentes, que l'indétectabilité visuelle et même statistique en utilisant les histogrammes est atteinte. Nous avons également calculé le PSNR qui varie entre $55\ dB$ et $66\ dB$. Ce qui est plus grand que la valeur limite $35\ dB$. Pour améliorer la complexité du premier schéma, nous avons proposé deux nouvelles méthodes qui permettent de réduire de façon significative la complexité. La première méthode utilise les tables de correspondance. Pour appliquer cette méthode, nous avons besoin de stocker les tables en mémoire et au besoin les consulter. Afin d'éviter ce stockage, une seconde méthode, plus simple, est proposée. Elle exploite la forme des syndromes pour calculer les leaders de cosets. Les deux méthodes proposées dans ce chapitre permettent de minimiser l'impact d'insertion. Nous avons également appliqué le schéma sur des images numériques dont les pixels sont préalablement permutés par la matrice de permutation bit-reversal avant insertion du message pour rendre plus sûr le stégo-système.

Chapitre 5: STEGANOGRAPHIE ADAPTATIVE BASEE SUR LES DECODAGES ALP ET SCL DES CODES POLAIRES

Introduction

Nous avons présenté dans le chapitre précédent l'utilisation des codes polaires en stéganographie. Comme nous l'avons vu, dans le chapitre 2, plusieurs types de décodage sont proposés pour les codes polaires tels que le SC [25], le SC-List (SCL) [27], le LP [29] et l'ALP (Adaptive LP) [30]. Le décodage LP/ALP présente l'avantage d'être ML-certificate [29], c'est-à-dire lorsque le décodage réussit la solution trouvée est le mot de code donné par le décodage ML. Ce qui constitue un atout très important pour la recherche du stégo médium optimal en stéganographie. En outre, LP/ALP a de fortes ressemblances avec le problème de minimisation de l'impact d'insertion en stéganographie. Nous proposons ainsi, dans ce chapitre, une nouvelle méthode pratique pour minimiser une fonction de distorsion additive convenablement définie en utilisant le décodage adaptative par programmation linéaire des codes polaires basé sur un nouveau facteur de graphe réduit. Les résultats expérimentaux montrent que la méthode de stéganographie adaptative proposée offrent une meilleure efficacité d'insertion que les codes STC souvent utilisés dans l'état de l'art. Cependant, le décodage peut échouer à fournir un stégo médium valide. La technique de décodage naturelle des codes polaires étant SC dont la version améliorée SCL offre, actuellement, aux codes polaires des taux d'erreur plus faibles par rapport aux codes LDPC et Turbo codes en longueurs finies, adaptées à la pratique. Nous présenterons un schéma de stéganographie adaptatif basé sur le décodage SCL des codes polaires permettant de minimiser la fonction de distorsion additive. Les résultats expérimentaux confirment les bonnes performances théoriques.

5.1 Optimalité des codes polaires au codage source et canal

Nous savons que le processus d'insertion stéganographique peut être vu comme un problème d'encodage source, soit avec le critère de fidélité tel que décrit par Shannon [22] ou avec des informations supplémentaires disponibles du côté de l'émetteur (le problème de Gelfand-Pinsker) [77]. En outre, les rôles des problèmes d'encodage et de décodage sont inversés pour le codage canal par rapport au codage source, c'est à dire la tâche de l'encodage/décodage en codage canal est la même que celle du décodage/encodage en codage

source. C'est ce qui explique l'utilisation du décodage canal en stéganographie. Pour la compression avec perte d'une source binaire symétrique avec une fonction de distorsion de Hamming, les codes polaires combinés avec un algorithme d'encodage SC de faible complexité sont présentés comme constituant le premier schéma de codage pratique qui permet d'atteindre la limite taux-distorsion [26]. De cette manière, les codes polaires sont également optimaux pour les problèmes de codage source de Slepian-Wolf, de Wyner-Ziv et de Gelfand-Pinsker. Pour le codage canal les codes polaires atteignent la capacité du canal de transmission [25]. Un autre avantage et raison pour laquelle nous nous intéressons aux codes polaires est leur optimalité pour la méthode d'insertion adaptative PLS qui consiste à insérer un message donné tout en minimisant la distorsion induite, comme souligné par Filler et al. [19].

5.2 Stéganographie adaptative basée sur le décodage ALP

Nous avons proposé dans [75] une méthode PCS (Polar Coding Steganography) composée de deux étapes. La première étape donne une solution initiale du problème stéganographique et la seconde fournit la solution optimale d'un solveur LP. Dans [45], [46], d'autres méthodes ont été présentées afin de minimiser l'impact d'insertion pour les cas de profil constant et du papier mouillé, tout en réduisant la complexité par rapport à PCS. Dans celle section, nous définissons la nouvelle méthode qui minimise une fonction de distorsion additive arbitraire convenablement définie en utilisant le décodage ALP des codes polaires [30]. Tout d'abord, pour construire des codes polaires pour la stéganographie, nous suivons les différentes étapes indiquées dans [75]. Rappelons que pour des images de couvertures numériques, l'expéditeur insert son message tout en minimisant une certaine fonction de distorsion D entre la couverture x et le stégo médium y. Cette distorsion peut être exprimée par :

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \varrho_i \cdot [\, y_i \neq x_i] \tag{5.1}$$

où $y_i \in I_i = \{x_i, \bar{x_i}\}$ et $\varrho_i = |\rho_i(\mathrm{x}, x_i) - \rho_i(\mathrm{x}, \bar{x}_i)| \geq 0$. Considérons que $\rho_i(\mathrm{x}, x_i) = 0$; implique que $\varrho_i = |\rho_i(\mathrm{x}, \bar{x}_i)| \in [0, \infty[$. Par conséquent, la distorsion devient

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} |\rho_i(\mathrm{x}, \bar{x}_i)| \cdot e_i\,, \tag{5.2}$$

où $e_i = [y_i \neq x_i]$, e étant le vecteur de modifications binaire. Pour simplifier la notation posons $\rho_i = |\rho_i(\mathrm{x}, \bar{x}_i)|$. Nous pouvons donc écrire $D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \rho_i \cdot e_i$.

5.2.1 Définition de l'algorithme du schéma

Rappelons que le décodage ML pour le code C est défini par $\arg\min_{c\in C} c\,\gamma^T$, avec γ le vecteur des rapports de vraisemblance logarithmiques LLR $\gamma_i = \log \frac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i=1]}$. La relaxation du ML correspondant au décodeur LP est définie par $\arg\min \langle c,\gamma\rangle,\ s.t.\ v \in \mathcal{P}(H_{\mathcal{P}}), \mathcal{P}(H_{\mathcal{P}})$ étant le polytope défini pour les codes polaires dont la projection $\bar{\mathcal{P}}(H_{\mathcal{P}}) = \{c \in \{0,1\}^n | \exists\ (a,u) : (c,a,u) = v \in \mathcal{P}(H_{\mathcal{P}})\}$ (sous-section 3.6.4.3). Pour ALP toutes les contraintes de contrôle de parité du polytope ne sont pas ajoutées en même temps au début, comme dans LP [69], mais plutôt de façon itérative. Nous allons définir un schéma basé sur le décodage LP/ALP.

Nous allons d'abord exploiter le fait que le vecteur coût γ dans la LP peut être uniformément redimensionné sans pour autant influer sur la solution du problème [69] (chapitre 3). L'objectif est de redimensionner le vecteur coût de distorsion de telle sorte que ses composants aient le même domaine de variation que les valeurs de LLR γ_i. Divisons, dans un premier temps, le vecteur coût de distorsion par sa moyenne comme suit $\rho_i^{norm} = \frac{\rho_i}{mean(\rho)}$, où $mean(\rho) = \sum_{i=1}^{n} \rho_i\,/n$. Donc $\rho_i^{norm} \in [0,\beta] \subset [0,\infty[$, avec $\beta = \frac{max(\rho)}{mean(\rho)}$. Appliquons ensuite le logarithme à ces valeurs $\rho_i^{log\_norm} = \log(\rho_i^{norm})$. Ainsi, le vecteur des valeurs de LLR et le vecteur coût de distorsion normalisé et logarithmique ont le même domaine de variation. Pour appliquer le décodage ALP en stéganographie, posons $\gamma(y) = \rho^{log\_norm}(x,y)$, c'est-à-dire $\rho_i^{log\_norm} = \gamma_i(y),\ i = 1,\cdots,n$. Pour simplifier la notation nous écrivons simplement $\rho^{log\_norm}(x,y) = \rho$. La recherche du vecteur des modifications optimal e_{opt} peut être écrite sous forme de décodage stéganographique ML [75]

$$\begin{aligned} &\arg\min_{e}\ \langle e,\rho\rangle \\ &s.t.\ \ e \in \mathcal{C}(s) \end{aligned} \tag{5.3}$$

avec $D(x,y) = \sum_{i=1}^{n} \rho_i \cdot e_i = \langle e,\rho\rangle$, $s = m - LSB(x)\cdot H^T$, m est le message secret, x l'image de couverture et $\mathcal{C}(s)$ le *coset* du syndrome s défini par $\mathcal{C}(s) = \{e \in \{0,1\}^n : eH^T = s\}$. L'insertion ML stéganographique devient le décodage LP stéganographique

$$\begin{aligned} &\arg\min_{e}\ \langle e,\rho\rangle \\ &s.t.\ \ e \in conv(\mathcal{C}(s)) \end{aligned} \tag{5.4}$$

où nous appelons $conv(\mathcal{C}(\mathrm{s}))$ *coset-word polytope* qui est défini comme étant le *convex hull* de l'ensemble des coset-words possibles. Le *polytope fondamental stéganographique* correspondant à la matrice de contrôle de parité H et du syndrome s est défini par $\mathcal{P}(\mathrm{H},\mathrm{s}) = \bigcap_{j\in J} conv(\mathcal{C}_j(\mathrm{s}))$, où chaque ligne j de H définit un *coset local* $\mathcal{C}_j(\mathrm{s}) = \left\{\mathrm{e} \in \{0,1\}^n \colon \langle \mathrm{e}, h_j\rangle = s_j\right\}$, avec $\mathcal{C}(\mathrm{s}) = \bigcap_{j\in J} \mathcal{C}_j(\mathrm{s})$ et h_j est la $j-$ème ligne de H. Nous savons que le problème LP est décrit par

$$\begin{aligned} &\arg\min_{\mathrm{c}} \quad \langle \mathrm{c}, \gamma\rangle \\ &s.t. \quad \mathrm{c} \in \mathcal{P}(\mathrm{H}) \end{aligned}, \tag{5.5}$$

où le polytope $\mathcal{P}(\mathrm{H})$ fondamental peut être décrit par

$$\sum_{i\in V}(1-c_i) + \sum_{i\in \mathcal{N}(j)\setminus V} c_i \geq 1 \;\; \forall \; V \subseteq \mathcal{N}(j) \colon |V| \; est \; impair, \tag{5.6}$$

Et le décodage LP est

$$\begin{aligned} &\arg\min_{\mathrm{c}} \quad \langle \mathrm{c}, \gamma\rangle \\ &s.t. \quad 0 \leq c_i \leq 1, \qquad \forall\, i = 1, \dots, n \\ &\qquad \sum_{i\in V}(1-c_i) + \sum_{i\in \mathcal{N}(j)\setminus V} c_i \geq 1, \\ &\qquad \forall\, j = 1, \dots, m, \;\; V \subseteq \mathcal{N}(j) \;\; : \; |V| \; est \; impair \end{aligned} \tag{5.7}$$

Ainsi, le décodage LP stéganographique peut être écrit comme suit :

$$\begin{aligned} &\arg\min_{\mathrm{e}} \quad \langle \mathrm{e}, \rho\rangle \\ &s.t. \quad \mathrm{e} \in \mathcal{P}(\mathrm{H},\mathrm{s}) \end{aligned} \tag{5.8}$$

Le polytope stéganographique $\mathcal{P}(\mathrm{H},\mathrm{s})$ peut être décrit par les contraintes de contrôle de parité qui dépendent des valeurs des bits individuels du syndrome s. Supposons que l'expéditeur obtient son message secret sous forme d'un flux de bits pseudo-aléatoire. Ainsi, le syndrome peut contenir des bits 0 (0-bit) et des 1 (1-bit). L'expression $\langle \mathrm{e}, h_j\rangle = s_j$ est équivalent à $\left[\sum_{i=1}^{n} h_{ji} e_i\right]_{mod\ 2} = \left[\sum_{i\in\mathcal{N}(j)} e_i\right]_{mod 2} = s_j$, $\mathcal{N}(j) = \{i \colon h_{ji} = 1\}$. Alors $\mathcal{C}_j(\mathrm{s}) = \left\{\mathrm{e} \in \{0,1\}^n \colon \left[\sum_{i\in\mathcal{N}(j)} e_i\right]_{mod\ 2} = s_j\right\}$.

- Si $s_j = 0$, $\mathcal{C}_j(\mathrm{s}) = \mathcal{C}_j$, $\mathcal{C}_j$ étant le coset code, et e est composé d'un nombre pair de 1-bits. Ainsi, pour tout sous-ensemble $V \subseteq \mathcal{N}(j)$ de taille paire, en mettant $e_i = 1$ pour tout $i \in V$, $e_i = 0$ pour tout $i \in \mathcal{N}(j)\backslash V$ et e_i arbitraire pour tout $i \notin \mathcal{N}(j)$, on obtient e $\in \mathcal{C}_j(\mathrm{s}) = \mathcal{C}_j$. Pour les sous-ensembles $V \subseteq \mathcal{N}(j)$ de taille impaire, les contraintes d'inégalités (5.6) sont vérifiées, c'est-à-dire $\sum_{i\in V}(1-e_i) + \sum_{i\in\mathcal{N}(j)\backslash V} e_i \geq 1$.
- Si $s_j = 1$, e est composé d'un nombre impaire de 1-bits. Ainsi, pour tout sous-ensemble $V \subseteq \mathcal{N}(j)$ de taille impaire, en posant $e_i = 1$ pour tout $i \in V$, $e_i = 0$ pour tout $i \in \mathcal{N}(j)\backslash V$ et e_i arbitraire pour tout $i \notin \mathcal{N}(j)$, mène à e $\in \mathcal{C}_j(\mathrm{s})$. Pour les sous-ensembles $V \subseteq \mathcal{N}(j)$ de taille paire, (5.6) est vérifiée. Finalement, pour un 0-bit du syndrome, les contraintes de contrôle de parité ne vont pas changer et seront les mêmes que dans (5.6). Alors que pour un 1-bit du syndrome la définition va légèrement changer. Ainsi, le polytope stéganographique $\mathcal{P}(\mathrm{H}, \mathrm{s})$ peut être décrit par un ensemble d'inégalités linéaires dans le décodage LP stéganographique formulé comme suit:

$$\begin{aligned} &\arg\min_{\mathrm{e}} \quad \langle \mathrm{e}, \rho \rangle \\ &s.t. \;\; 0 \leq e_i \leq 1, \qquad \forall\, i = 1, \dots, n \\ &\qquad \sum_{i\in V}(1-e_i) + \sum_{i\in\mathcal{N}(j)\backslash V} e_i \geq 1, \\ &\forall\; V \subseteq \mathcal{N}(j) : \; |V| \; est \begin{cases} impair & si\; s_j = 0 \\ pair & si\; s_j = 1 \end{cases} \end{aligned} \tag{5.9}$$

Ainsi, une contrainte qui génère une violation (ou cut) signifiera que

$$\begin{aligned} &\sum_{i\in V} e_i - \sum_{i\in\mathcal{N}(j)\backslash V} e_i > |V| - 1 \\ &\forall\, j = 1, \dots, m, \;\; V \subseteq \mathcal{N}(j) : \\ &|V| \; est \; impair \; si \; s_j = 0 \; ou \; pair \; si \; s_j = 1. \end{aligned} \tag{5.10}$$

Avant de définir le décodage ALP pour le Polar Coding Steganography (ALP-PCS), nous donnons d'abord l'algorithme qui recherche les inégalités de contrôle de parité qui violent la contrainte pour un vecteur de modification donné. L'algorithme de recherche de violation pour la stéganographie en codage polaire est donné par **Algorithme 5.1** [78].

Algorithme 5.1 Modified Cut Search Algorithm for Steganography (MCSA-S)

Entrées : nœud de contrôle de parité $\boldsymbol{j}$, vecteur de modifications $\mathbf{e}$ et le bit $\boldsymbol{s_j}$ du syndrome.

Sortie : ensemble des nœuds de variable en violation (cut) $\boldsymbol{\mathcal{V}}$.

$\boldsymbol{\mathcal{V}} \leftarrow \{i \in \mathcal{N}(j) |\ \boldsymbol{e_i} > 0.5\}$
si ($|\boldsymbol{\mathcal{V}}|$ est pair et $\boldsymbol{s_j} = 0$) ou ($|\boldsymbol{\mathcal{V}}|$ est impair et $\boldsymbol{s_j} = 1$) **alors**
    Identifier le symbole le moins certain $i^* \leftarrow \arg \min_{i \in \mathcal{N}(j)} |0.5 - \boldsymbol{e_i}|$.
    si $i^* \in \boldsymbol{\mathcal{V}}$ **alors**
        Supprimer le de $\boldsymbol{\mathcal{V}}$ ($\boldsymbol{\mathcal{V}} \leftarrow \boldsymbol{\mathcal{V}} \setminus \{i^*\}$).
    sinon
        Ajouter cette indice à $\boldsymbol{\mathcal{V}}$ ($\boldsymbol{\mathcal{V}} \leftarrow \boldsymbol{\mathcal{V}} \cup \{i^*\}$).
    fin si
fin si
si $\sum_{i \in \mathcal{V}} \boldsymbol{e_i} - \sum_{i \in \mathcal{N}(j) \setminus \mathcal{V}} \boldsymbol{e_i} > |\boldsymbol{\mathcal{V}}| - 1$ **alors**
    L'ensemble $\boldsymbol{\mathcal{V}}$ est une violaltion au vecteur $\mathbf{e}$ sur le nœud de contrôle $\boldsymbol{j}$.
sinon
    Il n'y a aucune violation au vecteur $\mathbf{e}$ sur le nœud de contrôle $\boldsymbol{j}$ (donc $\boldsymbol{\mathcal{V}} \leftarrow \emptyset$).
fin si
retourner $\boldsymbol{\mathcal{V}}$.

Afin d'utiliser le décodage ALP en stéganographie, nous définissons un polytope basé sur un nouveau facteur de graphe réduit. La représentation du facteur de graphe original [25], [29], [30] est basée sur la matrice génératrice du code polaire contrairement aux codes LDPC dont le graphe, appelé graphe de Tanner, est basé sur la matrice de contrôle de parité.

Nous savons que les codes polaires sont définis par une matrice génératrice inversible G_n via la relation $\mathrm{c} = \mathrm{u}\mathrm{G}_n$. Cette transformation est alors inversible et $\mathrm{u} = \mathrm{c}\mathrm{G}_n^{-1}$, avec $\mathrm{G}_n^{-1} = \mathrm{G}_n$. Ce qui donne $\mathrm{u} = \mathrm{c}\mathrm{G}_n$. Nous pouvons décomposer le mot source en deux parties $\mathrm{u} = (\mathrm{u}_A, \mathrm{u}_{A^c})$, où le mot d'information $\mathrm{u}_A = (u_i : i \in A)$ et le mot fixé (frozen) $\mathrm{u}_{A^c} = (u_i : i \in A^c)$ [73]. Ainsi, nous pouvons écrire $\mathrm{u} = (\mathrm{u}_A, \mathrm{u}_{A^c}) = (\mathrm{c}\mathrm{G}^A, \mathrm{c}\mathrm{G}^{A^c})$ où G^A et G^{A^c} désignent, respectivement, les sous-matrices de G_n constituées des colonnes d'indices dans A et A^c, respectivement. Il en découle les égalités $\mathrm{G}^{A^c} = \mathrm{H}^{\mathrm{T}}$, $\mathrm{u}_A = \mathrm{c}\mathrm{G}^A$ et $\mathrm{u}_{A^c} = \mathrm{c}\mathrm{G}^{A^c}$. La relation du coset $\mathrm{e}\mathrm{H}^{\mathrm{T}} = \mathrm{s}$ peut être remplacer par $\mathrm{e}\mathrm{G}^{A^c} = \mathrm{s}$. Sachant que pour tout $\mathrm{e} \in \{0,1\}^n$ il existe $\mathrm{u} \in \{0,1\}^n$ tel que $\mathrm{u} = \mathrm{e}\mathrm{G}_n$. Ainsi, nous avons $(\mathrm{u}_A, \mathrm{u}_{A^c}) = \left(\mathrm{e}\mathrm{G}^A, \mathrm{e}\mathrm{G}^{A^c}\right) = (\mathrm{e}\mathrm{G}^A, \mathrm{s})$. Puisque $\mathrm{s} = \mathrm{e}\mathrm{G}^{A^c} = \mathrm{u}_{A^c}$ est connu, alors chercher u revient à chercher u_A tel que $\mathrm{u}_A = \mathrm{e}\mathrm{G}^A$. Donc $\mathrm{e}\mathrm{G}^A + \mathrm{u}_A = 0$. Par suite

$$(\mathrm{e}, \mathrm{u}_A) \begin{bmatrix} \mathrm{G}^A \\ I_k \end{bmatrix} = 0 \qquad (5.11)$$

Soit $z = (e, u_A) \in \{0,1\}^{n+k}$ le vecteur de modification étendu et $H_{\mathcal{B}} = \left[G^{A^T} I_k\right] \in \{0,1\}^{k\times(n+k)}$ la matrice de contrôle de parité correspondante, alors $z\, H_{\mathcal{B}}{}^T = 0$. La matrice $H_{\mathcal{B}}$ correspond au graphe adjacent à une réduction du sparse facteur graphe original Gph_S. Nous notons ce graphe par $Gph_{\mathcal{B}}$ [78], où ses $n + k$ nœuds de variable correspondent aux colonnes de $H_{\mathcal{B}}$ et ses k nœuds de contrôle correspondent aux lignes de $H_{\mathcal{B}}$. Nous pouvons remarquer que le nouveau facteur graphe réduit $Gph_{\mathcal{B}}$ correspond au sparse facteur graphe original Gph_S (Figure 3.13) auquel on enlève les nœuds de variable auxiliaires et les nœuds de variable des frozen bits. Seuls les nœuds de variable du vecteur de modifications e et du mot d'information u_A sont présents dans le graphe. Le polytope correspondant est noté par $\mathcal{B}$. Nous remplaçons la matrice $H_{\mathcal{P}}$ par $H_{\mathcal{B}}$ dans l'Algorithme 1 proposé dans [30]. Par conséquent, le décodage ALP des codes polaires pour la stéganographie est défini par l'**Algorithme 5.2** [78].

Algorithme 5.2 Adaptive LP decoding of Polar Codes for Steganography (ALP-PCS)

Entrées: médium de couverture **x**, message secret **m**, la fonction coût **ρ**, la matrice de contrôle de parité **H** et la matrice adjacente $\mathbf{H_{\mathcal{B}}}$ du nouveau facteur de graphe réduit.

Sortie: Solution optimale du problème ALP-S $\mathbf{e_{opt}}$.

1: Calculer la valeur du syndrome $\mathbf{s} \leftarrow \mathbf{m} - \boldsymbol{LSB}(\mathbf{x}) \cdot \mathbf{H^T} = \mathbf{eH^T}$

2: Initialiser le problème LP avec les contraintes obtenues à partir de $\mathbf{H_{\mathcal{B}}}$.

3: Résoudre le problème LP obtenu avec la fonction coût **ρ** pour obtenir $\mathbf{e_{opt}}$.

4: **si** $\mathbf{e_{opt}}$ est non-intégrale **alors**

5: Construire la matrice d'induction de violation RPC $\widetilde{\mathbf{H}}$ à partir de **H** [21].

6: Appliquer **Algorithme 5.1** à chaque ligne j de $\widetilde{\mathbf{H}}$ avec les entrées j, $\mathbf{e_{opt}}$ et s_j

7: **si** Aucune violation n'est trouvée **alors**

8: Terminer et sortir.

9: **sinon**

10: Ajouter les contraintes de violation trouvées au problème LP et aller à la ligne 3.

Puisque l'opération d'insertion est binaire, après avoir obtenu le vecteur de modifications optimal, e_{opt}, nous pouvons obtenir le vecteur stégo simplement par la relation d'addition binaire $y = x + e_{opt}$.

5.2.2 Résultats des tests

Pour évaluer les performances de notre schéma de stéganographie basé ALP-PC nous donnons l'efficacité d'insertion $emb_{eff} = m/D(\mathrm{x},\mathrm{y})$ pour les cas de profil constant et du papier mouillé. Pour le profil constant nous fixons les symboles du vecteur de modifications à 1, $\rho_i = 1$, pour tout $i = 1, \ldots, n$. Pour une fonction de distorsion, le vecteur coût est normalisé comme décrit dans la section précédente avant d'appliquer le solveur ALP stéganographique.

Figure 5.1 montre la comparaison de l'efficacité d'insertion du code polaire pour le profil constant avec le code syndrome-treillis (STC) [12] et le code Hamming [4].

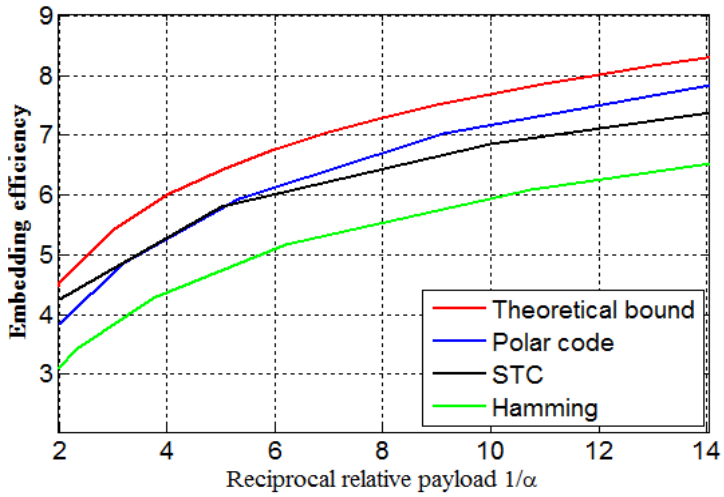

Figure 5.1 : Efficacité d'insertion de ALP-PCS pour le profile constant

L'efficacité d'insertion du code polaire est proche de la limite théorique et est supérieure à celle du STC à partir d'une charge relative $1/\alpha = 5{,}5$. Ce qui est de plus en plus significatif lorsque la charge relative augmente (la réciproque de la charge relative diminue). Cela n'est pas surprenant car les codes polaires sont connus comme étant optimaux [19] pour le problème de stéganographie PLS (Payload-Limited Sender) qui consiste à déterminer la distribution π_{x} qui permet l'insertion d'un message de taille fixe m bits avec une distorsion minimale (voir sous-Section 2.3.2). La seule contrainte était la très grande longueur de bloc nécessaire pour appliquer les codes polaires. Mais, avec le décodage ALP il n'est pas nécessaire d'avoir une longueur de bloc infinie du code polaire pour obtenir de bonnes performances du décodage.

Figure 5.2 donne l'efficacité d'insertion de la méthode proposée pour le cas d'un canal à papier mouillé en fonction de l'humidité relative $\tau = |\{i \mid \rho_i = \infty\}|/n$. Les éléments mouillés

sont caractérisés par $\rho_i = \infty$. Dans la pratique nous leur affectons une très grande valeur, soit ρ_{max}. Les pixels secs seront affectés à une valeur arbitraire comprise entre 0 et ρ_{max}. Pour une humidité relative τ variant entre 0,25 et 0,5, l'efficacité d'insertion augmente de 2,9 à 8,2. L'augmentation est d'autant plus rapide que l'humidité relative est grande. Ceci prouve les bonnes performances en termes d'efficacité d'insertion.

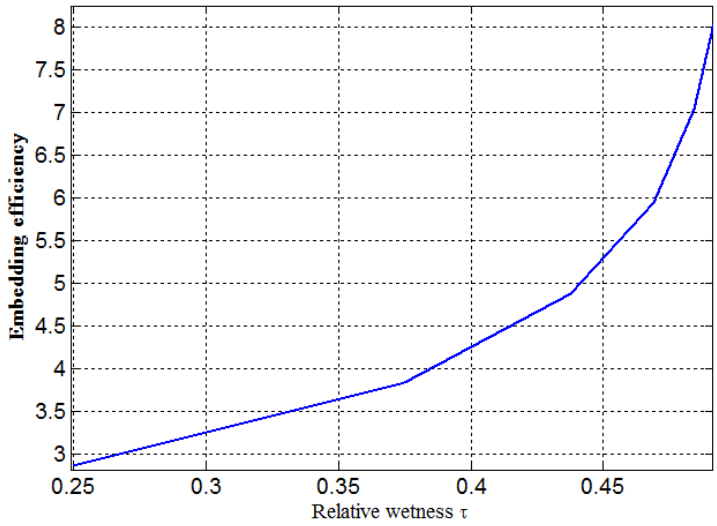

Figure 5.2 : Efficacité d'insertion de ALP-PCS pour les codes à papier mouillé

5.3 Schéma de stéganographie adaptatif basé sur le décodage SCL

Etant donné que SCL est une généralisation du SC, nous allons proposer, dans un premier temps, la méthode basée sur le décodage SC puis allons l'adapter au cas du décodage SCL. Nous allons, auparavant, rappeler brièvement le principe des décodages SC et SCL et expliquer l'optimalité des codes polaires au problème d'insertion en stégaographie.

5.3.1 Choix des décodages SC et SCL

Comme nous l'avons vu dans le chapitre 3, le décodage SC est le premier type de décodage proposé pour les codes polaires. Il est basé sur la structure récursive des codes polaires. Le décodeur fournit en sortie une estimation û du mot source u. Tout d'abord, au premier niveau, le bit $\hat{u}_1$ est donné à partir du mot reçu r. Ensuite, au niveau 2, le bit $\hat{u}_2$ est décodé à partir du mot reçu r et du bit déjà décodé $\hat{u}_1$. De façon générale, le bit $\hat{u}_i$ est estimé à partir du mot reçu r et des bits déjà décodés $\hat{u}_1, \ldots, \hat{u}_{i-1}$. Lorsqu'un bit fixé est rencontré $i \in A^c$, sa valeur est donnée directement sans estimation $\hat{u}_i = u_i$. Le décodage SC repose sur une représentation

sous forme de graphe en FFT (Fast Fourier Transform) dans lequel le processus le calcul des probabilités de transitions se fait de façon séquentielle avec les fonctions de calcul f et g. Ces calculs commencent à partir des probabilités de transitions du canal élémentaire $W(r_i|c_i)$. Les calculs peuvent se faire dans le domaine LR ou LLR. Dans le domaine LR, pour chaque nœud, les deux LRs entrant, notés L_a et L_b sont combinés pour produire le LR correspondant à ce nœud. Le processus de décodage SC peut être représenté sous forme d'un arbre de codes. Le décodage commence à la racine qui a deux branches avec les labelles 0 et 1 et les métriques $W_n^{(i)}(\mathrm{r}|0)$ et $W_n^{(i)}(\mathrm{r}|1)$, respectivement. Le décodage SC choisit l'arrête avec la plus grande métrique et l'autre est abandonné. L'arrête choisi donne, à son tour, deux arrêtes avec des labelles 0 et 1 avec les métriques $W_n^{(i)}(\mathrm{r}, \hat{u}_1|0)$ et $W_n^{(i)}(\mathrm{r}, \hat{u}_1|1)$, respectivement. Généralement, à chaque niveau le bit $\hat{u}_i$ est décodé en comparant les deux métriques $W_n^{(i)}\left(\mathrm{r}, \hat{u}_1^{i-1}\middle|0\right)$ et $W_n^{(i)}\left(\mathrm{r}, \hat{u}_1^{i-1}\middle|1\right)$ si i n'est pas fixé. Si i est fixé, le décodeur SC fournit $\hat{u}_i = u_i$. Cette procédure continue jusqu'aux nœuds feuilles où la dernière estimation est faite avec $\hat{u}_n$.

Proposé par Tal et Vardy [27], le décodage SCL est une amélioration du SC basée sur une liste de taille L. Tout comme le SC, le SCL peut être représenté sous forme d'un arbre de codes. Comme SC, le SCL décode les bits d'entrée $\hat{u}_i$, $i = 1, \ldots, n$, successivement. Contrairement au SC où un seul chemin est conservé après traitement à chaque niveau, SCL permet d'exploiter simultanément jusqu'à L chemins candidats. Pour chaque niveau, le décodeur SCL double le nombre de chemins candidats ($\hat{u}_i = 0$ et $\hat{u}_i = 1$) (Figure 3.12). Si $2L$ chemins candidats sont obtenus alors une procédure d'élagage est utilisée pour sélectionner les L chemins les plus probables. Ces L chemins sont stockés dans une liste pour le traitement au niveau suivant. Pour un bit fixé, le nombre de chemins candidats n'est pas doublé car un tel bit est fixé et sa valeur est connue. A la fin du processus de décodage, le plus probable parmi les L chemins de décodage est sélectionné comme la sortie du décodeur.

Puisque le décodage polaire SCL est une généralisation du décodage classique SC proposé par Arikan, avant de donner notre méthode de stéganographie par décodage polaire SCL, nous proposons tout d'abord la version basée SC et ensuite, expliquons comment l'appliquer au cas du SCL qui est une version SC munie d'une liste.

5.3.2 Version du schéma basée sur le décodage SC

Afin de proposer une méthode de décodage polaire SC en stéganographie nous allons étudier deux étapes : la première consiste à remplacer les frozen bits du mot source par les bits du message secret et la seconde donne les calculs des métriques pour le décodage des non-frozen bits.

5.3.2.1 Remplacement des frozen bits par les bits du message secret à insérer

Nous allons montrer que l'encodage source SC (Successive Cancellation) stéganographique (SC-S) des codes polaires est équivalent au décodage SC des codes polaires dans le codage canal lorsque $\mathrm{u}_{A^c} = \mathrm{m}$. En effet, il est connu, d'après [25], que les codes polaires sont définis à partir d'une matrice inversible G_n via la relation $\mathrm{c} = \mathrm{u}\mathrm{G}_n$. Ainsi, cette transformation est inversée comme suit $\mathrm{u} = \mathrm{c}\mathrm{G}_n^{-1} = \mathrm{c}\mathrm{G}_n$, car $\mathrm{G}_n^{-1} = \mathrm{G}_n$. Par souci de simplification, nous notons $\mathrm{G} = \mathrm{G}_n$. Le mot source peut être divisé en deux parties $\mathrm{u} = (\mathrm{u}_A, \mathrm{u}_{A^c})$, où le mot d'information $\mathrm{u}_A = (u_i : i \in A)$ et le mot fixé $\mathrm{u}_{A^c} = (u_i : i \in A^c)$ [73]. Ensuite, nous pouvons écrire $\mathrm{u} = (\mathrm{u}_A, \mathrm{u}_{A^c}) = (\mathrm{c}\mathrm{G}^A, \mathrm{c}\mathrm{G}^{A^c})$ où G^A et G^{A^c} sont les sous-matrices de G_n constituées des colonnes d'indices dans A et A^c, respectivement. Avec la définition de la matrice de contrôle de parité des codes polaires dans [29], nous avons $\mathrm{G}^{A^c} = \mathrm{H}^{\mathrm{T}}$ avec $\mathrm{u}_A = \mathrm{c}\mathrm{G}^A$ et $\mathrm{u}_{A^c} = \mathrm{c}\mathrm{G}^{A^c} = \mathrm{c}\mathrm{H}^{\mathrm{T}}$.

La relation de coset $\mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{m}$ en stéganographie peut alors être réécrite de façon équivalente $\mathrm{y}\mathrm{G}^{A^c} = \mathrm{m}$. Soit u tel que $\mathrm{u}_{A^c} = \mathrm{m}$. Ainsi, la recherche de mot stégo y tel que $\mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{y}\mathrm{G}^{A^c} = \mathrm{m}$ est équivalente au décodage d'un code polaire lorsque les bits fixés ne sont pas tous nécessairement mis à zéro, mais ils sont identiques aux bits du message secret. Nous obtenons ainsi un code polaire $\mathrm{m}-$coset. De ce fait, nous pouvons redéfinir la technique de décodage SC classique en tenant compte du fait que les bits du message seront les bits fixés dans le mot source. Notons toutefois que ce choix des valeurs des bits fixés n'affecte pas les performances du décodeur polaire [25]. Ce raisonnement s'inspire de celui fourni dans la section 5.2.1, lors de la définition du polytope de code polaire stéganographique.

Pour le décodage SC original, connaissant les bits fixés u_{A^c}, on donne le mot source $\mathrm{u} = (\mathrm{u}_A, \mathrm{u}_{A^c}) = \{u_i | \, i \in A \; ou \; i \in A^c\}$ tel qu'il existe un mot de code c vérifiant $\mathrm{c} = \mathrm{u}\mathrm{G}_n = (\mathrm{u}_A, \mathrm{u}_{A^c})\mathrm{G}_n = (\mathrm{u}_A, 0)\mathrm{G}_n$. Dans le contexte de la stéganographie, le décodage SC-S sera implémenté avec $\mathrm{u} = (\mathrm{u}_A, \mathrm{u}_{A^c}) = (\mathrm{u}_A, \mathrm{m})$, le mot stégo $\mathrm{y} = \mathrm{u}\mathrm{G}_n$ et $\mathrm{y}\mathrm{H}^{\mathrm{T}} = \mathrm{m}$. De cette manière, le mot stégo y est un mot de code du code polaire $\mathrm{m}-$coset. Ainsi, naturellement,

pour trouver le mot stégo nous pouvons appliquer le décodage polaire SC. Cette technique de décodage fournit en sortie un mot source u plutôt qu'un mot de code. Cependant, nous pouvons obtenir le mot de code simplement en appliquant la relation d'encodage $\mathrm{uG}_n = \mathrm{y}$.

Notons que la construction des codes polaires est fournie par la méthode proposée dans [75].

5.3.2.2 Calcul des métriques LR et LLR de SC pour la stéganographie

Les métriques utilisées dans le décodage SC peuvent être basées dans le domaine LR ou le domaine LLR. Dans les deux cas, d'abord les LRs des mots reçus du canal sont calculés $L_1^{(1)}(r_i) = L(r_i) = \gamma_i = \frac{w(r_i|0)}{w(r_i|1)}$. Nous allons montrer, dans le développement qui suit, comment calculer les différentes métriques LRs dans le contexte de la stéganographie.

Dans un système de communication numérique sur un canal caractérisé par ses probabilités de transition, le mot d'information u est encodé à un mot de code c qui sera transmis à travers le canal W et peut être transformé à un autre mot reçu noté r. Dans ce scénario, la tâche du décodage SC consiste à trouver le mot d'information u à partir du mot reçu r. Avec le décodage SC classique, les métriques sont calculés en utilisant les probabilités de transition $W(reçu|encodé) = W(r_i|c_i) = p_e$ si $r_i \neq c_i$ et $W(r_i|c_i) = 1 - p_e$ si $r_i = c_i$ pour un canal binaire symétrique BSC, où $p_e < 0{,}5$ est la probabilité d'erreur, r_i le $i-$ème bit reçu et c_i le $i-$ème bit du mot de code transmis. Pour un canal fiable $p_e \ll 0{,}5$ et $1 - p_e \gg p_e$. La remarque que nous pouvons en faire est que la probabilité que le c_i soit modifié est inférieure à $0{,}5$. Plus cette probabilité est grande, plus le bit c_i a des chances d'être modifié.

Nous avons démontré dans la section précédente que le principe du décodage SC peut être appliqué pour la recherche du mot stégo. Dans le contexte de la stéganographie, connaissant le mot de couverture x et le message secret m, l'expéditeur doit chercher un mot stégo y tel que $\mathrm{yH}^{\mathrm{T}} = \mathrm{m}$. En tenant compte de ce fait, nous pouvons noter les probabilités de transition par $W(couverture|stégo) = W(x_i|y_i)$ que nous pouvons interpréter comme étant la probabilité de modification ou non du pixel correspondant. Ainsi $W(x_i|x_i)$ représente la probabilité de non-modification et $W(x_i|1 - x_i)$ la probabilité de modification du pixel. La définition de $W(x_i|y_i)$ devra tenir compte des conditions que doivent vérifier les probabilités de transition dans le décodage SC et être fonction des coûts d'insertion $\rho_i, 1 \leq i \leq n$, où ρ_i est le coût de remplacer x_i par y_i.

En effet, dans la définition, il sera nécessaire d'avoir une grande valeur pour les probabilités de transition lorsque ρ_i est élevé et une petite valeur si ρ_i est faible. Pour une grande valeur de ρ_i, $W(x_i|y_i)$ doit être également grande et pour une petite valeur de ρ_i, $W(x_i|y_i)$ doit être petite. Dans le cas où tous les pixels ont la même sensibilité à la modification $\rho_i = 1$ (profil constant), nous affecterons à toutes les probabilités de transition la valeur $W(x_i|y_i) = 0{,}5$.

Lorsque nous considérons le cas de distorsion quelconque, l'idée est d'affecter des *probabilités de modification* $W(x_i|1-x_i)$ *grandes* (donc des *probabilités de non-modification* $W(x_i|x_i)$ *petites*) pour les pixels ayant des ***coûts d'insertion faibles*** et des *probabilités de modification petites* (donc des *probabilités de non-modification grandes*) pour les pixels ayant des ***coûts d'insertion grandes***. Pour y arriver, nous allons subdiviser l'ensemble des coûts d'insertion en deux sous-ensembles ρ_{set_inf} et ρ_{set_sup} des coûts faibles et des coûts élevés, respectivement, tels que $\rho_{set\_inf} = \{\rho_i \leq \rho_{mean}\}$ et $\rho_{set\_sup} = \{\rho_i > \rho_{mean}\}$, où $\rho_{mean} = \frac{1}{n}\sum_{i=1}^{n} \rho_i$ est la valeur moyenne de l'ensemble des coûts d'insertion. Dans les deux cas, nous pouvons choisir :

$$W(x_i|y_i) = \begin{cases} \frac{\rho_i}{\rho_{max}} & si \quad y_i = x_i \\ & ou \\ 1 - \frac{\rho_i}{\rho_{max}} & si \quad y_i \neq x_i \end{cases} \tag{5.12}$$

où ρ_{max} est le maximum des coûts de modification. Pour les coûts $\rho_i \in \rho_{set\_inf}$, $\frac{\rho_i}{\rho_{max}}$ est plus proche de 0 et $1 - \frac{\rho_i}{\rho_{max}}$ est plus proche de 1 et pour $\rho_i \in \rho_{set\_sup}$, $\frac{\rho_i}{\rho_{max}}$ est plus proche de 1 et $1 - \frac{\rho_i}{\rho_{max}}$ est plus proche de 0. En outre, il est facile de vérifier que $W(x_i|0) + W(x_i|1) = 1$. Ainsi,

$$W(x_i|y_i) = ([x_i = y_i])\left(\frac{\rho_i}{\rho_{max}}\right) + ([x_i \neq y_i])\left(1 - \frac{\rho_i}{\rho_{max}}\right) \tag{5.13}$$

où $[q]$ désigne l'opération logique qui est égale à 1 lorsque la relation q est vraie et 0 sinon. Donc, $[x_i = y_i] = 1 - |x_i - y_i|$ et $[x_i \neq y_i] = |x_i - y_i|$. Ainsi, nous pouvons écrire

$$W(x_i|y_i) = (1 - |x_i - y_i|)\left(\frac{\rho_i}{\rho_{max}}\right) + (|x_i - y_i|)\left(1 - \frac{\rho_i}{\rho_{max}}\right) \tag{5.14}$$

Contrairement au décodage SC classique, ici la probabilité de transition $W(x_i|y_i)$ n'est pas nécessairement la même que $W(x_j|y_j)$ lorsque $x_i = x_j$ et $y_i = y_j$ mais cela dépend de l'égalité ou non des coûts ρ_i et ρ_j.

Après avoir calculé $W(x_i|y_i)$, nous pouvons obtenir les probabilités de transition correspondant aux canaux polarisés $W_n^{(i)}$ par [25] :

$$W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}|u_i\right) = \sum_{\hat{u}_{i+1}^n \in \mathcal{X}^{n-i}} \left(\frac{1}{2^{n-1}} \prod_{i=1}^{n} W(x_i|y_i)\right) \tag{5.15}$$

Malheureusement, cette expression n'est pas pratique pour une implémentation. C'est la raison pour laquelle Arikan a défini deux formules récursives pour le calcul de ces probabilités de transition.

$$W_n^{(2i-1)}\left(x_1^n, \hat{u}_1^{2i-2}|\hat{u}_{2i-1}\right) = f(W_a, W_b) = \sum_{\hat{u}_{2i}\in\{0,1\}} \frac{1}{2} W_a \cdot W_b \tag{5.16}$$

et

$$W_n^{(2i)}\left(x_1^n, \hat{u}_1^{2i-1}|\hat{u}_{2i}\right) = g(W_a, W_b) = \frac{1}{2} W_a \cdot W_b \tag{5.17}$$

où $W_a = W_{n/2}^{(i)}\left(x_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}|\hat{u}_{2i-1} \oplus \hat{u}_{2i}\right)$, $W_b = W_{n/2}^{(i)}\left(x_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}|\hat{u}_{2i}\right)$. Soit $L_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}\right) = \frac{W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}|0\right)}{W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}|1\right)}$, alors les LRs peuvent être calculés de façon récursive en utilisant les deux fonctions de calcul f et g (3.39) $f(L_a, L_b) = L_n^{(2i-1)}\left(x_1^n, \hat{u}_1^{2i-2}\right)$ and $g_{\hat{u}_{sum}}(L_a, L_b) = L_n^{(2i)}(x_1^n, \hat{u}_1^{2i-1})$, où $L_a = L_{n/2}^{(i)}\left(x_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)$ et $L_b = L_{n/2}^{(i)}\left(x_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}\right)$:

$$\begin{cases} f(L_a, L_b) = \frac{1+L_a \cdot L_b}{L_a + L_b} \\ g(L_a, L_b, \hat{u}_{sum}) = g_{\hat{u}_{sum}}(L_a, L_b) = {L_a}^{(1-2\hat{u}_{sum})} \cdot L_b = \begin{cases} L_a \cdot L_b \ si\ \hat{u}_{sum} = 0 \\ ou \\ L_b/L_a \ si\ \hat{u}_{sum} = 1 \end{cases} \end{cases} \tag{5.18}$$

Dans ces fonctions, le calcul d'un LR de longueur n est déduit de deux LRs de longueur $n/2$. Cette récursivité continue jusqu'au dernier niveau des LRs de longueur 1, où le calcul est donné par $L_1^{(1)}(x_i) = L(x_i) = \frac{W(x_i|0)}{W(x_i|1)}$. A partir de la relation (5.13), le LR est défini par

$$L(x_i) = \frac{W(x_i|0)}{W(x_i|1)} = \frac{(1-x_i)\left(\frac{\rho_i}{\rho_{max}}\right) + (x_i)\left(1-\frac{\rho_i}{\rho_{max}}\right)}{(x_i)\left(\frac{\rho_i}{\rho_{max}}\right) + (1-x_i)\left(1-\frac{\rho_i}{\rho_{max}}\right)} \tag{5.19}$$

$$= \begin{cases} \dfrac{\rho_i}{\rho_{max}-\rho_i} & si \quad x_i = 0 \\ ou & \\ \dfrac{\rho_{max}-\rho_i}{\rho_i} & si \quad x_i = 1 \end{cases} \tag{5.20}$$

Dans le domaine logarithmique, les LLRs (Log-Likelihood Ratios) deviennent :

$$LL(x_i) = \log\big(L(x_i)\big) = \begin{cases} \log\left(\dfrac{\rho_i}{\rho_{max}-\rho_i}\right) & si \quad x_i = 0 \\ ou & \\ -\log\left(\dfrac{\rho_i}{\rho_{max}-\rho_i}\right) & si \quad x_i = 1 \end{cases} \tag{5.21}$$

$$LL(x_i) = \ (1-2x_i) \cdot \log\left(\frac{\rho_i}{\rho_{max}-\rho_i}\right) \tag{5.22}$$

Une fois qu'on donne la valeur des bits fixés et les métriques pour les choix des bits non-fixés, on peut implémenter le décodage SC des codes polaires en stéganographie. Pour le décodage SC, la version stéganographique est donnée par l'**Algorithme 5.3** :

Algorithme 5.3 SC-decoder of Polar Codes for Steganography (SC-PCS)

Entrées : objet de couverture **x**, coût d'insertion **ρ** et message **m**.

Sortie : le mot source **u**.

1: **Pour** tout $i \in \{1, 2, \cdots, n\}$

2: **si** $i \in A^c$ **alors** // les bits du message sont équivalents au frozen bits dans SC

3: Prendre le bit i du message $u_i \leftarrow m_i$;

4: **sinon si** $\left(LL_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}\right) \geq 0\right)$ **alors**

5: $u_i \leftarrow 0$;

6: **sinon**

7: $u_i \leftarrow 1$;

8: **fin si**

9: **fin Pour**

10: **retourner u** ;

L'exécution de l'**Algorithme 5.3** SC-PCS fournit un mot source qui sera transformé en stégo médium par la relation $y \leftarrow uG_n$. Ce stégo médium sera transmis au destinataire du message secret. A la réception du stégo médium, puisque la matrice génératrice G_n des codes polaires est inversible, le destinataire retrouve le mot source par $u = yG_n$. Par conséquent, $u_{A^c} = (yG_n)_{A^c} = yG^{A^c} = yH^T$, où G^{A^c} est la sous-matrice de G_n constituée des colonnes d'indices dans A^c. Finalement, le processus d'insertion donne le stégo objet de syndrome égal au message secret $m = u_{A^c}$ minimisant la fonction de distorsion additive.

5.3.3 Version du schéma basée sur le décodage SCL

Puisque SCL est une généralisation de SC, la détermination des bits fixes et le calcul des métriques sont les mêmes que pour le SC. Cet algorithme de SCL en stéganographie est le suivant :

Algorithme 5.4 SCL-decoder of Polar Codes for Steganography (SCL-PCS)

Entrées : objet de couverture **x**, coût d'insertion **ρ**, message **m** et la taille de la liste $\boldsymbol{L}$.

Sortie : objet stégo **y**.

Pour tout $i \in \{1, 2, \cdots, n\}$
  si $i \in A^c$ **alors**
    Prendre le bit du message $\hat{u}_i[l] \leftarrow m_i, \forall l \in \{1, 2, \ldots, \boldsymbol{L}\}$;
  sinon
    si Moins de $\boldsymbol{L}$ chemins sont actifs **alors**
      Dupliquer tous les chemins et continuer avec les deux valeurs possibles de u_i;
    sinon
      Trier les probabilités $\left\{W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}[l] \middle| u_i\right) : \forall l \in \{1, 2, \ldots, L\}, \forall u_i \in \{0,1\}\right\}$;
      Continuer avec les $\boldsymbol{L}$ chemins les plus probables ;
    fin si
  fin si
fin Pour
$l^* \leftarrow$ l'indice du chemin le plus probable;
retourner y $= u[l^*] \cdot G_n$;

Pour l'adaptation à l'implémentation des calculs de métriques de l'algorithme de haut niveau ci-dessus, nous allons utiliser les métriques basées LLR proposées par Balatsoukas et al. [65, Théorème 2]. Ces métriques sont définies comme suit :

$$PM_l^{(i)} \triangleq \sum_{j=1}^{i} \ln\left(1 + e^{-\left((1-2\hat{u}_j[l])\cdot L_n^{(j)}[l]\right)}\right) \tag{5.23}$$

pour le $l-$ ème chemin et le $i-$ ème niveau, où

$$L_n^{(i)}[l] = \ln \frac{W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}[l] \middle| 0\right)}{W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}[l] \middle| 1\right)}. \tag{5.24}$$

Cette définition des métriques de chemin vérifie l'équivalence suivante :

$$W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}[l] \middle| \hat{u}_i[l]\right) < W_n^{(i)}\left(x_1^n, \hat{u}_1^{i-1}[l'] \middle| \hat{u}_i[l']\right) \tag{5.25}$$

si et seulement si

$$PM_l^{(i)} > PM_{l'}^{(i)}. \tag{5.26}$$

Ces métriques de chemin peuvent être récursivement mise à jour comme suit :

$$PM_l^{(i)} = PM_l^{(i-1)} + \ln\left(1 + e^{-\left((1-2\hat{u}_i[l])\cdot L_n^{(i)}[l]\right)}\right), \tag{5.27}$$

dont une approximation est la suivante :

$$PM_l^{(i)} \approx \begin{cases} PM_l^{(i-1)} & si\ \hat{u}_i[l] = \frac{1}{2}\left(1 - sign\left(L_n^{(i)}[l]\right)\right), \\ PM_l^{(i-1)} + \left|L_n^{(i)}[l]\right|, & sinon. \end{cases} \tag{5.28}$$

Nous allons utiliser cette approximation afin de calculer les métriques de chemins pour l'implémentation du décodeur SCL stéganographique.

Notons que nous n'aurons pas besoin d'utiliser l'opération de CRC lors du décodage SCL en stéganographie. En effet, le rôle du CRC est de choisir à partir des L candidats de la liste générée, ceux qui vérifient le CRC. Lorsque deux ou plusieurs candidats vérifient le CRC, le

décodeur choisit le plus probable (celui qui a la plus grande métrique). Ainsi, le décodeur fournit en sortie le mot le plus probable avec un CRC correct. C'est pourquoi le candidat choisi finalement par le processus du CRC n'est pas nécessairement celui qui a la plus grande métrique de la liste. Cependant, en stéganographie, l'expéditeur a pour but de trouver le médium stégo y qui minimise la fonction de distorsion $D(\mathrm{x},\mathrm{y})$ et vérifiant $\mathrm{yH}^{\mathrm{T}} = \mathrm{m}$. Ce qui correspond à la sortie donnée par le décodeur SCL stéganographique lorsque la taille de liste nécessaire est choisie.

5.4 Application de l'algorithme et résultats des testes

L'implémentation du SCL-PCS suit les étapes suivantes :

- Construire le code polaire utilisé et initialiser ses paramètres ;
- Ensuite, diviser l'ensemble des coûts de modification en deux parties ρ_{set_inf} et ρ_{set_sup};
- Mettre à jour les probabilités de transition de canal calculées (en utilisant (5.20) ou (5.22)) avec les coûts qui peuvent être obtenus à partir de l'une des méthodes de définition de fonction de distorsion existantes comme HUGO, HUGO-BD, WOW, UNIWARD, ASO, HILL, MVGG ou Synch-A ;
- Appliquer le décodeur SCL stéganographique des codes polaires de l'**Algorithme 5.4** en utilisant les métriques de chemins (5.27) ;
- A la fin du processus du décodage, choisir le meilleur stégo élément de la liste.

Exemple 5.1:

Considérons le médium de couverture $\mathrm{x} = (0,1,1,1,0,0,1,0)$, le message $\mathrm{m} = (1,0,1,0)$ et stégo médium correspondant y. Les différentes étapes du processus d'insertion SC-PCS avec un code polaire de longueur $n = 8$ et de dimension $k = 4$ sont données comme suit :

- Lorsque nous utilisons la méthode de construction des codes polaires, $A = \{4,6,7,8\}$ et $A^c = \{1,2,3,5\}$. La matrice de contrôle de parité est :

$$\mathrm{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \text{ et } \mathrm{H}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (5.28).$$

- Générerons aléatoirement, un coût d'insertion $\rho = (39, 57, 8, 6,\ 54, 78, 94, 13)$, par exemple. Dans ce cas, la valeur moyenne $\rho_{mean} = 43{,}6250$, la valeur maximale $\rho_{max} = 94$ et les deux parties sont $\rho_{set\_inf} = (39, 8, 6, 13)$ et $\rho_{set\_sup} = (57, 54, 78, 94)$.

- La mise à jour des probabilités de transition dans le domaine logarithmique donne :

$LL = (-0.3438,\ -0.4321,\ 2.3749,\ 2.6856,\ 0.3001,\ 1.5841,\ -Inf,\ -1.8295)$.

- Si on applique l'algorithme SCL-PCS avec $L = 1, ou\ 2, \dots, ou\ 16$, le stégo médium obtenu est $\mathrm{y} = (0, 1,1,1,1,0,1,0)$ et correspond à celui qui minimise la fonction de distorsion.

Tableau 5-1: Distorsion totale en fonction de la taille *L* de la liste pour l'exemple 1.

Taille de liste	**Stégo médium**	**Distorsion totale $D(\mathbf{x}, \mathbf{y})$**
$L = 1$ (c-à-d SC)	(0,1,1,1, 1, 0, 1, 0)	54
$L = 2$ ou $L = 4$ ou $L = 8$ ou $L = 16$	(0,1,1,1, 1, 0, 1, 0)	54

Dans cette exemple, le stégo médium optimal peut être obtenu seulement en utilisant le décodeur SC (c'est-à-dire le décodeur SCL avec $L = 1$) et la distorsion totale est $D(\mathrm{x}, \mathrm{y}) = 54$. Les autres valeurs de *L* donnent le même stégo médium.

Exemple 5.2:

Considérons un deuxième exemple où le médium de couverture $\mathrm{x} = (1, 1,0,1,1,0,0,1)$, le message $\mathrm{m} = (1, 1,0,1)$ et le vecteur coût $\rho = (66, 18, 71, 4, 28, 5, 10, 83)$. Ainsi, la valeur moyenne $\rho_{mean} = 35{,}6250$, la valeur maximale $\rho_{max} = 83$ et les deux parties sont $\rho_{set\_inf} = (18, 4, 28, 5,10)$ et $\rho_{set\_sup} = (66, 71, 83)$. Après application de l'algorithme SCL-PCS pour $L = 1, ou\ 2, \dots, ou\ 16$, nous avons

$LL = (-1.3564,\ 1.2840,\ 1.7778,\ 2.9832,\ 0.6751,\ -2.7473,\ -1.9879, -Inf)$.

et les les résultats dans le tableau ci-après :

Tableau 5-2: Distorsion totale en fonction de la taille L de la liste pour l'exemple 2.

Taille de liste	**Stégo médium**	**Distorsion totale $D(\mathbf{x},\mathbf{y})$**
$L=1$	$(0,0,0,0,1,0,1,1)$	98
$L=2$	$(1,1,0,0,1,0,0,0)$	87
$L=4$	$(1,1,1,1,1,0,1,1)$	81
$L=8$	$(1,0,0,1,1,1,0,1)$	23
$L=16$	$(1,0,0,1,1,1,0,1)$	23

Comme le montre les résultats du tableau, avec cette exemple, le stégo médium optimal et la distorsion minimale peuvent être obtenus à partir de $L=8$.

Pour prouver les performances de notre schéma adaptatif SCL-PCS, nous évaluons également l'efficacité d'insertion $e_{eff}=m/D(\mathrm{x},\mathrm{y})$. Lors de l'application du SCL-PCS sur des images en niveaux de gris 512×512, pour augmenter les performances du SCL, nous considérons l'image dans son entièrement c'est à dire la taille de couverture $n=512\cdot512=2^{18}$. Cette valeur est suffisamment grande pour offrir des bonnes performances du SC. En outre, lorsqu'elle est combinée avec la liste fournit le stégo médium optimal. Puisque n est une puissance de 2, alors les différentes valeurs du payload relatif $\alpha=1/2,1/4,1/6,\cdots,1/20$ sont obtenues en fixant $n=2^{18}$ et en faisant varier m tel que $m/n\approx\alpha$. Ainsi, les différentes valeurs de m sont $2^{17},2^{16},2^{17}/3,2^{15},\cdots,\ 2^{16}/5$.

Notons qu'avec les méthodes de calcul de coûts, ceux élevés sont concentrés dans les zones lisses et ceux faibles dans les zones texturées de l'image de couverture. Ainsi, après avoir calculé les coûts, nous utiliserons la matrice de permutation bit-reversal B_n comme suggéré dans [79]. Cela aura pour conséquence de mélanger les pixels de l'image de couverture permettant ainsi d'accroître les chances de réussite de la recherche de la stégo image optimale.

Comme le montre la Figure 5.3, SCL-PCS fournit de meilleurs performances en termes d'efficacité d'insertion que STC mais pour $\alpha=\ [1/2,1/3]$ elles sont plus faibles. Ce résultat n'est pas surprenant et pourrait être meilleur si on se réfère aux résultats théoriques donnés dans la littérature sur les PC (capacity-achieving) lorsqu'ils sont appliqués au problème PLS comme souligné dans [19]. Cependant une amélioration pour les faibles valeurs de $\alpha=\ [1/2,1/3]$ est nécessaire.

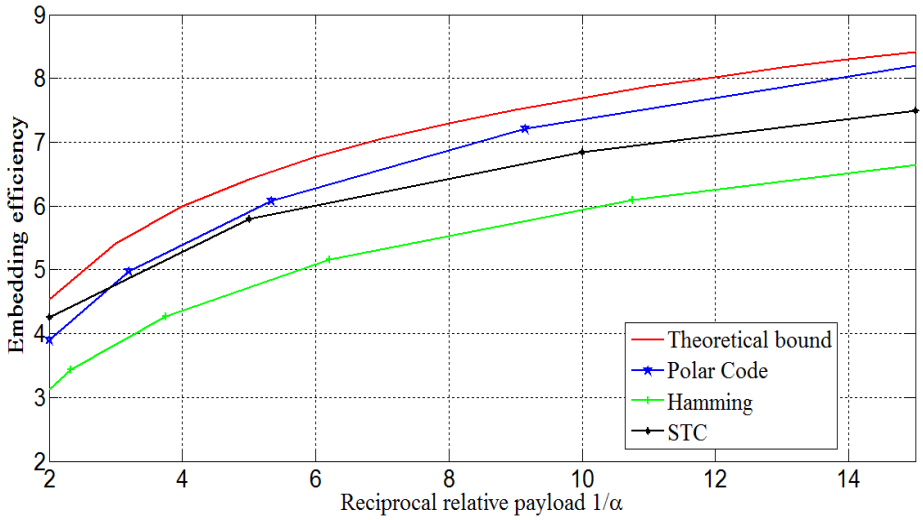

Figure 5.3 : Efficacité d'insertion de SCL-PCS pour le profile constant.

La méthode proposée permet donc de trouver le stégo médium optimal du problème stéganographique posé lorsque la taille de liste suffisante est choisie. Elle permet ainsi de minimiser la fonction de distorsion préalablement définie en utilisant l'une des méthodes de définition de coûts de modification des pixels de l'image de couverture.

Conclusion

La stéganographie moderne qui consiste à dissimuler une communication secrète tout en minimisant une certaine fonction de distorsion est utilisée par la majorité des méthodes de stéganographie actuelles pour améliorer la sécurité stéganographique. Nous avons proposé, dans ce chapitre, deux nouvelles méthodologies de stéganographie adaptative pratique ; la première basée sur le décodage par programmation linéaire adaptative (ALP) et la seconde utilise le décodage d'annulation successive en liste (SLC) des codes polaires. L'avantage de cette méthode de décodage SCL par rapport au décodage ALP est qu'elle offre toujours un stégo médium valide. Contrairement aux autres méthodes de décodage, SCL donne en sortie une liste de stégo médiums dans laquelle celui minimisant la fonction de distorsion est donné comme sortie. SCL peut être utilisé pour simuler l'insertion optimale et trouver la solution optimale du problème stéganographique en choisissant une taille de la liste $L = 2^{n-m}$. Comme le montrent les résultats des tests ainsi que les exemples d'application donnés, les deux schémas minimisent une fonction de distorsion additive arbitraire convenablement définie lors de l'insertion d'un message secret donné. Pour évaluer leur performance, nous avons comparé

leur efficacité d'insertion avec celle des codes de l'état de l'art par rapport à la limite théorique. Nous avons prouvé qu'ils fournissent de bonnes performances en termes de minimisation de fonction de distorsion qui peut être définie utilisant l'une des méthodes de définition de coûts de modification des pixels de l'image de couverture. Ceci est prouvé par l'efficacité d'insertion qui est supérieure à celle du STC. Pour les codes à papier mouillés, l'efficacité d'insertion augmente de 2,817 avec une humidité relative de $\tau = 0{,}25$ à 8,24 où $\tau = 0{,}5$. Cela montre les bonnes performances des codes polaires en termes d'efficacité d'insertion.

CONCLUSION GENERALE

L'objectif de cette thèse était de proposer les codes polaires comme outils d'insertion en stéganographie aussi bien pour la minimisation du nombre de modifications du médium de couverture (profil constant) avec la possibilité de verrouiller les zones les plus sensibles à une modification (papier mouillé) que pour une insertion adaptative sur des images numériques dans le domaine spatial voire servir de codes à papier mouillé. Nous y sommes parvenus en proposant quatre schémas de stéganographie basés sur les codes polaires : les deux premiers pour le profil constant et le cas du papier mouillé et les deux derniers pour une méthode d'insertion adaptative.

Nous avons présenté dans le premier chapitre les codes correcteurs d'erreurs à savoir les codes en blocs et les codes convolutifs. Nous avons présenté les différents concepts de base des codes, quelques exemples de codes (Hamming, BCH, RS, LDPC et les codes convolutifs) et des exemples de décodages. Le deuxième chapitre a été consacré à la présentation de la stéganographie, de la stéganalyse, des techniques de stéganographie existantes et des différents schémas proposés en utilisant les codes correcteurs d'erreurs. Dans le troisième chapitre, nous avons exposé le nouveau paradigme de codage canal, la construction, l'encodage et le décodage des codes polaires utilisés pour définir notre schéma. Le quatrième chapitre a été l'occasion pour nous de prouver l'applicabilité des codes polaires en stéganographie et de proposer un schéma adapté à la minimisation du nombre de modifications du médium de couverture ainsi qu'au cas où certains pixels ne doivent pas être modifiés (papier mouillé). Des tests sont effectués sur des images en niveaux de gris pour insérer et extraire le message secret. Pour évaluer la sécurité, nous avons tracé la courbe de PSNR qui a montré les bonnes performances en efficacité d'insertion. La recherche du stégo médium de PCS étant composée de deux étapes dont chacune nécessite un temps de calcul, nous avons réduit la complexité en proposant une autre méthode dans la deuxième partie du chapitre 4. Nous y avons appliqué une permutation des pixels de l'image de couverture avant insertion afin d'augmenter la sécurité. Afin de proposer une insertion adaptative basée sur les codes polaires, nous avons utilisé, dans le cinquième et dernier chapitre, le décodage ALP appliqué sur un nouveau graphe réduit adapté à la stéganographie ainsi que le décodage SCL. Certes, le décodage ALP a la propriété ML mais peut échouer dans la recherche du stégo médium optimal. C'est pour cette raison que nous avons proposé le décodage SCL qui offre aux codes polaires des taux d'erreurs de décodage

plus faibles que ceux des codes LDPC et Turbo-codes. Ce schéma basée sur le SCL a, d'une part l'avantage de fournir un stégo médium valide en sortie, quelques soient le médium et le message, et d'autre part, il peut s'appliquer sur une fonction de distorsion définie avec une des méthodes de définitions de coûts de modifications de pixels actuelles. Pour trouver le stégo médium optimal, qui minimise la fonction de distorsion, il suffit de choisir la taille de liste nécessaire pour le décodage SCL. Cependant, cette taille de liste peut dans certains cas être grande ce qui influe négativement sur la complexité du schéma. Cela peut être dû à la méthode de calcul des métriques utilisée lors du décodage SCL.

- Dans le but d'améliorer le schéma adaptatif basé sur le décodage SCL, nous pouvons envisager d'utiliser une autre méthode de calcul des métriques des chemins sur l'arbre de décodage qui permettrait au décodeur d'être informé dès qu'il explore un mauvais chemin (un chemin sous-optimal) afin de converger plus rapidement vers la solution optimale.

- Puisque nous avons appliqué nos schémas sur des images dans le domaine spatial, nous pouvons étudier leur applicabilité dans le domaine de la transformée (sur images JPEG). Nous prévoyons également d'appliquer la construction multicouche pour permettre au stéganographe de choisir de façon dynamique l'amplitude des modifications $(\cdots, -2, -1, 0, +1, +2, \cdots)$ en fonction du contenu de l'image.

- Vue que plusieurs architectures d'implémentation sur cartes FPGA du décodage SCL ont été proposées aussi bien pour le domaine LR que pour le domaine LLR, nous pouvons nous en appuyer pour implémenter le schéma adaptatif basé SCL sur cartes FPGA.

- Nous pensons également appliquer la stéganographie basée sur les codes polaires au cas de la stéganographie de réseau (network steganography) ou à la sécurité dans l'Internet des Objets (IdO). La stéganographie de réseau regroupe toutes les techniques de dissimulation d'informations qui peuvent être appliquées dans des réseaux de télécommunications pour permettre un échange de données cachées. On peut citer par exemple la stéganographie appliquée au service de VoIP ou ***stéganophonie***.

- Une dernière, et pas la moindre perspective, concerne la stéganalyse en utilisant les réseaux de neurones avec un classifieur universel basé sur l'algorithme d'apprentissage perceptron multicouche (PMC). Nous pouvons aussi exploiter les outils de data mining.

REFERENCES

[1]. J. Barbier, "Analyse de canaux de communication dans un contexte non coopératif : Application aux codes correcteurs d'erreurs et à la Stéganalyse", Thèse de doctorat, Ecole Polytechnique, France, Novembre 2007.

[2]. R. Crandall, "Some notes on steganography", Posted on Steganography Mailing List, 1998.

[3]. J. Bierbrauer, "On Crandall's Problem", [Online]. Available: http://www.ws.binghamton.edu/fridrich/covcodes.pdf, 1998.

[4]. A. Westfeld, "High capacity despite better steganalysis (F5–a steganographic algorithm)", In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302, Springer, Heidelberg, 2001.

[5]. M. van Dijk and F. Willems, "Embedding information in grayscale images," in Proc. 22nd Symp. Inf. Commun. Theory, Enschede, The Netherlands, May 15–16, 2001, pp. 147–154.

[6]. D. Schönfeld and A. Winkler, "A Embedding with syndrome coding based on BCH codes", In Proceedings of the 8th ACM Workshop on Multimedia and Security, pp. 214 – 223, 2006.

[7]. R. Zhang, V. Sachnev and H. J. Kim, "Fast BCH syndrome coding for steganography", S. Katzenbeisser and A.-R. Sadeghi (Eds.), IH 2009, LNCS 5806, pp. 44-58, Springer-Verlag Berlin Heiderbelg 2009.

[8]. F. Galand and C. Fontaine, "How Reed-Solomon Codes Can Improve Steganographic Schemes", In Information Hiding, Rennes, France, 2009.

[9]. J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in Proc. SPIE, Electron. Image, Security, Steganography, Watermark. Multimedia Contents IX, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 02–03.

[10]. W. Zhang and X. Wang, "Generalization of the ZZW embedding construction for steganography," IEEE Trans. Inf. Forensics Security, vol. 4, pp. 564–569, Sep. 2009.

[11]. I. Diop, S. M. Farssi, M. Chaumont, O. Khouma, et H. B. Diouf, « Utilisation des codes LDPC en stéganographie, », COmpression et REprésentation des Signaux Audiovisuels (CORESA'2012), pp. 98-104, Lille, France, 24-25 mai, 2012.

[12]. T. Filler, J. Judas, and J. Fridrich, "Minimizing Embedding Impact in Steganography Using Trellis-Coded Quantization," Proc. SPIE, Electronic Imaging, Media Forensics and Security XII, vol. 7541, San Jose, CA, January 17–21, pp. 05 1–14, 2010.

[13]. T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," In R. Böhme and R. Safavi-Naini, editors, Information Hiding, 12th International Conference, volume 6387 of Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.

[14]. S. Kouider, M. Chaumont et W. Puech, « Stéganographie Adaptative par Oracle (ASO). In COmpression et REprésentation des Signaux Audiovisuels », CORESA'12, Lille, France, 2012.

[15]. V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," In Fourth IEEE International Workshop on Information Forensics and Security, Tenerife, Spain, December 2–5, 2012.

[16]. V. Holub, J. Fridrich, and T. Denemark, "Universal distortion design for steganography in an arbitrary domain," EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop, Vol. no.1, pp. 1–13, January, 2014.

[17]. B. Li, M. Wang, J. Huang and X. Li, "A new cost function for spatial image steganography," Proceedings IEEE, International Conference on Image Processing (ICIP), Paris, France, pages 4206–4210, Oct. 2014.

[18]. V. Sedighi, R. Cogranne and J. Fridrich, "Content-Adaptive Steganography by Minimizing Statistical Detectability," *IEEE Transaction on Information Forensics and Security*, vol. 11, n° 2, pp. 221-234, February. 2016.

[19]. T. Filler, J. Judas and J. Fridrich, "Minimizing Additive Distortion in steganography Using Syndrome-Trellis Codes," IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, September 2011.

[20]. I. Diop, S. M. Farssi, K. Tall, P. A. Fall, M. L. Diouf and A. K. Diop, "Adaptive Steganography scheme based on LDPC codes," In Proc. of the IEEE 16th **I**ntern. **C**onf. on **A**dvanced **C**ommunications **T**ech. (ICACT), pp. 162-166, Pyeongchang, South Korea, February, 2014.

[21]. C. E. Shannon, "A mathematical theory of communication," Bell System Tech. J., vol. 27, pp. 379–423, 623–656, July-Oct. 1948.

[22]. C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," IRE Nat. Conv. Rec., vol. 4, pp. 142–163, 1959.

[23]. R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, 1963.

[24]. C. Berrou, A. Glavieus, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in Proc. 1993 Int. Conf. Commun., pp. 1064-1070, May 1993.

[25]. E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels", IEEE Trans. Inform. Theory, vol. IT-55, pp. 3051–3073, July 2009.

[26]. S. B. Korada, "Polar codes for channel and source coding", PhD thesis, EPFL, Lausanne, Switzerland, July 2009.

[27]. I. Tal and A. Vardy, "List decoding of polar codes," in Proc. EEE International Symposium on Information Theory Proceedings (ISIT), August 2011.

[28]. E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," IEEE Communications Letters, vol. 12, no. 6, June 2008.

[29]. N. Goela, S. B. Korada and M. Gastpar, "On LP Decoding of Polar Codes", submitted to IEEE Trans. Inform. Theory, 2010.

[30]. V. Taranalli and P. H. Siegel, "Adaptive Linear Programming Decoding of Polar Codes," IEEE Symposium on Information Theory (ISIT), pp 2982 – 2986, June-July 2014.

[31]. U. U. Fayyaz and J. R. Barry, "Low-Complexity Soft-Output Decoding of Polar Codes", IEEE Journal on Selected Areas in Communications, 32(5):958–966, 2014.

[32]. C. Berrou, « Codes et turbocodes », 1ère édition, Springer-Verlag, France, 2007.

[33]. B. Martin, « Codage, Cryptologie et applications », Collection technique et scientifique des télécommunications, 1ère édition, Presses Polytechniques et Universitaires Romandes (PPUR) - Collection : Informatique, France, 368 pages, Avril 2004.

[34]. T. J. Richardson and R. L Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes", IEEE Trans. Inform. Theory, vol. 47, pp. 638-656, February 2001.

[35]. I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, pp. 300-304, June 1960.

[36]. M. Barbier, « Décodage en liste et application à la sécurité de l'information », Thèse de doctorat, Cryptographie et sécurité, Ecole Polytechnique X, France, 2011.

[37]. A. Viterbi and J. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," IEEE Trans. Inf. Theory, vol. 20, pp. 325–332, May 1974.

[38]. V. Holub, "Content Adaptive Steganography – Design and Detection," PhD thesis, Binghamton University, May 2014.

[39]. A. D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich and T. Pevný, "Moving Steganography and Steganalysis from laboratory to Real World," In Proceedings of the IH&MMSec'13, ACM, Montpellier, France, June 17–19, 2013.

[40]. T. Filler, T. Pevný, and P. Bas, "BOSS (Break Our Steganography System)". http://www.agents.cz/boss, July 2010.

[41]. X. Luo, Q. Cheng and J. Tan, "A Lossless Data Embedding Scheme for Medical in Application of e-Diagnosis", Proceedings of the 25th Annual International Conference of the IEEE EMBS Cancun, Mexico. September 17-21, 2003.

[42]. J. Fridrich, M. Goljan, P. Lisonek and D. Soukal, "Writing on wet paper", in IEEE Trans. on Sig. Proc., Third Supplement on Secure Media, vol. 53, pp. 3923–3935, Oct. 2005.

[43]. T. Filler, "Imperfect Stegosystems–Asymptotic Laws and Near-Optimal Practical Constructions," PhD thesis, Binghamton University, NYC, USA, April 1, 2011.

[44]. T. Filler and J. Fridrich, "Gibbs construction in steganography," IEEE Transactions on Information Forensics and Security, vol. 5, n. 4, pp. 705–720, December 2010.

[45]. T. Filler and J. Fridrich, "Minimizing additive distortion functions with non-binary embedding operation in steganography," IEEE International Workshop on Information Forensics and Security, pp. 1-6, December 2010.

[46]. B. Diouf, I. Diop, S. M. Farssi and O. Khouma, "Minimizing Embedding Impact in Steganography Using Polar Codes," In Proceedings of 4rd IEEE International Conference on Multimedia Computing and Systems (ICMCS'14), Marrakesh, Morocco, April, 2014.

[47]. B. Diouf, I. Diop, S. M. Farssi and O. Khouma, "Practical Polar Coding Method to Minimize the Embedding Impact in Steganography," In Proceedings of the IEEE International Science and Information (SAI) Conference, London, United Kingdom, July, 2015.

[48]. C. Cachin, "An Information-Theoretic Model for Steganography," In Information Hiding - 2ed International Workshop, volume 1525, pages 306–318, Portland, Oregon, USA. Springer-Verlag, 1998.

[49]. A. Ker, "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, 2006.

[50]. J. Kodovsky and J. Fridrich, "Calibration revisited," in Multimedia and Security Workshop, MM&Sec'09 Proceedings of the 11th ACM multimedia, pages 63–74, Princeton, New Jersey, USA. ACM, 2009.

[51]. J. Kodovský and J. Fridrich, "Quantitative structural steganalysis of jsteg", IEEE Transactions on Information Forensics and Security, vol. 5, no. 4, pp. 681–693, 2010.

[52]. T. Pevný, J. Fridrich, and A. D. Ker, "From blind to quantitative steganalysis," In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI, volume 7254, pages 0C 1–0C 14, San Jose, CA, January 18–21, 2009.

[53]. L. Fillatre, « Contributions en Détection et Classification Statistique Paramétrique », Habilitation à diriger des recherches, HDR, Université de Technologie de Compiègne, France, 2011.

[54]. R. Cogranne, « Détection statistique d'informations cachées dans une image naturelle à partir d'un modèle physique ». Thèse de doctorat, Université de Technologie de Troyes (UTT), France, 2011.

[55]. J. Kodovský and J. Fridrich, "On completeness of feature spaces in blind steganalysis," In Proceedings of the 10th ACM Multimedia & Security Workshop, pages 123–132, Oxford, UK, September, 2008.

[56]. J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," In Proceedings of the 9th ACM Multimedia & Security Workshop, pages 3–14, Dallas, TX, September, 2007.

[57]. J. Fridrich and J. Kodovský "Rich models for steganalysis of digital images," IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 868–882, June 2011.

[58]. S. Kouider, « Insertion adaptative en stéganographie : application aux images numériques dans le domaine spatial », Thèse de doctorat, Université de Montpellier II, France, 2013.

[59]. J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media;" IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 432–444, 2012.

[60]. V. Sedighi, J. Fridrich, and R. Cogranne, "Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model," In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015, volume 9409, San Francisco, CA, February, 2015.

[61]. T. Denemark and J. Fridrich, "Improving Steganographic Security by Synchronizing the Selection Channel," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pp. 5–14, New York, NY, USA, 2015.

[62]. I. Diop, S. M. Farssi, O. Khouma, H. B. Diouf, K. Tall, K. Sylla, "New Steganographic Scheme Based of Reed-Solomon Codes," International Journal of Distributed and Parallel System (IJDPS), vol. 3, no. 2, 2012.

[63]. E. Arikan and E. Telatar, "On the rate of channel polarization," *IEEE International Symposium on Information Theory*, pp. 1493-1495, 2009.

[64]. C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware Architectures for Successive Cancellation Decoding of Polar Codes," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1665-1668, May 2011.

[65]. M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," IEEE Trans. on Comm., vol. 47, no. 5, pp. 673 –680, May. 1999.

[66]. I. Tal and A. Vardy, "List decoding of polar codes," in Proceedings IEEE Transactions on Information Theory, vol. 61, pp. 2213-2226, May, 2015.

[67]. K. Niu and K. Chen, "Stack decoding of polar codes," Electronics Letters, vol. 48, no. 12, pp. 695-696, 2012.

[68]. B. Yuan and K. K. Parhi, "Successive cancellation list polar decoder using Log-likelihood ratios," in Proc. of Asilomar Conf. on Signal, Systems and Computers, pp. 548-552, 2014.

[69]. A. B. Stimming, M. B. Parizi and A. Burg, "LLR-based successive cancellation list decoding of polar codes," in Proc. of 2014 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 3903–3907, May 2014.

[70]. J. Feldman, M. J. Wainwright and D. R. Karger, "Using linear programming to decode binary linear codes," IEEE Trans. Inform. Theory, vol. 51, no. 3, pp. 954–972, March, 2005.

[71]. P. O. Vontobel and R. Koetter, "Graphe-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," IEEE Trans. Inform. Theory, December 2005.

[72]. M. H. Taghavi and P. H. Siegel, "Adaptive methods for linear programming decoding," IEEE Trans. Inform. Theory, vol. 54, no. 12, pp. 5396–5410, December, 2008.

[73]. X. Zhang and P. H. Siegel, "Adaptive cut generation algorithm for improved linear programming decoding of binary linear codes," IEEE Trans. Inform. Theory, vol. 58, no. 10, pp. 6581–6594, October, 2012.

[74]. E. Arikan, "Systematic Polar Coding," IEEE Communications Letters, vol. 15, no. 8, August 2011.

[75]. G. Berhault, « Exploration architecturale pour le décodage de codes polaires », Thèse de doctorat, Université de Bordeaux, France, 2015.

[76]. B. Diouf, I. Diop, S. M. Farssi, K. Tall, P. A. Fall, A. K. Diop and K. Sylla, "Using of Polar Codes in Steganography," In Proc. of the 2nd Intern. Conf. on Advances in Computer Science and Engineering (CSE), vol. 42, pp. 262-266, Atlantis Press, Los Angeles, USA, July, 2013.

[77]. B. Diouf, I. Diop and S. M. Farssi, "Performances of Polar Codes in Steganographic Embedding Impact Minimization", In Proc. of the IEEE Intern. Conf. on Advanced Communications Tech. (ICACT) and CACT Transactions on Advanced Communications Technology (ICACT-TACT), vol. 5, no. 5, pp. 927–935, South Korea, 2016.

[78]. S. I. Gelfand and M. S. Pinsker, "Coding for channel with random parameters," Problems Control Information Theory, vol. 9, no. 1, pp. 19–31, 1980.

[79]. B. Diouf, I. Diop, K. Wone, S. M. Farssi, O. Khouma, M. Diouf, K. Tall, "Adaptive Linear Programming of Polar Codes to Minimize Additive Distortion in Steganography," In Proceedings of the IEEE International Science and Information (SAI) Conference, pp. 1086-1092, London, United Kingdom, 13-15 July, 2016.

[80]. S. B. Korada, E. Sasoglu and R. Urbanke, "Polar Codes: Characterization of Exponent, Bounds and Construction", submitted to IEEE Trans. Inform. Theory, January 2009.

[81]. A. Djamel, « Étude numérique comparative entre des méthodes de résolution d'un problème de transport à quatre indices avec capacités », Thèse de doctorat, École Doctorale de Mathématiques, pôle de Constantine, France, 2010.

[82]. N. Karmarkar, "A New Polynomial Time Algorithm for Linear Programming," Combinatorial, vol 4, no. 4, pp. 373–395, 1984.

[83]. J. Fridrich, "Steganography in Digital Media: Principles, Algorithms, and Application," Binghamton University, State University of New York, Cambridge University Press, 2010.

[84]. J. Fridrich and D. Soukal, "Matrix embedding for large payloads," IEEE Trans. on Information Forensics and Security, vol. 1, no. 3, 390 –395, September 2006.

[85]. T. M. Cover and J. A. Thomas, "Elements of Information Theory," New York: Wiley, 2006.

[86]. H. Farid, "Detecting hidden messages using higher-order statistical models, In Proceedings IEEE ICIP, vol. 2, pp 905-908, 2002.

[87]. P. Moulin and R. Koetter, "Data-hiding codes," Proceedings of the IEEE, Vol. 93, no. 12, pp. 2083–2126, 2005.

[88]. F. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding – a survey," Proceedings of the IEEE, Special Issue on Protection of Multimedia Content, vol. 87, no. 7, pp. 1062–1078, 1999.

[89]. B. Li, S. Tan, M. Wang, and J. Huang, "Investigation on cost assignment in spatial image steganography," IEEE TIFS, vol. 9, pp. 1264–1277, August 2014.

ANNEXE

Décodage ML

En supposant que les mots de code sont équiprobables, le décodage ML est équivalent à résoudre le problème d'optimisation $\arg\max_{c\in\mathcal{C}}(\Pr[\mathrm{r}|\mathrm{c}])$, où $\mathrm{c} \in \mathcal{C}$ est transmis, r est reçu.

$$\arg\max_{c\in C}\ (\Pr[\mathrm{r}|\mathrm{c}]) = \arg\max_{c\in C}\left(\prod_{i=1}^{n}\Pr[r_i|c_i]\right)$$

où $c_i \in \{0,1\}$ et $r_i \in \{0,1\}$ désignent, respectivement, le $i^{ème}$ symbole du mot de code candidat c et de la séquence reçu r, respectivement. Ainsi

$$= \arg\max_{c\in C}\left(\sum_{i=1}^{n}\log\Pr[r_i|c_i]\right) = \arg\min_{c\in C}\left(-\sum_{i=1}^{n}\log\Pr[\Pr[r_i|c_i]]\right)$$

Si on ajoute la constante $\sum_{i=1}^{n}\log\Pr[r_i|c_i=0]$ à l'expression de droite de l'équation précédente, on obtient

$$\begin{aligned}\arg\min_{c\in C}&\left(\sum_{i=1}^{n}-\log\Pr[r_i|c_i]+\sum_{i=1}^{n}\log\Pr[r_i|c_i=0]\right)\\ &= \arg\min_{c\in C}\left(\sum_{i=1}^{n}(\log\Pr[r_i|c_i=0]-\log\Pr[r_i|c_i])\right)\\ &= \arg\min_{c\in C}\left(\sum_{i=1}^{n}\log\frac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i]}\right)\end{aligned}$$

On peut écrire

$$\log\frac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i]} = \begin{cases}0 & si\ \ c_i=0\\ \log\dfrac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i=1]} & si\ \ c_i=1\end{cases} = \left(\log\frac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i=1]}\right)\cdot c_i$$

Soit $\gamma = (\gamma_1,\dots,\gamma_n)$ le vecteur des rapports de vraisemblance LLR (Log-Likelihood Ratios) (LLR) γ_i tels que $\gamma_i = \log\frac{\Pr[r_i|c_i=0]}{\Pr[r_i|c_i=1]}$. Alors le décodage ML devient

$$\arg\min_{c\in C}\left(\sum_{i=1}^{n}\gamma_i c_i\right) = \arg\min_{c\in C}(\langle \mathrm{c},\gamma\rangle) = \arg\min_{c\in C}\ \mathrm{c}\,\gamma^T$$