Stéganographie : Technique BPCS

Introduction

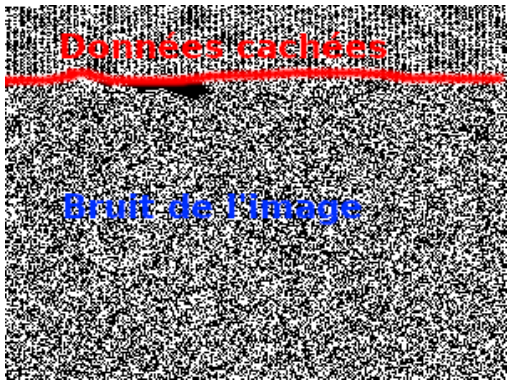
Contairement aux méthode de chiffrement de l'information, la stéganographie ne cherche pas à rendre une information illisible mais à la cacher. En effet, quelle que soit la méthode de chiffrement utilisée, l'interception d'un message chiffré par un attaquant lui donne tout de même quelques informations, notamment le fait que des informations aient été échangées.

La stéganographie vise à pallier à ce problème en cachant l'information importante (à cacher) au sein d'une autre information d'apparence anodine.
Ainsi, la stéganographie peut par exemple être utilisée pour envoyer des messages secrets sans éveiller les soupçons, ou pour un usage plus banal insérer des métadonnées dans des fichiers.

Limites d'une technique basique: LSB

Une technique simple (et vue en cours) pour cacher de l'information dans une image est d'utiliser les bits de poids faible d'une image (**L**east **S**ignificant **B**its).
Bien que simple à implémenter, cette technique présente plusieurs défauts, par exemple :

- Seuls 10 à 15 % des bits de l'images porteuse peuvent être utilisés pour cacher de l'information. Utiliser des bits de poids trop fort dégraderait fortement l'image originale.
- Cette méthode est très peu résistante à la stéganalyse. Une simple exploration des plans de bits permet de voir à l'oeil nu la présence de données cachées. Sur l'image ci-dessous, on peut voir que les données cachées sont visibles sur le plan représentant le bit de poids faible du canal rouge.

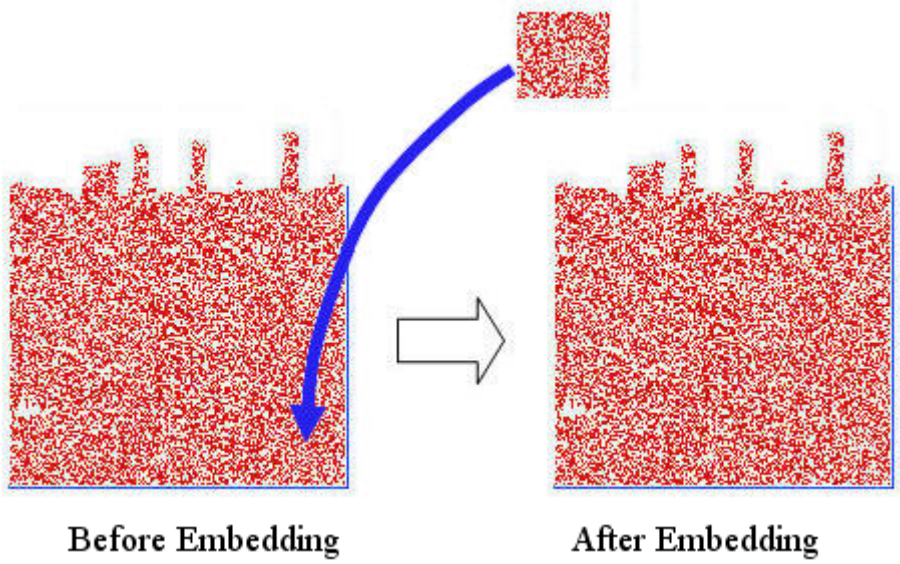


Nous cherchons donc une méthode nous permettant de stocker plus d'informations dans une image, c'est à dire utiliser plus de bits, sans que cela soit décelable par l'oeil humain.
De plus, les données cachées ne devront pas être trop facilement identifiables par visualisation des plans de bits.

Présentation de la méthode BPCS

La méthode BPCS (**B**it-**P**lane **C**omplexity **S**egmentation) aborde le problème de la dissimulation d'information dans une image d'une manière différente de la méthode LSB.
Au lieu de s'appuyer sur les propriétés de la représentation binaire des couleurs, BPCS va plutôt exploiter les propriétés de l'oeil humain.

En effet, notre oeil n'est pas vraiment équipé pour différentier des images constituées uniquement de "bruit", c'est à dire des images complexes. Pour le dire plus simplement, il est beaucoup plus facile de différentier un carré et un cercle qu'un bruit avec un autre bruit.



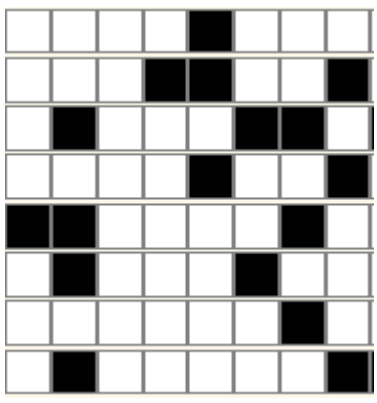
Pour mesurer cet effet, on introduit une mesure de la complexité d'une image (dans notre cas, il s'agit de la complexité d'un plan de bits). On commencer par calculer la **longueur de la frontière noir-blanc** notée **k**, qui est définie comme la somme de tous les changements entre noir et blanc sur toutes les lignes et sur toutes les colonnes (k est donc maximal pour un damier par exemple).

On définit la complexité globale alpha d'une image comme suit :

$$\alpha = \frac{k}{T}$$

Avec **T** la plus grande longueur possible de la frontière noir-blanc. On a donc alpha compris entre 0 et 1.

Le concept de base de BPCS va donc être de remplacer des portions complexes de l'image originale, c'est-à-dire des portions semblables à du bruit, avec nos données arbitraires que l'on cherche à cacher. Pour ce faire, on va commencer par diviser chaque plan de bits de l'image originale en zones 8 x 8 pixels.



Pour chacune de ces zones locales, on va calculer une complexité locale alpha comme décrit ci-dessus.
Si cette complexité est supérieure à un certain seuil, alors cette zone est considérée comme **complexe** et on peut donc la remplacer par une autre zone **complexe**.

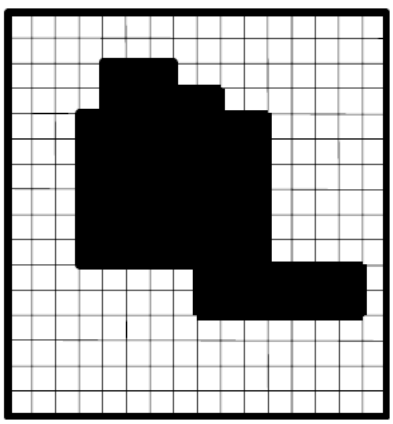
Quelles sont donc ces zones par lesquelles on les remplace ? Il s'agit de "zones 8 x 8" construites à partir des bits de l'information à cacher, groupés par paquets de 8 octets.

On se heurte donc à un problème : il est possible d'ignorer les zones simples de l'image originale afin de ne pas en remplacer, mais il faut bien écrire tous les bits contenus dans l'information à cacher.
Or, les zones 8 x 8 artificielles construites à partir des données à cacher n'auront pas toutes une complexité supérieure au seuil et seront donc peut-être simples !

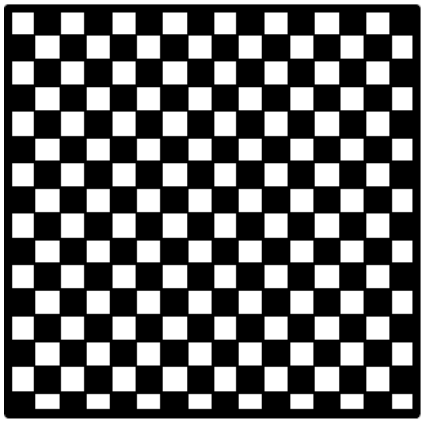
Dans ce cas, on ne peut pas remplacer une zone complexe par une zone simple. Il va donc falloir transformer les zones simples de l'information à cacher en zones complexes par un processus dit de **conjugaison**.

Le processus de conjugaison

Soit **P** la zone 8 x 8 simple que l'on veut conjuguer pour en faire une zone complexe.
Soit **Wc** une zone 8 x 8 en damier, avec un bit blanc en haut à gauche.

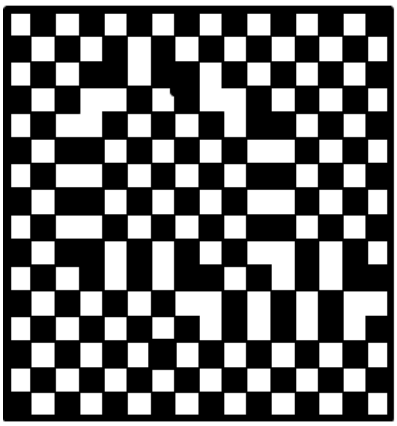


P



Wc

On définit le conjugué **P*** de **P** comme étant le XOR bit à bit entre **P** et **Wc**.



P*

On a les propriétés suivantes pour P* :

- $P^\star = P \oplus Wc$
- $(P^\star)^\star = P$

Et surtout :

- $\alpha\left(P^\star\right) = 1 - \alpha\left(P\right)$

Autrement dit, si **P** est simple, alors **P*** est complexe.
De plus, la propriété 2 nous permet de retrouver nos données simplement en réappliquant l'opération.

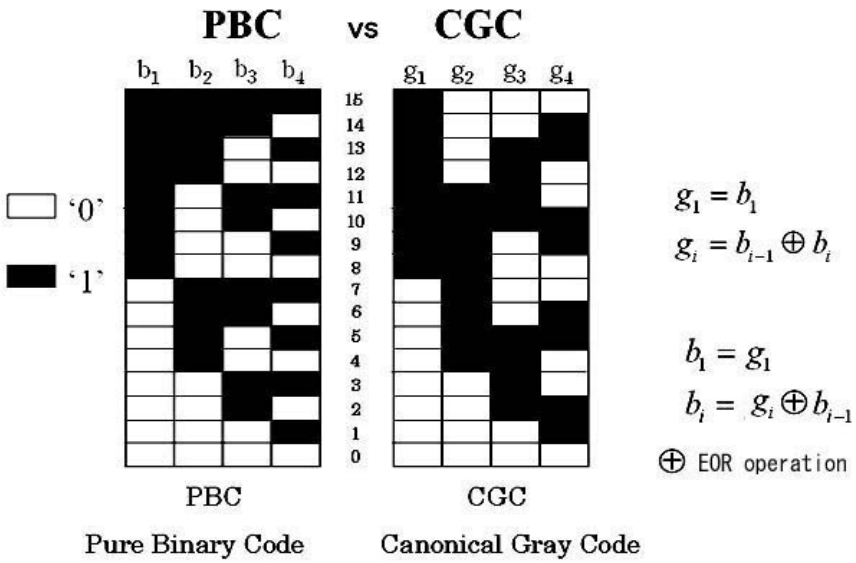
Il reste bien évidemment un dernier problème qui surviendra lors du processus d'extraction des données pour le réceptionnaire du message: Comment faire la différence entre les blocs ayant été conjugués avant leur écriture et les autres ?

Pour résoudre ce problème, on inclut également la carte des conjugaisons dans l'image. Typiquement une suite de bits indiquant si chaque bloc a été conjugué ou non. (1 si oui, 0 si non).
La technique BPCS ne décrit pas sous quelle forme et de quelle manière inclure cette carte dans l'image source, c'est pourquoi il y a différentes **implémentations** de BPCS.

Détails importants

Les couleurs d'une image sont la plupart du temps représentées par un nombre binaire 8-bit. Historiquement, le codage de ce nombre binaire se fait avec la convention PBC (**P**ure **B**inary **C**ode), c'est-à-dire que le nombre décimal 128 sera représenté par la suite de bits **10000000** et 127 par **01111111**.

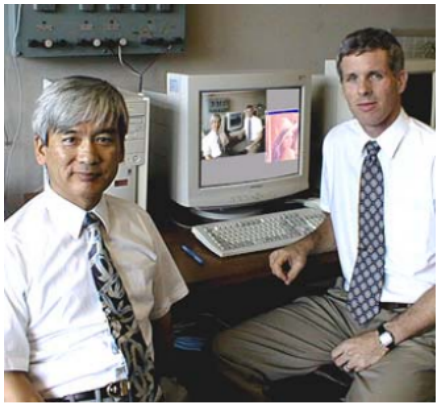
Pour implémenter la méthode BPCS, on préférera toutefois un autre codage binaire nommé CGC (**C**anonical **G**ray **C**ode), obtenu par des XOR deux-à-deux des bits de la représentation PBC.



Les avantages d'utiliser CGC au lieu de PBC sont multiples en stéganographie.
En effet, si on se place sur un plan de bits quelquonque, par example le quatrième, et que l'on change un bit, la valeur du nombre représenté changera toujours de 8 en utilisant le codage PBC.
Avec le codage CBC en revanche, changer un bit à la position 4 ne changera pas toujours le nombre par la

même valeur. Cela permet d'éviter certains artéfacts dans l'image obtenue (avec le bonus de rendre un peu plus difficile la stéganalyse).

De plus, le codage CBC permet d'éviter l'effet dit des "Falaises de Hamming", c'est à dire qu'un changement de bit de poids fort dans un codage PBC provoque un gros changement de couleur. Prenons un exemple concret :



On peut constater que le mur en arrière plan de l'image est dans une couleur relativement pleine.

Analysons maintenant les plans de bits en codage PBC :

(A) PBC red plane 3 (B) PBC red plane 4 (C) PBC red plane 5

On peut observer deux choses ici:

- De manière assez intuitive, plus le plan de bit a un poids faible, plus le nombre de régions complexes où l'on peut insérer des données est important.
- Le mur de couleur pleine en arrière-plan présente tout de même des régions complexes même dans des plans de bits relativement élevés.
  Changer un bit sur ces plans avec un codage PBC reviendrait à changer drastiquement la couleur du pixel et ainsi d'une part dégrader la qualité de l'image et d'autre part risquer qu'un observateur remarque la modification de l'image.

Regardons maintenant les mêmes plans en codage CGC :

(D) CGC red plane 3

(E) CGC red plane 4

(F) CGC red plane 5

Ici, on peut constater plusieurs choses :

- Le mur en arrière plan est bien composé de zones simples dans les plans de bits de poids fort, ainsi on s'assure de ne pas changer trop drastiquement les couleurs ici (effet qui est de toutes façons un peu atténué avec ce codage).
- La quantité de zones complexes a diminué sur tous les plans de bits. Cela a pour conséquence de diminuer la taille maximale de l'information que l'on pourra cacher dans l'image.
  Ce n'est en fait pas trop grave puisque BPCS est déjà un algorithme permettant des volumes de données embarquées très importants par rapport à d'autres algorithmes. C'est donc un prix que l'on est prêt à payer pour obtenir des images convainquantes.

Déroulement de l'algorithme

Voici un résumé de la suite d'opérations nécéssaire au processus d'écriture d'un message caché dans une image avec la méthode BPCS.

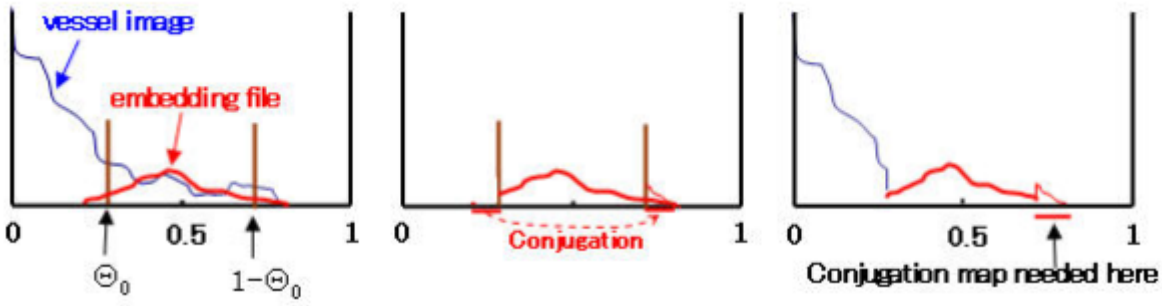
1. Convertir le codage de chaque canal de couleur de l'image source en CGC.
2. Séparer les plans de bits CGC de l'image source et les découper en blocs 8 x 8, faire de même pour l'information à cacher.
3. Calculer la complexité alpha pour chacun de ces blocs.
4. Convertir les blocs simples de l'information à cacher en blocs complexes par une opération de conjugaison. Noter les blocs conjugués dans la carte de conjugaison.
5. Remplacer les blocs complexes de l'image source successivement avec les blocs d'information à cacher.
6. Insérer la carte des conjugaisons dans l'image source.
7. Reconvertir le codage de l'image source en PBC.

Conclusion

Nous avons donc présenté une technique permettant de cacher des informations au sein d'une image. Comparé à la technique LSB, BPCS permet (selon le seuil choisi) d'utiliser jusqu'à 50% de l'image source pour

cacher les données.
Ces résultats impressionnants sont toutefois à relativiser, puisque la méthode BPCS comporte quelques désavantages:

- La portion de l'image de l'image pouvant être utilisée pour cacher de l'information est variable, ce qui rend difficile de prévoir de combien de place on dispose à l'avance.
- Toutes les images ne sont pas adaptées pour servir de support. En effet, les "bonnes" images sont les images venant par exemple de caméras numériques car elles contiennent beaucoup de bruit et donc de région contenant peu d'information. En revanche, une image plus simple comme un carré noir sur fond blanc par exemple sera un très mauvais support.
- La technique n'est pas résistante à la stéganalyse à priori. En effet, il est théoriquement possible en traçant l'histogramme de la complexité des régions d'une image modifiée de distinguer des irrégularités.



Auteurs

Eric Jean-François
Tristan Delapierre