Utilisation des codes LDPC en stéganographie

Idy Diop, Sidi Mohamed Farsi, Marc Chaumont, Ousmane Khouma, Heulieu Bamar Diouf

Utilisation des codes LDPC
en stéganographie

I. DIOP[1] ; S .M FARSSI [1]; M.CHAUMONT [2,3,4] ;O. KHOUMA [1]; H. B DIOUF [1]

(1) Ecole Supérieure Polytechnique de l'Université Cheikh Anta Diop Dakar Sénégal ;

(2) Université de Nîmes, F-30021 Nîmes Cedex 1, France ;

(3) Université de Montpellier 2, UMR5506-LIRMM, F-34095 Montpellier Cedex 5, France

(4) CNRS, UMR5506-LIRMM, F-34392 Montpellier Cedex 5, France

email: marc.chaumont@lirmm.fr ; idydiop@yahoo.fr; farsism@yahoo.com

Résumé

La stéganographie est l'art de la communication secrète [1]. Depuis l'avènement de la stéganographie moderne, dans les années 2000, de nombreuses approches basées sur les codes correcteurs d'erreurs (Hamming, BCH, RS…) ont été proposés pour réduire le nombre de modifications du support de couverture tout en insérant le maximum de bits. Les travaux de Jessica Fridrich ont montré que les codes à matrices creuses approchent le mieux la limite théorique d'efficacité d'insertion. Nos travaux de recherches sont dans la continuité de ceux sur les codes à faible densité (LDGM) proposés par T. Filler en 2007. Dans cet article nous proposons une nouvelle approche avec l'utilisation des codes correcteurs LDPC[1] plutôt que les codes LDGM.[2] La complexité de notre approche est bien moindre que celle de T. Filler ce qui la rend utilisable en pratique.

Mots-clefs

Codes LDPC, Encodage, Décodage, Stéganographie.

1 Introduction

La stéganographie est l'art de la communication secrète. Elle consiste à insérer un message dans un médium anodin par exemple une image, une vidéo, un son de manière à ce que cette insertion soit statistiquement indécelable.

Une des hypothèses retenues avant *2011* consistait à dire qu'il suffisait de minimiser le nombre de modification du médium pour assurer la sécurité maximale du schéma. Cette hypothèse est remise en question depuis la compétition BOSS. Ceci dit, l'étude des codes correcteurs permettant d'insérer un message tout en minimisant le nombre de modification reste un problème intéressant.

[1]Low Density Parity Check

[2] Low Density Generator Matrix

Beaucoup de schémas stéganographiques reposant sur le principe de « matrix embedding » (il y a une utilisation détourné des codes correcteurs) ont été proposés par le passé BCH, RS…*[2][3][4]*. Ces schémas sont en général éloignés de la borne théorique d'efficacité.

Notre travail poursuit la même optique de minimisation du nombre de modification du support hôte en proposant une approche basée sur l'utilisation des codes correcteurs LDPC. Notre approche a l'avantage d'être moins complexe que l'approche LDGM *[5]* tout en étant très proche de la borne théorique d'efficacité d'insertion. Dans la section *2*, nous rappelons le principe du concept de « matrix embedding ». Dans la section *3*, nous passons en revue le principe de fonctionnement des codes LDPC. Dans la section *4*, nous expliquons notre approche. Enfin dans la section *5*, nous présentons les résultats expérimentaux.

2 Concept de « matrix embedding »

La technique de « matrix embedding » est une méthode de codage par syndrome, utilisant la théorie de codes correcteurs d'erreur. Elle a été découverte par Grandall *[6]* en *1998*, et fut implémentée pour la première fois par Westfeld *[7]* avec l'algorithme de stéganographie F5. Cette technique consiste à détourner l'utilisation classique des codes détecteurs et correcteurs d'erreur. L'objectif est de transmettre un message au sein d'une image via la modification de l'image, mais avec la contrainte de minimiser le nombre de coefficients de l'image modifiés. L'idée consiste du côté décodeur (c'est-à-dire à la réception de l'image) à calculer les syndromes de chaque bloc de coefficients à partir de la matrice de contrôle du code correcteur. Le syndrome correspond au message qui est contenu dans l'image. Toute l'astuce consiste donc, du côté codeur (c'est-à-dire à l'émission de l'image), à modifier l'image de sorte que les syndromes calculés au niveau du décodeur représentent le message et également de sorte que l'image soit la moins modifiée.

Soit $\mathbf{x} \in F_q^n$ un vecteur extrait du medium de couverture et $\mathbf{m} \in F_q^{n-k}$ notre message à cacher. Il s'agit d'insérer le message **m** dans **x** en le modifiant le moins possible. Pour cela la technique est de modifier **x** en **y** tel que :

$$H\mathbf{y} = \mathbf{m}$$

Où H représente la matrice de contrôle de parité du code.
Pour trouver **y**, nous cherchons un vecteur e ayant comme syndrome $\mathbf{m} - H\mathbf{x}$.
Nous posons alors $\mathbf{y} = \mathbf{x} + \mathbf{e}$ ce qui donne

$$H\mathbf{y} = H(\mathbf{x} + \mathbf{e}) = \mathbf{m} \Leftrightarrow H\mathbf{e} = \mathbf{m} - H\mathbf{x}$$

Le problème consiste donc à trouver ce vecteur e (code ayant comme syndrome **m**-H**x**) ayant un nombre minimal de coefficients afin de diminuer la dégradation du médium de couverture.

3 Les codes LDPC

3.1 Définition

Un code LDPC est un code dont la matrice de contrôle de parité H est de faible densité. La faible densité signifie qu'il y a plus de « 0 » que de « 1 » dans la matrice H *[8]*. Un code LDPC peut être représenté sous forme matricielle ou bien sous la forme d'un graphe bipartite (représentation de Tanner). Par exemple, la matrice suivante :

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix},$$

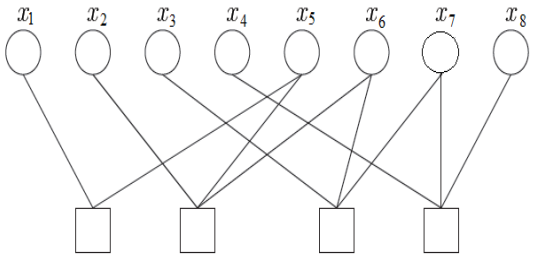

Figure 1: *Graphe bipartite d'un code LDPC*

La matrice peut être représentée par le graphe de la figure 1. Les lignes de la matrice sont représentées par des carrés et sont appelées nœuds de contrôle, les colonnes de la matrice sont représentées par des cercles et sont appelées nœuds de données et les « 1 » représentent les arrêtes du graphe.

Il y a deux familles de codes LDPC : les codes réguliers et les codes irréguliers. Les codes LDPC réguliers sont les codes dont le nombre de « 1 » par ligne et le nombre de « 1 » par colonne sont constants. Par extension, les codes LDPC irréguliers sont les codes définis par des matrices de contrôle de parité où le nombre de « 1 » par ligne ou par colonne n'est pas constant.

L'irrégularité de ces codes se spécifie à travers deux polynômes $\lambda(x)$ et $\rho(x)$.

$$\lambda(x) = \sum_{i \geq 1} \lambda_i \, x^{i-1} \qquad (1)$$

$$\rho(x) = \sum_{i \geq 2} \rho_i \, x^{i-1} \qquad (2)$$

Où λ_i (resp. ρ_i) caractérise la proportion du nombre de branches connectées aux nœuds de données (resp. aux nœuds de contrôle) de degré i par rapport au nombre total de branche. Le degré est défini comme le nombre de branches connectées à un nœud.

3.2 Encodage

Les travaux de T.J. Richardson et R.L Urbanke *[9]* ont montré que la matrice de contrôle doit subir un prétraitement avant l'opération d'encodage. L'objectif de ce prétraitement est de mettre la matrice H de taille *m×n* sous une forme presque triangulaire inférieure, comme illustré sur la Figure *2*, en utilisant uniquement des permutations de lignes ou de colonnes. Cette matrice est composée de *6* sous-matrices creuses, notées A, B, C, D, E et d'une sous-matrice triangulaire inférieure T de taille *m-g* × *m-g*. Une fois que le prétraitement de H est achevé, le principe d'encodage est basé sur la résolution du système représenté par l'équation matricielle suivante *[7]* :

$$\mathbf{c}H^T = 0 \qquad (3)$$

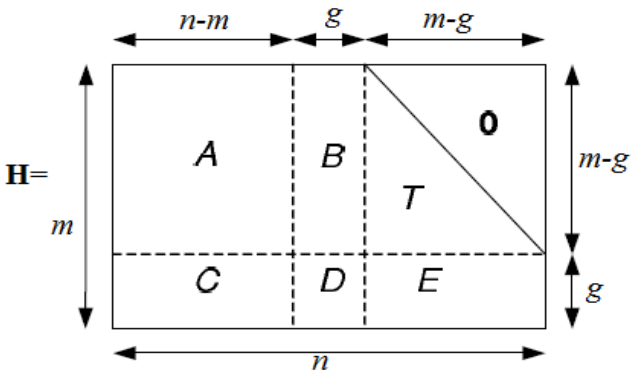

Figure 2: *Représentation sous forme pseudo-triangulaire inférieure de la matrice H*

L'algorithme de prétraitement est décrit ci-dessous de manière succincte *[10]*:
1- [Triangulation] Permutations des lignes ou des colonnes pour avoir une approximation de la matrice H sous forme triangulaire inférieure *:*

$$H = \begin{pmatrix} A & B & T \\ C & D & E \end{pmatrix}$$

avec un gap g petit si possible. Nous verrons dans la section suivante comment ceci peut être accompli efficacement.

2- [Contrôle de rang] Élimination gaussienne pour effectuer la prémultiplication par $\begin{pmatrix} \mathrm{I} & 0 \\ -\mathrm{ET}^{-1} & \mathrm{I} \end{pmatrix}$. Cette prémultiplication permet d'obtenir :

$$\begin{pmatrix} \mathrm{I} & 0 \\ -\mathrm{ET}^{-1} & \mathrm{I} \end{pmatrix} \begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{T} \\ \mathrm{C} & \mathrm{D} & \mathrm{E} \end{pmatrix} =$$

$$\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{T} \\ -\mathrm{ET}^{-1}\mathrm{B}+\mathrm{C} & -\mathrm{ET}^{-1}\mathrm{B}+\mathrm{D} & 0 \end{pmatrix}.$$

Il est nécessaire de vérifier que $-ET^{-1}B + D$ est inversible pour que le processus de prétraitement soit utilisable pour la résolution de l'équation (*3)*.

Lors de la résolution de l'équation (*3*), le mot de code recherché est décomposé en trois parties : $\mathbf{c} = (\mathbf{d}, \mathbf{r_1}, \mathbf{r_2})$, où **d** est la partie systématique (c'est-à-dire un élément de la base canonique du sous espace vectoriel de dimension *n-m* comme indiqué sur la figure *2*), où les bits de redondances recherchés sont séparés en deux vecteurs $\mathbf{r_1}$ et $\mathbf{r_2}$ de tailles respectives *g* et *m-g.* Après multiplication à droite par la matrice $\begin{pmatrix} \mathrm{I} & 0 \\ -\mathrm{ET}^{-1} & \mathrm{I} \end{pmatrix}$, l'équation (*3*) devient :

$$\mathrm{A}\mathbf{d}^{\mathrm{T}} + \mathrm{B}\mathbf{r_1}^{\mathrm{T}} + \mathrm{T}\mathbf{r_2}^{\mathrm{T}} = 0 \qquad (4)$$

$$(-\mathrm{ET}^{-1}\mathrm{A} + \mathrm{C})\mathrm{d}^{\mathrm{T}} + (-\mathrm{ET}^{-1}\mathrm{B} + \mathrm{D})\mathbf{r_1}^{\mathrm{T}} = 0 \qquad (5)$$

L'équation (*5)* permet de trouver $\mathbf{r_1}^{T}$ en inversant $\Phi = -\mathrm{ET}^{-1}\mathrm{B} + \mathrm{D}$. L'équation (*4*) permet ensuite de trouver $\mathbf{r_2}^{\boldsymbol{T}}$. Remarquons que la phase de prétraitement est une phase qui nécessite de nombreuses opérations coûteuses en temps de calcul. Par contre, toutes les opérations répétées au cours de l'encodage ont une complexité en $\mathrm{O}(n)$ excepté la multiplication de $(-\mathrm{ET}^{-1}\mathrm{A} + \mathrm{C})\mathbf{d}^{\mathrm{T}}$ par la matrice carrée $(-\Phi^{-1})$ de taille $g\ x\ g$ qui après insertion n'est plus creuse d'où une complexité en $\mathrm{O}(g^2)$. T.J. Richardson et R.L Urbanke ont aussi montré que l'on peut obtenir une valeur de g égale à une faible fraction de n : $g = \alpha n$ où α est un coefficient suffisamment faible pour que $\mathrm{O}(g^2) \ll \mathrm{O}(n)$ pour des valeurs de *n* allant jusqu'à 10^5. Ainsi, la complexité de l'approche d'encodage est de complexité O(n).

3.3 Décodage

Le décodage des codes LDPC s'effectue à partir d'algorithmes itératifs dont le plus utilisé est l'algorithme de propagation de croyance nommé BP[3]. Cet algorithme est itératif et repose sur un mécanisme classique de propagation de croyance. À chaque itération, il y a échange de messages entre les nœuds de données et les nœuds de contrôle sur un même arc du graphe bipartite. L'algorithme consiste à mettre à jour, d'abord les nœuds de données, puis les nœuds de contrôle et enfin prendre une décision de décodage du mot de code le plus probable (voir travaux de Jean-Baptiste Doré [*11*]).

[3]Belief Propagation

La mise à jour des messages $\mathrm{m_{vc}}$ issus du nœud de données *v* à l'itération *i* est calculée de la façon suivante :

$$\mathrm{m_{vc}^{i}} = v_0 + \sum_{c' \in C_v \setminus c} \mathrm{m_{c'v}^{i-1}} \qquad (6)$$

Où v_0 représente le log-rapport de vraisemblance issue de l'observation y_v en sortie du canal :

$$v_0 = \ln \frac{\Pr(\mathrm{y_v}|\mathrm{v}=0)}{\Pr(\mathrm{y_v}|\mathrm{v}=1)}, \qquad (7)$$

et où C_v représente l'ensemble des nœuds de contrôle connectés au nœud de données *v*. À la première itération, les messages provenant des nœuds de contrôle sont nuls.

La mise à jour des messages $\mathbf{m_{cv}}$ issus du nœud de contrôle *c* à l'itération *i* est calculée de la façon suivante :

$$\mathrm{m_{cv}^{i}} = 2\tanh^{-1}\left(\prod_{\mathrm{v}' \in \mathrm{C_c} \setminus \mathrm{v}} \tanh\left(\frac{\mathrm{m_{v'c}^{i-1}}}{2}\right)\right) \qquad (8)$$

Où C_c représente l'ensemble des nœuds de données connectés au nœud de contrôle *c*.

4 Principe du schéma basé sur les codes LDPC

4.1 Minimisation de l'impact d'insertion

Lorsque nous souhaitons insérer un message en utilisant un schéma de stéganographie par « matrix embedding », nous prenons une représentation du support hôte sous forme d'un vecteur $\mathbf{x} \in \mathrm{F}_2^n$ (par exemple les LSB d'une image en niveau de gris). Ce vecteur hôte est alors modifié en un vecteur stégo $\mathbf{y} \in \mathrm{F}_2^n$. Le message binaire est représenté par un vecteur binaire $\mathbf{m} \in \mathrm{F}_2^m$ avec $m < n$.

Dans le cas des codes LDPC, la matrice de contrôle de parité H est utilisée pour l'encodage et également pour le décodage. Soit $C(\mathbf{m}) = \{\mathbf{v} \in \mathrm{F}_2^n | \mathrm{H}\mathbf{v} = \mathbf{m}\}$ le coset[4] correspondant au syndrome $\mathbf{m} \in \mathrm{F}_2^m$ (avec **m** le message secret). L'insertion (Emb) et l'extraction (Ext) du message peuvent être modélisées par :

$$\mathbf{y} = \mathrm{Emb}(\mathbf{x}, \mathbf{m}) \triangleq \arg \min_{\mathrm{v} \in \mathrm{C(m)}} \|\mathbf{x} - \mathbf{v}\|, \qquad (9)$$

$$\mathrm{Ext}(\mathbf{y}) = \mathrm{H}\mathbf{y} = \mathbf{m}, \qquad (10)$$

avec $\| \ \|$ le poids de Hamming, c'est-à-dire le nombre de modification du vecteur.

L'approche par « matrix embedding » consiste à déterminer le vecteur **u** membre du coset tel que :

$$\min_{\mathbf{v} \in C(\mathrm{m})} \|\mathbf{x} - \mathbf{v}\| = \min_{\mathbf{c} \in C} \|\mathbf{x} - \mathbf{u} - \mathbf{c}\|, \qquad (11)$$

où c est un mot de code. Toute la problématique du « matrice embedding » est de déterminer le vecteur **y = v = u+c.** En effet, le vecteur **u+c** fait partie du Coset *C(**m**)*.

[4]L'ensemble des mots de code qui ont le même syndrome

À la réception de la stego image, le récepteur pourra effectuer le calcul H.**y** et ainsi retrouver le message **m.**

4.2 Le schéma proposé

T. Filler s'est intéressé dans ses travaux à la conception des schémas stéganographiques optimaux.
En effet, il a montré que le problème de minimisation de l'impact statistique d'insertion en stéganographie est équivalent à la quantification binaire généralement rencontrée dans les normes de compression.
À ce titre, il a proposé un algorithme (Belief Propagation) permettant la résolution de quantification binaire en stéganographie.
En pratique, la validité de cette approche a été testée grâce à la génération des mots codes obtenus avec la matrice génératrice LDGM.
En étudiant la complexité et la condition de convergence algorithmique du problème de la quantification binaire en stéganographie, il a montré les bonnes dispositions des codes à densité faibles pour la construction de schémas de stéganographie optimaux.
Nous nous inspirons non seulement de Filler mais aussi du concept des « matrix embedding » pour proposer l'utilisation des codes LDPC.
Dans ses travaux, Filler a fait appel à deux matrices carrés qui permettent de traiter la matrice génératrice sous forme triangulaire. Nous allons montrer que ce traitement est plus simple avec les codes LDPC.
Notre travail est donc un prolongement du travail de T. Filler, pour ce qui nous concerne, l'utilisation des codes à densité faibles en stéganographie. Nous proposons une matérialisation pratique de son approche en simplifiant certains aspects de son approche plus théorique.

Nous allons chercher à minimiser le nombre de pixels à changer pour insérer un message dans une image de couverture. Nous prenons comme représentant de l'image couverture le vecteur binaire **x** composé des LSB (Least Signifiants Bits) des pixels choisis pour insérer le message.

L'insertion du message dans **x** est alors réalisée comme cela :

1. [Traitement de la matrice de contrôle] Le prétraitement de la matrice de contrôle de parité est réalisé comme nous l'avons expliqué en Section *3.2*. Notons que la méthode proposée par T. Filler est différente pour cette phase de prétraitement.
2. [Calcul d'un vecteur **u**] Nous calculons le vecteur u membre du coset selon :
$$\mathbf{u} = \mathrm{P}^{\mathrm{T}}.\mathbf{m} \qquad (12)$$
où P^{T} est la transposée de la matrice H' obtenue via l'algorithme de prétraitement décrit dans la Section *3.2*. Ce qui diffère de l'approche de Filler. En effet pour déterminer **u**, T. Filler a fait un traitement au niveau de la matrice génératrice G du code LDGM avec l'introduction de deux matrices dont il suppose l'existence : $\mathrm{P_r}$ matrice qui permute les lignes, $\mathrm{P_c}$ la matrice qui permute les colonnes.
Filler a abouti à la relation suivante :
$$(\mathrm{P_r G P_c})^{\mathrm{T}} = \mathrm{G'}^{\mathrm{T}} = \begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{T} \\ \mathrm{C} & \mathrm{D} & \mathrm{E} \end{pmatrix}$$
où $\begin{pmatrix} \mathrm{A} & \mathrm{B} & \mathrm{T} \\ \mathrm{C} & \mathrm{D} & \mathrm{E} \end{pmatrix}$ est une matrice presque triangulaire comme illustré sur le figure 2.
Pour faire ce travail, il s'est inspiré des travaux de Richardson et Urbanke, finalement il a posé :
$\mathbf{u}=\mathrm{P_r}^{-1}.(\mathbf{m},\underline{\mathbf{0}})$ ou $\underline{\mathbf{0}}$ est un vecteur qu' il concatène au message **m** pour que la multiplication matricielle soit possible.
Nous voyons que cette approche est très complexe pour la détermination du vecteur **u**.
Raison pour laquelle nous proposons une méthode beaucoup plus simple avec les codes LDPC.
3. [Calcul du vecteur **c**] Calcul de **c = (d, r1, r2**), avec **d** la partie systématique (c'est-à-dire un élément de la base canonique du sous espace vectoriel de dimension *n-m* comme indiqué sur figure *2*). Les vecteurs $\mathbf{r_1}$ et $\mathbf{r_2}$ sont déterminés comme expliqué dans la Section *2.2*.
4. [détermination du vecteur de modification **r**] Nous déterminons le vecteur **r** qui approche $\mathbf{x} - \mathbf{u} - \mathbf{c}$ comme décrit dans l'équation *(13)* en exécutant l'algorithme BP qui prend en entrée **x-u-c** et retourne le mot de code **r**. Notons que T. Filler utilise également l'algorithme BP avec la même approche pour obtenir le vecteur de modification **r**. Une fois que le vecteur de modification **r** est obtenue, nous calculons le vecteur stego **y = r + u**.

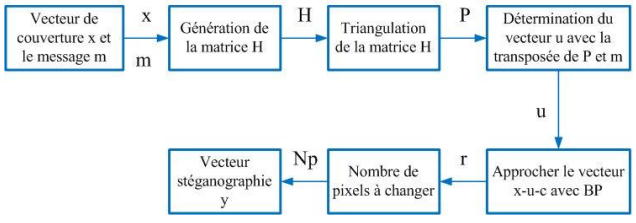

Figure 3: *Schéma basé sur les codes LDPC*

La Figure *3* donne une représentation des différentes étapes du processus d'insertion d'un message dans un vecteur de couverture en utilisant les codes LDPC

5 Présentation et analyses des résultats

La construction d'un code LDPC consiste à remplir avec des valeurs non nulles la matrice de parité. Pour optimiser la construction des codes LDPC, nous utilisons l'approche en trois étapes de Claude Berrou *(*optimisation des profils d'irrégularité, optimisation de la taille des cycles, sélection du code par la méthode impulsionnelle).

La Figure *4* donne une représentation d'une matrice de contrôle de parité d'un code LDPC en respectant ces principes.

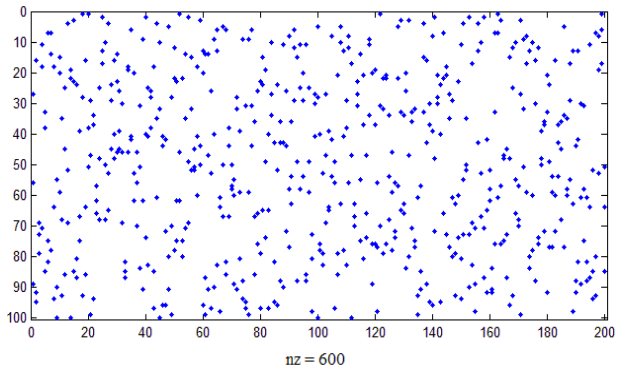

Figure 4: *Exemple de matrice de parité LDPC*

Sur la Figure *4*, la matrice est de taille *100 x 200*, c'est-à-dire *100* lignes et *200* colonnes. Les valeurs non nulles (nz=none zero dans la figure) sont représentées par des « points en bleu» et sont au nombre de *600*. Le taux vaut $\tau = 600/(100 \times 200) = 0.03$. Pour cet exemple, *3*% des éléments de la matrice sont non nuls.

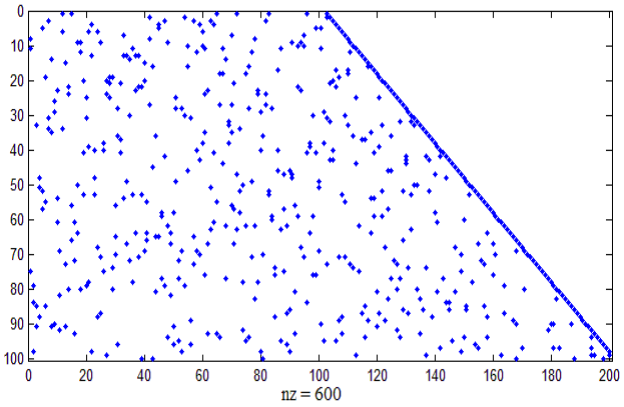

Figure 5 : *Représentation sous forme presque triangulaire de la matrice de parité*

La Figure *5* donne une représentation de la matrice de contrôle de parité en sortie de l'algorithme de prétraitement décrit en Section *3.2*.

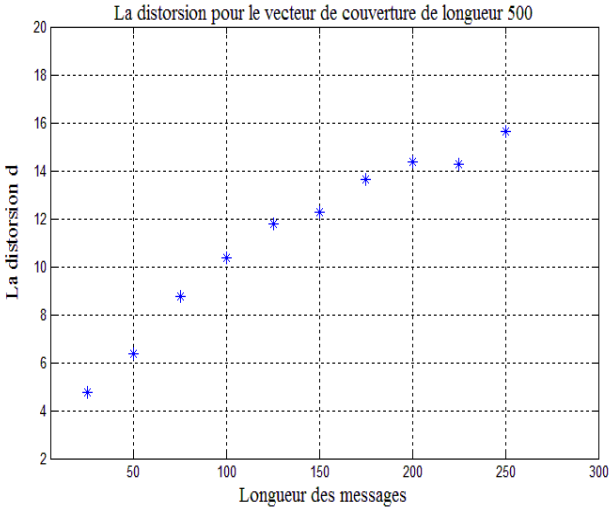

Figure 6*: Distorsion pour les différents messages insérés dans un vecteur de longueur 500*

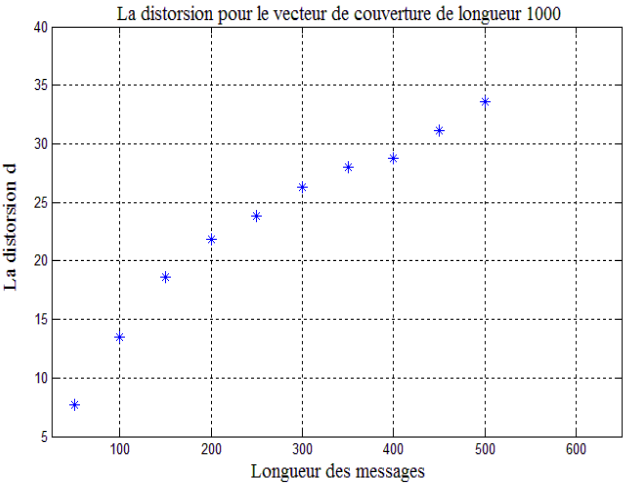

Figure 7: *Distorsion pour les différents messages insérés dans un vecteur de longueur 1000*

Les Figure *6* et Figure *7* donnent une représentation des différentes valeurs de la distorsion (nombre de pixels modifiés) dans les vecteurs de couverture de longueurs *500* bits et *1000* bits respectivement en fonction de la longueur du message. Les messages et les vecteurs de couverture sont générés de façon aléatoire. Pour le vecteur de *500* bits, les messages insérés ont des tailles allant de *25* à *250*. La borne supérieure est égale à *250* bits parce que la taille du message doit satisfaire l'inégalité suivante : $2m \leq n$ avec m et n les tailles respectives du message et du vecteur de couverture. Pour le vecteur de *1000* bits, les messages insérés ont des tailles allant de *50* à *500*. Notons que les vecteurs de couverture peuvent être

considérés comme les LSB des pixels d'une image de couverture. Remarquons aussi que la distorsion est pratiquement linéaire par rapport à la taille du message

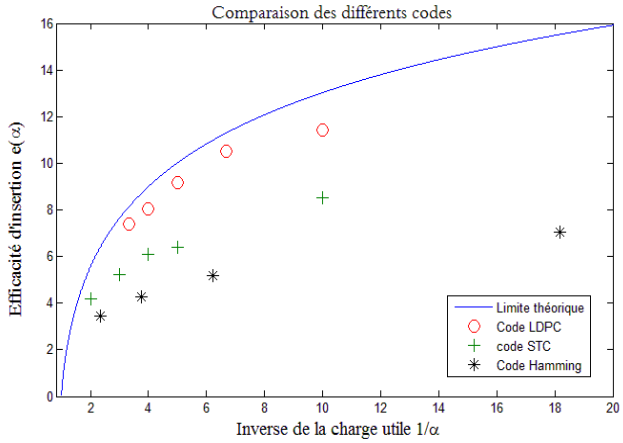

Figure 8*: Comparaison de performances entre les codes*

La limite théorique supérieure de l'efficacité d'insertion est *[11]* :

$$e \leq \frac{\alpha}{H^{-1}(\alpha)}$$

avec $\alpha = \frac{m}{n}$ (le payload) et H est la fonction d'entropie qui est définie par :

$H(x) = -x\log_2 x - (1-x)\log_2(1-x)$.

La Figure 8 présente une comparaison des performances d'insertion des codes LDPC, STC et Hamming ainsi que le tracé de la borne théorique d'efficacité d'insertion. Les valeurs de l'efficacité d'insertion pour les codes LDPC sont obtenues par insertion de *10* messages dans un vecteur de taille $n = 1000$. Les valeurs d'efficacité d'insertion pour le code STC sont obtenues en exécutant le code source de Tomáš Filler et Jessica Fridrich (avec un réglage de coût de détectabilité constant). Les valeurs d'efficacité des codes de Hamming sont déterminées par la formule suivante $e = \frac{p}{1-2^{-p}}$ avec *p* un entier naturel. La figure 8 confirme l'hypothèse de Jessica Fridrich selon laquelle les codes à matrices creuses approchent de très prêt la limite théorique de l'efficacité d'insertion.

Les codes LDPC sont donc de bons candidats pour la stéganographie par minimisation du nombre de pixels modifiés.

6 Conclusion

Dans cet article, nous avons rappelé le principe de fonctionnement des codes LDPC. Nous avons ensuite présenté une approche de « matrix embedding » s'inspirant de l'approche plus théorique de T. Filler qui a montré en s'appuyant sur la « Low Density Generatrice Matrix » (matrice génératrice) que le problème de minimisation de l'impact statistique d'insertion en stéganographie est équivalent à la quantification binaire.

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 la réduction de la complexité grâce à une étape de prétraitement. Les résultats obtenus confirment que les codes LDPC binaires permettent d'insérer un message en minimisant le nombre de pixels modifiés. En comparaison avec les codes STC binaires *[12]* (avec une utilisation d'un coût de détectabilité constant), les codes LDPC présentent une performance, car ils s'approchent bien plus de la limite théorique d'efficacité d'insertion.

Nos recherches futures s'orienteront après vers l'utilisation de codes photographes *[13]* issus des codes LDPC et qui ont une configuration beaucoup plus creuse. Par ailleurs, nous envisageons aussi d'étudier des codes LDPC non binaires. Enfin, comme nous l'avons souligné en introduction, actuellement les schémas de stéganographie les plus sûrs prennent en compte la détectabilité de chaque pixel lors de l'insertion. Il est donc nécessaire d'intégrer cette carte de détectabilité dans la création de nouveaux codes (le code STC est un exemple de code prenant en compte la carte de détectabilité) *[14]* et procéder à la stéganalyse de ce schéma pour nous prononcer de façon plus approfondie sur la performance.

Références

[1] Jessica Fridrich, *Steganography In digital media principles, Algorithms, and Application*, Binghamton University, State University of New York, Cambridge University Press, 2010.

[2] Rongyue Zhang, Vasiliy Sachnev, Hyoung Joong 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.

[3] Vasiliy Sachnev, Hyoung Joong Kim, Rongyue Zhang, *Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding*; MM& Sec '09, Princeton, New Jersey, USA, September 2009.

[4] F. Galand C. Fontaine. How can Reed-Solomon codes improve Steganographic schemes. In Information Hidding, Rennes, France, 2009.

[5] Tomas Filler, *Minimizing Embedding Impact in Steganography Using Low Density Codes*, Thesis, Department of Electrical and Computer Engineering, SUNY Binghamton, USA, 2006/2007.

[6] R. Crandall. *Some notes on steganography.* Posted on Steganography Mailing List (1998).

[7] 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).

[8] Claude Berrou, *Codes et turbocodes* ; 1e édition, Springer - Verlag, France, 2007

[9] T.J. Richardson and R.L Urbanke, « Efficient Encoding of Low-DensityParity-Check Codes », *IEEE Trans. Inform. Theory*, vol. 47, pp. 638-656, February 2001.

[10] Jean-Baptiste Doré, « Optimisation conjointe de codes LDPC et de leurs architectures de décodage et mise en œuvre sur FPGA », Thèse pour obtenir le grade de Docteur à l'INSA de Rennes, Spécialité : Electronique, Soutenue le 26 Octobre 2007.

[11] Tomáš Filler, Jan Judasand Jessica Fridrich, *Minimizing Additive Distortion in steganography Using Syndrome-Trellis Codes*, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 6, NO. 3, SEPTEMBER 2011.

[12] Tomáš Filler, Jan Judas et Jessica Fridrich, *Minimizing Embedding Impact in Steganography using Trellis-Coded Quantization*, Department of Electrical and Computer Engineering SUNY Binghamton, Binghamton, NY 13902-6000, USA 2010.

[13] Todd K Moom, *ERROR correction CODING Mathematical Methods and Algorithms*, Utah State University, Copyright ©2005 by John Wiley &Sons, Inc.

[14] Tomáš Pevnỳ, Tomáš Filler and Patrick Bas, *Using High-Dimensional Image Models to Perform Highly Undetectable Steganography*, Czech Technical University in Prague, Czech Republic; State University of New York in Binghamton, NY, USA; CNRS-LAGIS, Lille, France, 2010.