Applications des codes de Reed-Solomon en Stéganographie

17044 Hamdoun Anas

2021

Plan

Stéganographie

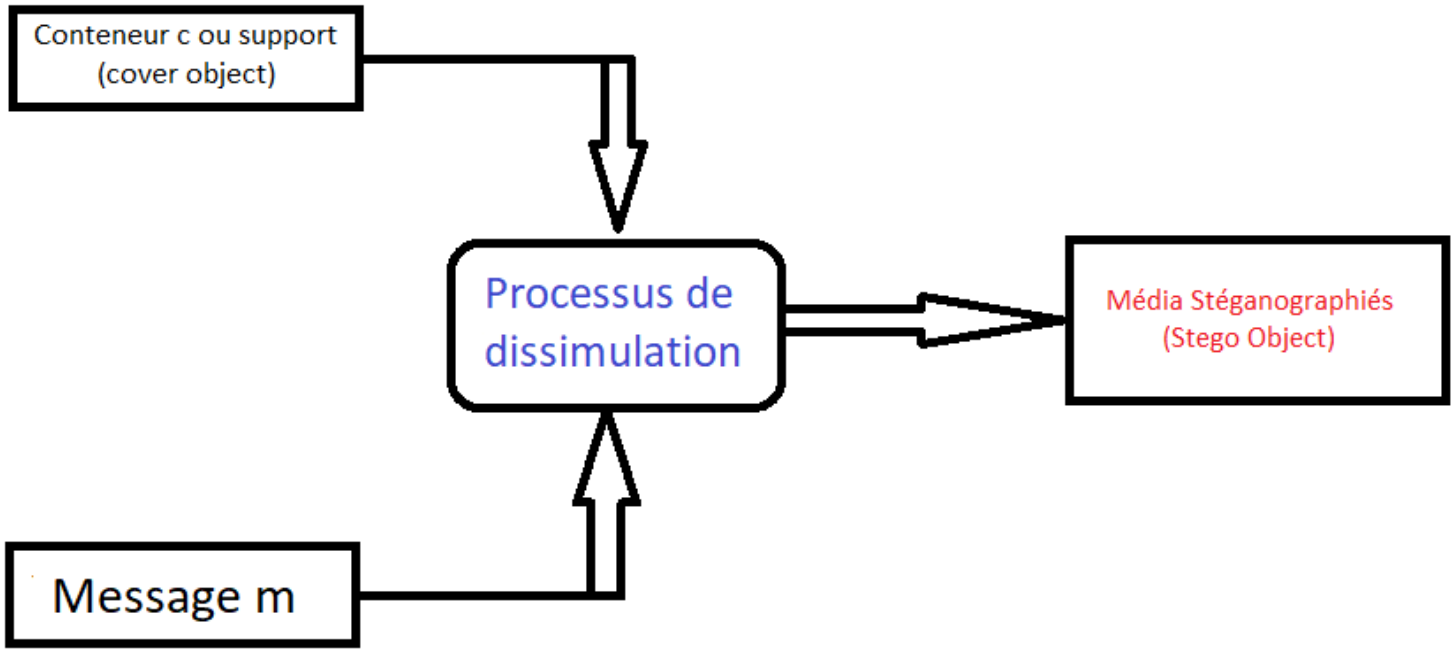
Introduction

La Stéganographie qui est l'art de la dissimulation des messages dans des supports, est soumise à un compromis: plus on injecte de l'information dans un support, plus celui-ci est détérioré, plus l'information est détectable.

Problématique

Comment peut-on donc minimiser les modifications du support lors de l'insertion de l'information ?

Schéma simplifié de la Stéganographie



Positionnement problématique

On souhaite transmettre un message au travers d'un canal de communication bruité : les données émises sont modifiées par quelques erreurs lorsqu'elles sont reçues. Pour permettre néanmoins la transmission fiable d'un message, la donnée émise est un codage du message, plus long que le message lui-même. La redondance ainsi ajoutée permet de corriger ensuite des erreurs qui peuvent apparaître lors de la transmission.

Les codes correcteurs d'erreurs dans leur contexte courant

Un code correcteur est un ensemble des techniques de codage basé sur la redondance. Celle-ci est destinée à corriger les erreurs de transmission d'une information (le plus souvent appelée message) sur un canal de communication non suffisamment fiable. La théorie des codes correcteurs ne se limite pas qu'aux communications classiques (radio, câble coaxial, fibre optique, etc.) mais également aux supports pour le stockage comme les disques compacts, la mémoire RAM et d'autres applications où l'intégrité des données est importante.

Définitions

Soit un corps fini $\mathbb{F}_q$ et soient deux entiers k et n, $0 \leqslant k \leqslant n$.

- Un code linéaire de longueur n et de dimension k sur l'alphabet $\mathbb{F}_q$ est un sous-espace vectoriel C de $\mathbb{F}_q^n$ de dimension k. On dit qu'il s'agit d'un code linéaire de paramètres (q,n,k). Si ce code admet d pour distance minimale, on parlera de code linéaire de paramètres (q,n,k,d).
- Une matrice génératrice d'un code linéaire C (de paramètres (q,n,k)) est une matrice $G \in M_{k,n}(\mathbb{F}_q)$ dont les lignes forment une base de C. G est donc une matrice de rang k telle que:

$$C = \{xG; x \in \mathbb{F}_q^k\}$$

Définitions

- Une matrice de contrôle H (parity-check matrix) d'un [n, k]-code linéaire est une matrice de taille (n - k) x n, avec n - k lignes linéairement indépendantes, telle que chaque ligne de G est orthogonal à toutes les lignes de H.
  La donnée d'une matrice de contrôle H permet de déterminer facilement si un mot y $\in \mathbb{F}_q^n$ appartient au code C:
  $\forall y \in \mathbb{F}_q^n, y \in C \Leftrightarrow Hy^T = 0$
- (Distance de Hamming): Soient u, v $\in \mathbb{F}_q^n$. La distance de Hamming entre u et v, notée $d_H(u, v)$, est donné par l'expression suivante : $d_H(u, v) = card\{i, u_i \neq v_i\}$ en notant $u, v = (u_i)_{i \in [\![1,n]\!]}, (v_i)_{i \in [\![1,n]\!]}$.

Définitions

- (Poids d'un mot): Soit v un mot de l'espace $\mathbb{F}_q^n$, le poids de Hamming de v, noté w(v), est défini comme suit :

$$w(v) = card\{i, u_i \neq 0\}$$

- (Distance minimale d'un code): La distance minimale d'un code C, notée d, est le minimum des distances de Hamming entre les mots du code. On a donc :

$$d = min\{d_H(u, v)/u, v \in C, u \neq v\}$$

Borne du Singleton

Proposition

Soit C un [n, k]-code de distance minimal d. L'inégalité suivante est appelé borne de singleton : $d \leqslant n - k + 1$.
Cette borne permet de trouver une borne maximale sur la distance minimale d par rapport aux valeurs n et k.

Remarques

- Dans le domaine du codage les mots x $\in \mathbb{F}_q^k$ sont généralement représentés par des vecteurs lignes.
- $d_H(u, v) = w(u - v)$
- On pose $t = \lfloor \frac{d-1}{2} \rfloor$, t s'appelle capacité de correction.

Détection et correction des erreurs

L'encodage d'un mot $u \in \mathbb{F}_q^k$ s'effectue en transformant le vecteur u en un mot c du code C de longueur n tel que :

$$c = u \times G$$

Puis on envoie c au travers du canal. A la sortie du canal, on reçoit un mot r. L'objectif du décodeur est de reconstruire le mot qui a été envoyé à partir du mot reçu. Le décodeur il cherche d'abord à détecter s'il y a eu une erreur, puis dans un deuxième temps, il cherche à corriger cette ou ces erreurs.

Détection et correction des erreurs

Une méthode pour détecter la présence d'erreurs et de calculer le syndrome : $s = r \times H^t$

- Si le syndrome s est égal au vecteur nul, étant donné que le produit d'un des mots du code C avec la transposée de la matrice de contôle est nulle, on peut déduire que r est un mot appartenant au code C. Le récepteur va alors considérer qu'il n'y a pas d'erreur, et qu'il s'agit du mot envoyé.
- Si le syndrome est différent du vecteur nul, cela signifie que le mot reçu r n'est pas un mot du code C. Par conséquent, on est sûr qu'une erreur ou plus est apparue lors de la transmission. Le but du décodeur va donc être de trouver où se situe cette erreur et de la corriger.

- Voyons plus précisément le cas des codes binaires : On pose $e = (e_1, e_2, ..., e_{n-1})$, le vecteur d'erreur, représentant les erreurs ayant été introduites par le canal. Ce vecteur a des zéros partout sauf aux positions où une erreur s'est produite. On a alors:

$$r = c + e$$

Le syndrome s est alors égal à :

$$\begin{aligned} s &= (c + e) \times H^t \\ &= c \times H^t + e \times H^t \\ &= e \times H^t \end{aligned}$$

($c \times H^t = 0$ puisque $c \in C$)

- On peut donc calculer les syndromes possibles à partir des vecteurs d'erreurs et constituer une table associant à chaque erreur pouvant apparaître dans le canal le syndrome que l'on obtiendrait. Ainsi lors du décodage, on pourra comparer s avec les entrées de la table et retrouver le vecteur erreur. Lorsque e est connu, on retrouve facilement c en calculant :

$$c = r - e$$

Définition des GRS (Generalized Reed Solomon codes) :

On désigne l'ensemble des polynômes à coefficients dans $\mathbb{F}_q$ et de degré strictement inférieur à k par
$\mathbb{F}_q[X]_k = \{P \in \mathbb{F}_q[X], degP < k\}$,
de sorte que $dim_{\mathbb{F}_q}(\mathbb{F}_q[X]_k) = k$.
Soient $0 \leqslant k \leqslant n \leqslant q$, $v = (v_0, v_1, ..., v_{n-1}) \in \mathbb{F}_q^{*n}$ et:
$\alpha = (\alpha_0, \alpha_1, ..., \alpha_{n-1}) \in \mathbb{F}_q^n$ tel que $i \neq j \Rightarrow \alpha_i \neq \alpha_j$.
On définit alors le code de Reed-Solomon par:

$$GRS_{n,k}(\alpha, v) = \{ev_{\alpha,v}(P), P \in \mathbb{F}_q[X]_k\} \subset \mathbb{F}_q^n,$$

où l'évaluation se fait par:

$$ev_{\alpha,v}(P) = (v_0 P(\alpha_0), v_1 P(\alpha_1), ..., v_{n-1} P(\alpha_{n-1})) \in \mathbb{F}_q^n.$$

Programme Python : Codage GRS

```python
def valeur(U, x):
    d = len(U) - 1
    s = U[d]
    for i in range(d-1,-1,-1):
        s = s * x + U[i]     #Algorithme de Horner
    return s

def codageGRS(n, alpha, v, P):
    return [v[i]*valeur(P, alpha[i])
            for i in range(n)]
#Donc:
```

$C = GRS_{n,k}(\alpha, v) = [codageRS(n, \alpha, v, P), \text{ for P in } \mathbb{F}_q[X]_k]$

L'interpolation de Lagrange pour décoder dans le cas sans erreur :

Soit $\alpha = (\alpha_0, \alpha_1, ..., \alpha_{n-1}) \in \mathbb{F}_q^n$ tel que $i \neq j \Rightarrow \alpha_i \neq \alpha_j$.
Définissons les polynômes de Lagrange par:

$$\forall i \in [\![0, n-1]\!],\ L_i(X) = \prod_{j \in [\![0,n-1]\!],\ j \neq i} (X - \alpha_j)\ \in \mathbb{F}_q[X]_n\ ,$$

Ces polynômes rendent facile l'interpolation polynomiale:
$\forall P \in \mathbb{F}_q[X]_n$, on a l'identité:

$$P(X) = \sum_{i=0}^{n-1} P(\alpha_i)(L_i(\alpha_i))^{-1} L_i(X)\ .$$

Les GRS sont des codes linéaires MDS :

Soient $0 \leqslant k \leqslant n \leqslant q$, $\mathrm{v} = (v_0, v_1, ..., v_{n-1}) \in \mathbb{F}_q^{*n}$ et $\alpha = (\alpha_0, \alpha_1, ..., \alpha_{n-1}) \in \mathbb{F}_q^n$ tel que $i \neq j \Rightarrow \alpha_i \neq \alpha_j$.
Alors $GRS_{n,k}(\alpha, v)$ est un code linéaire sur $\mathbb{F}_q$ de dimension k. De plus, c'est un code MDS (Maximum Distance Separable), i.e sa distance minimale atteint la borne de singleton : d = n - k + 1. Un code $GRS_{n,k}(\alpha, v)$ a donc pour paramètres (q, n, k, n - k + 1).

Preuve

- L'application d'évaluation:
$$\begin{array}{rccc} ev_{\alpha,v} & : & \mathbb{F}_q[X]_k & \to & \mathbb{F}_q^n \\ & & P & \mapsto & (v_0 P(\alpha_0), ..., v_{n-1} P(\alpha_{n-1})) \end{array}$$
est $\mathbb{F}_q$-linéaire et injective (car $P \neq 0$ admet au plus k-1 zéros distincts). Par suite $GRS_{n,k}(\alpha, v) = Im(ev_{\alpha,v})$ est un $\mathbb{F}_q$-sous-espace vectoriel de $\mathbb{F}_q^n$ de dimension k.
- Encore par le fait que $P \neq 0$ ne peut avoir que k-1 zéros au plus, il vient:
$\forall P \neq 0,\ w(ev_{\alpha,v}(P)) \geqslant n - (k-1) = n - k + 1,$
d'où d = n - k + 1 (en utilisant la borne du singleton).

Programme Python:Taux de correction des GRS pour q = 5

```
1  from matplotlib import pyplot as plt
2  import numpy as np
3  def Taux(n, k):
4      d = n - k + 1
5      return d/n
6  def RScorrection(q):
7      t = []
8      for k in range(1, q+1):
9          for n in range(k, q+1):
10             t = t + [(n, k)]
11     tau = [Taux(t[i][0], t[i][1])
12            for i in range(len(t))]
13     return t, tau
14 q = 5
15 t, tau = RScorrection(q)
16 x = np.array([i for i in range(len(t))])
17 y = np.array(tau)
18 plt.xticks(x, t, rotation=90)
19 plt.plot(x, y)
20 plt.show()
```

Visualisation des taux de correction des GRS

(exemple : pour q=5)

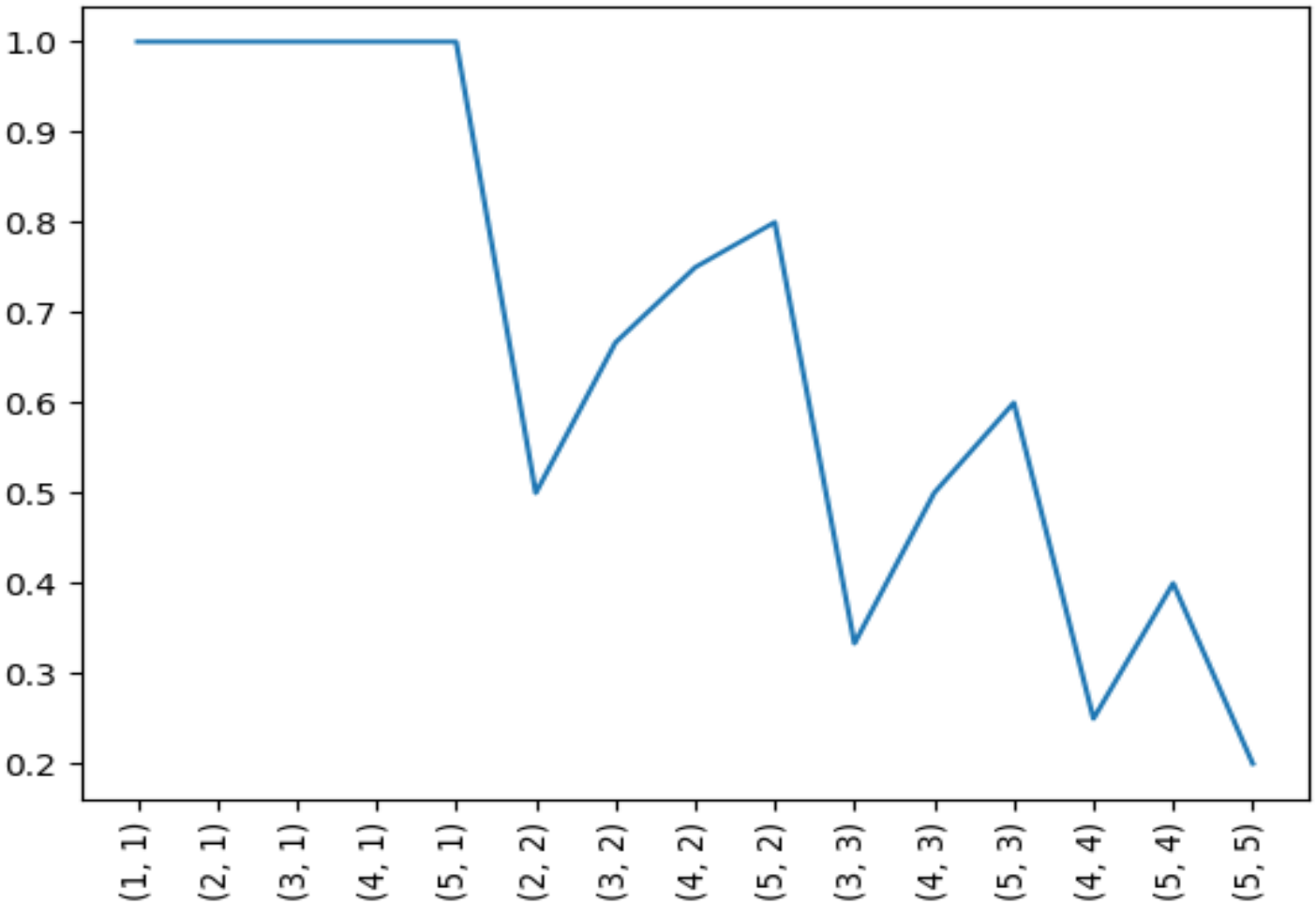
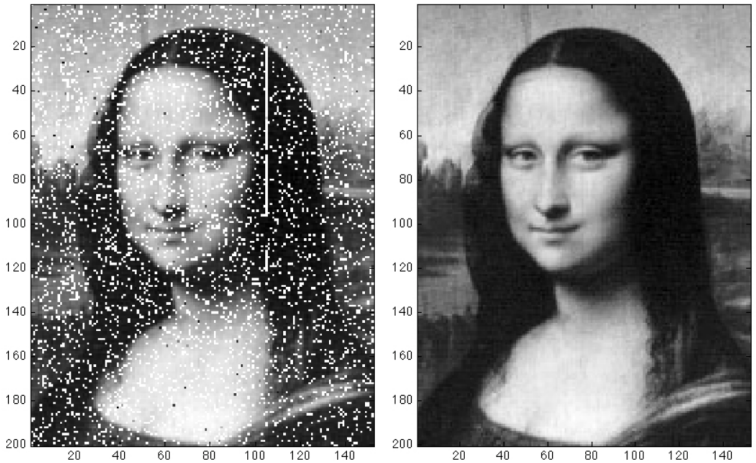


Image de La Joconde, transmis au Lunar Reconnaissance Orbiter. L'image à gauche, captée par la NASA.



À droite, le Code de Reed–Solomon remet de l'ordre dans l'image. Pour nettoyer les erreurs de transmission introduites par l'atmosphère terrestre (à gauche), les scientifiques ont appliqué la correction d'erreur Reed-Solomon (à droite). Les erreurs typiques incluent les pixels manquants (blanc) et les faux signaux (noir). La bande blanche indique une brève période pendant laquelle la transmission a été interrompue.

Théorème (Le code dual d'un GRS est un GRS)

Soient $0 \leqslant k \leqslant n \leqslant q$, $\mathrm{v} = (v_0, v_1, ..., v_{n-1}) \in \mathbb{F}_q^{*n}$ et $\alpha = (\alpha_0, \alpha_1, ..., \alpha_{n-1}) \in \mathbb{F}_q^n$ tel que $i \neq j \Rightarrow \alpha_i \neq \alpha_j$. Alors on a: $GRS_{n,k}(\alpha, v)^{\perp} = GRS_{n,n-k}(\alpha, u)$, où $u = (u_0, u_1, ..., u_{n-1}) \in \mathbb{F}_q^{*n}$ se calcule par:
$\forall i \in [\![0, n-1]\!], u_i = v_i^{-1} L_i(\alpha_i)^{-1}$.

Preuve

Soient $P \in \mathbb{F}_q[X]_k$ et $Q \in \mathbb{F}_q[X]_{n-k}$. On a:

$$\begin{aligned}\langle ev_{\alpha,v}(P), ev_{\alpha,u}(Q)\rangle &= \sum_{i=0}^{n-1} v_i P(\alpha_i) u_i Q(\alpha_i) \\ &= \sum_{i=0}^{n-1} P(\alpha_i) Q(\alpha_i) L_i(\alpha_i)^{-1} = 0 \ ,\end{aligned}$$

puisqu'on a reconnu le coefficient de X^{n-1} du polynôme $P(X)Q(X) \in \mathbb{F}_q[X]_{n-1}$ qu'on peut exprimer par interpolation de Lagrange:

$$P(X)Q(X) = \sum_{i=0}^{n-1} P(\alpha_i) Q(\alpha_i) L_i(\alpha_i)^{-1} L_i(X).$$

Exploitons cette dualité pour obtenir une autre expression du code $C = GRS_{n,k}(\alpha, v)$.
Posons r = n-k.

$$
\begin{aligned}
y \in C = (C^{\perp})^{\perp} &\Leftrightarrow y \perp C^{\perp} = GRS_{n,r}(\alpha, u) \\
&\Leftrightarrow \forall j \in [\![0, r-1]\!], \langle y, ev_{\alpha,u}(X^j)\rangle = 0 \\
&\Leftrightarrow \sum_{j=0}^{r-1} \langle y, ev_{\alpha,u}(X^j)\rangle X^j = 0 \\
&\Leftrightarrow \sum_{j=0}^{r-1} \left(\sum_{i=0}^{n-1} y_i \; u_i \; \alpha_i^j\right) X^j = 0
\end{aligned}
$$

$$\Leftrightarrow \sum_{i=0}^{n-1} y_i \; u_i \; (\sum_{j=0}^{r-1} (\alpha_i X)^j) = 0.$$

Avec $\sum_{j=0}^{r-1} (\alpha_i X)^j \equiv (1 - \alpha_i X)^{-1} [X^r]$, puisque:

$(1 - \alpha_i X)(\sum_{j=0}^{r-1} (\alpha_j X)^j) = 1 - (\alpha_i X)^r \equiv 1[X^r]$.

D'où le résultat suivant :

Théorème (Formulation de Goppa pour les codes GRS)

Soient $0 \leqslant k \leqslant n \leqslant q, r = n - k, v = (v_0, v_1, ..., v_{n-1}) \in \mathbb{F}_q^{*n}$ et $\alpha = (\alpha_0, ..., \alpha_{n-1}) \in \mathbb{F}_q^n$, tel que $i \neq j \Rightarrow \alpha_i \neq \alpha_j$. Alors on a:
$GRS_{n,k}(\alpha, v) = \{y = (y_0, ..., y_{n-1}) \in \mathbb{F}_q^n :$
$\sum_{i=0}^{r-1} y_i \ u_i \ (1 - \alpha_i X)^{-1} \equiv 0[X^r]$, où $u = (u_0, ..., u_{n-1}) \in \mathbb{F}_q^{*n}$ se calcule encore par:
$\forall i \in [\![0, n-1]\!] \quad u_i = v_i^{-1} L_i(\alpha_i)^{-1}$ et
$(1 - \alpha_i X)^{-1} \equiv \sum_{j=0}^{r-1} (\alpha_i X)^j \ [X^r]$.

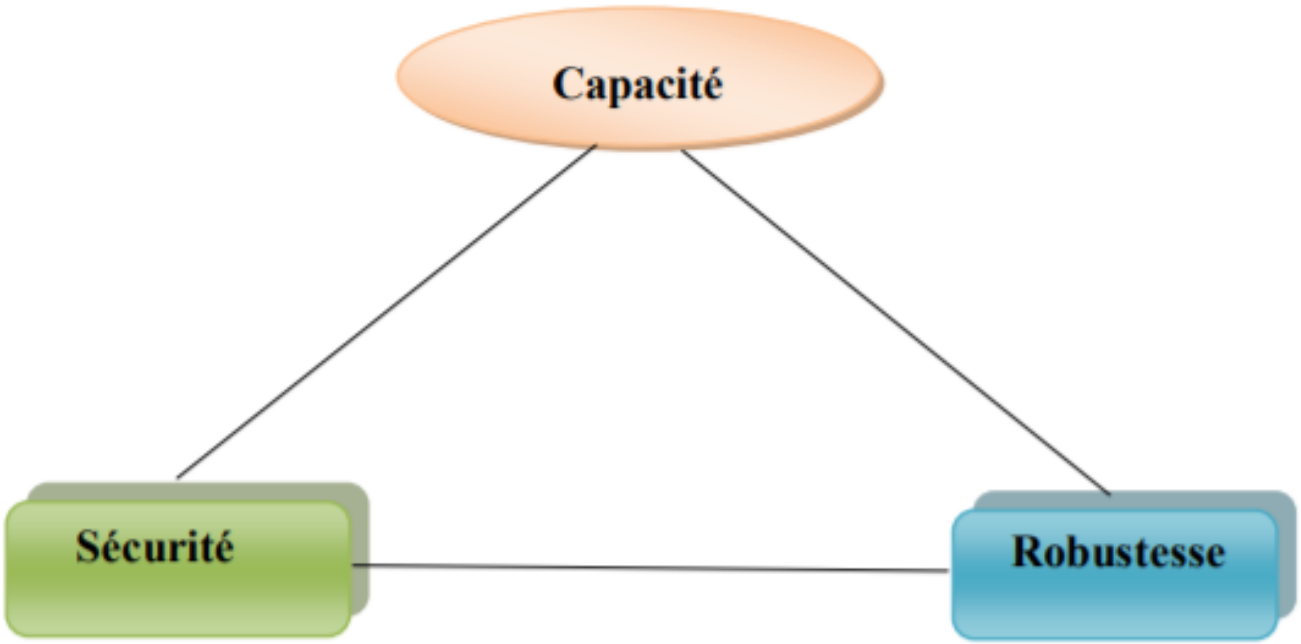
Caractéristiques d'un Schéma Stéganographique

Trois critères permettent de classer les algorithmes stéganographiques : La capacité, la transparence et la robustesse.

1. La capacité correspond à la taille de données qui peut être incorporé dans l'objet de couverture, relativement à la taille de celui-ci,
2. La transparence ou l'imperceptibilité dépende directement à la distorsion introduite par le processus de dissimulation pendant l'insertion de données, la distorsion est tous simplement les nombres de modification ou de changement dans l'objet de couverture,
3. La robustesse signifie la résistance de notre stégo-objet, c-à-d rester normale même s'il subit des transformation (filtrage, etc...).

Caractéristiques d'un Schéma Stéganographique

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



Triangle des caractéristiques d'un schéma Stéganographique

Définition (Schéma stéganographique)

Soient n, k deux entiers positifs, $k \leqslant n$ et soit B un ensemble fini. Un protocole stéganographique de type [n, k] sur B c'est une paire de fonctions $\Gamma = (f, g)$ telles que :

$$f : B^k \times B^n \to B^n \; ; \; g : B^n \to B^k$$

avec g(f(s, x)) = s, $\forall s \in B^k$ et $x \in B^n$, f c'est la fonction de dissimulation et g la fonction de récupération. Le nombre $\rho = max\{d(x, f(s, x)), s \in B^k, x \in B^n\}$ c'est le rayon de protocole. Ainsi, la fonction de dissimulation d'un [n, k] protocole stéganographique de rayon ρ, (dorénavant ce type de protocole est noté [n, k, ρ]) nous permet de cacher les symboles de l'information de longueur k dans une suite de n symboles de couverture.

Définition (Schéma stéganographique)

L'information secrète s plus tard est extraite à partir de f(s,x) en employant la fonction de récupération. Un bon protocole doit remplir deux conditions principales :

1. il faut avoir des algorithmes efficaces de dissimulation/récupération ;
2. avoir les bons paramètres k, n, ρ, (tels que $\frac{k}{n}$ soit le plus grand possible et $\frac{\rho}{n}$ le plus petit possible). Le protocole serait approprié si f(s, x) est l'élément le plus proche à x dans l'ensemble $g^{-1}(s) = \{ y \in B^n / g(y) = s\}$ (noter que , étant donné que $n \geqslant k$ alors g est non injective en général et par conséquent $g^{-1}(s)$ est un ensemble et pas un élement simple).

Exemple

Considérer un protocole de LSB appliqué à une image BMP. Si chaque Pixel est représenté par h bits, nous pouvons considérer:

1. le support de couverture est l'image entière ; le protocole (pour chaque Pixel) est la paire de fonctions $f : \mathbb{F}_2 \times \mathbb{F}_2^h \to \mathbb{F}_2^h; g : \mathbb{F}_2^h \to \mathbb{F}_2$ définies par: $f(s, x_1, ..., x_h) = (x_1, x_2, ..., x_{h-1}, s), g(y_1, ..., y_h) = y_h$. Alors ce couple de fonctions (f, g) définit un [1, h, 1]-protocole stéganographique.
2. le support de couverture est l'ensemble de tous les bits de poids faible (LSB); le protocole (pour chaque Pixel) est la paire de fonctions $f : \mathbb{F}_2 \times \mathbb{F}_2^h \to \mathbb{F}_2^h; g : \mathbb{F}_2^h \to \mathbb{F}_2$, avec f(s, x) = s; $g = Id_{\mathbb{F}_2}$, dans ce cas (f, g) représente le protocole [1, 1, 1]-stéganographique.

Sécurité d'un schéma de Stéganographie

Le but d'un schéma stéganographique est d'insérer un message dans un support, en cachant l'existence même de ce message. Ainsi, on peut discerner deux critères d'évaluation :

- la capacité.
- l'indétectabilité de ce message.

Il est évident de remarquer à première vue que ces deux critères sont en opposition : plus le message est gros, plus il sera difficile à cacher. L'émetteur et le récepteur vont donc devoir trouver un compromis entre la capacité et l'indetectabilité afin de construire un bon schéma stéganographique. Concernant la capacité, plusieurs mesures sont utilisées. Tout d'abord la capacité d'insertion, définie par $\log_2 |M|$, $|M|$ étant le nombre de message possibles. On peut également définir la capacité d'insertion relative, qui est la capacité divisé par la taille du support.

Sécurité d'un schéma de Stéganographie

Enfin, le critère le plus utilisé est "l'efficacité d'insertion" (embedding efficiency), qui est par définition le nombre de bits insérés pour une modification dans le support. Évidemment, plus ce nombre est grand, plus nous avons la possibilité d'insérer un grand nombre de bits dans le message, car il y aura alors peu de modifications, donc peu de détection. Cependant, le nombre de modifications apportées au support n'est pas directement le critère le plus adapté pour caractériser la sécurité d'un schéma. En effet, nous allons voir que la manière de le modifier est plus importante pour préserver l'indétectabilité des données cachées. Afin de déterminer la sécurité d'un système, nous allons comparer la distribution d'un support, avant et après insertion. En effet, soit c un support de couverture de distribution de probabilité Pc, assimilé à une variable aléatoire sur l'alphabet Γ et s, de distribution de probabilité Ps, l'image de c par la fonction d'insertion d'un schéma stéganographique.

Sécurité d'un schéma de Stéganographie

La sécurité de ce schéma est alors dépendante des similitudes entre Pc et Ps. Plus précisément, on utilise l'entropie relative des deux distributions de probabilité pour définir ce critère. Cette distance (qui n'est pas une distance au sens mathématique du terme, car non symétrique) est définie par : $D(Pc\|Ps) = \sum_{i\in\Gamma} Pc(i)\log_2(\frac{Pc(i)}{Ps(i)})$, Le schéma est alors considéré comme sûr si $D(Pc\|Ps) = 0$, et considéré comme ε sûr si $D(Pc\|Ps) < \varepsilon$. Plaçons nous du côté de l'attaquant. Considérons que celui-ci peut commettre deux erreurs lors de l'interception d'un message entre Alice et Bob :

- le message est considéré comme illégal alors qu'il ne contient pas de données cachées. Supposons la probabilité de cette erreur égale à α.
- le message est considéré comme légal alors qu'il contient des données cachées. Supposons la probabilité de cette erreur égale à β.

Sécurité d'un schéma de Stéganographie

Face à un message intercepté, l'attaquant doit répondre 0 (message légal) ou 1 (message illégal). Alors :

- face à un message légal, l'attaquant va répondre 1 avec une probabilité égale à α , et 0 avec une probabilité égale à $1 - \alpha$.
- face à un message illégal, l'attaquant va répondre 1 avec une probabilité égale à $1 - \beta$, et 0 avec une probabilité égale à β. Alors l'entropie relative entre les deux distributions du detecteur de l'attaquant est égale à :
  $D(\alpha\|\beta) = (1-\alpha)\log_2(\frac{(1-\alpha)}{\beta}) + \alpha\log_2(\frac{\alpha}{(1-\beta)})$ α et β résultants de traitements sur Pc et Ps, leur entropie relative ne peut pas être supérieure à $D(Pc\|Ps)$. Ainsi on a :
  $D(\alpha\|\beta) \leqslant D(Pc\|Ps)$

Sécurité d'un schéma de Stéganographie

si $\alpha = 0$, c'est à dire que l'on interdit à l'attaquant de considérer des messages illégaux alors qu'ils sont légaux, alors:
$\log_2(\frac{1}{\beta}) \leqslant D(Pc\|Ps)$. Or comme $D(Pc\|Ps) \leqslant \varepsilon$:
$\log_2(\frac{1}{\beta}) \leqslant \varepsilon \Leftrightarrow \beta \geqslant 2^{-\varepsilon}$
Conclusion: Ainsi, plus ε est petit, plus la probabilité qu'un message illégal ne soit pas détecté est grande.

Codes de Reed-Solomon en Stéganographie

Nous avons un vecteur $v \in \mathbb{F}_q^n$ qui est extrait d'un média de couverture et un message m de longueur r composé d'élément de $\mathbb{F}_q$. Nous voulons modifier v en s tel que m soit incrusté dans s en changeant au plus T coordonnées dans v.
On propose d'utiliser le codage par syndrôme avec un code GRS. Pour cela, on utilise $GRS_{n,k}$ pour introduire le message m et ainsi trouver le vecteur s suffisamment proche de v. On suppose alors que l'on a fixé une famille $\gamma \in \mathbb{F}_q^n$. L'extraction sur s contient le message m si $s \times H^t = m$. Pour construire s, nous cherchons un mot y tel que son syndrôme est $m - v \times H^t$. En effet,
si on pose y = s - v, alors:
$y \times H^t = (s - v) \times H^t = s \times H^t - v \times H^t = m - v \times H^t$
Alors le poids de Hamming de y est exactement le nombre de changement effectué pour passer de v à s. On veut que ce poids soit inférieur à T.

Or, quand T est inférieur ou égal au rayon de recouvrement du code ρ, un tel vecteur existe toujours, car:
$w(y) = d_H(0, y) = d_H(v, y + v)$
$\leqslant \rho = max\{d_H(v, \underbrace{f(m, v)}_{s = y + v}), m \in \mathbb{F}_q^{(n-k)}, v \in \mathbb{F}_q^n\}$
Soit le polynôme M(X) qui a pour coefficients m_i associés aux monômes X^{k+i} avec i $\in \{0, ..., n - k - 1\}$. On a alors $ev(M) \times H^t = m$ car la multiplication renvoie exactement les coefficients de degré au moins k. Maintenant, pour trouver y, il nous suffit de calculer un polynôme P de degré inférieur à k tel que pour au moins k éléments γ_i de la famille γ:

$$P(\gamma_i) = M(\gamma_i) - V(\gamma_i)$$

Ainsi, avec un tel P, on a y = ev(M - V - P) qui a k coordonnées égales à zéro.

Le syndrôme est alors:

$$\begin{aligned} y \times H^t &= ev(M - V - P) \times H^t \\ &= (ev(M) - ev(V) - ev(P)) \times H^t \\ &= ev(M) \times H^t - ev(V) \times H^t - ev(P) \times H^t \\ &= m - v \times H^t \end{aligned}$$

On pose alors s = y + v et on obtient notre vecteur modifié comme désiré. De plus, on peut remarquer que verrouiller la position i requiert $s_i = v_i$ et que c'est équivalent à prendre $y_i = 0$ et donc que $P(\gamma_i) = M(\gamma_i) - V(\gamma_i)$. De plus T peut être aussi grand que le rayon de recouvrement $\rho = n - k$ car on peut verrouiller jusqu'à k positions.

Pour construire P, on propose d'utiliser la méthode d'interpolation de Lagrange. On choisit un ensemble de k coordonnées $I = \{i_1, ..., i_k$ et on calcule :
$P(X) = \sum_{i \in I}(M(\gamma_i) - V(\gamma_i))\dot{L}_i(X)$, avec:
$L_i(X) = \prod_{i \in I \setminus \{i\}} \frac{X - \gamma_j}{\gamma_i - \gamma_j}$
P satisfait donc: $\forall i \in I, P(\gamma_i) = M(\gamma_i) - V(\gamma_i)$. Donc $\forall i \in I, y_i = 0$ et donc $s_i = v_i + y_i = v_i$. Ce qui signifie que l'on a bloqué les positions dont l'indice appartient à I.
D'après ce qui précède nous pouvons donc introduire n - k éléments de $\mathbb{F}_q$ en changeant au plus T = n - k coordonnées parmi n. On en déduit donc que dans le pire des cas, l'efficacité est égale à 1.

En fait, en incrustant n - k symboles tout en bloquant k symboles parmi n est optimal. Comme nous l'avons écrit précédemment, le blocage des positions d'indice appartenant à I conduit à une équation $y \times H_I^t = m$, où H_I^t est de dimension $(n-k) \times (n-|I|)$. Ainsi, pour $|I| > k$ il existe certaines valeurs de m pour lesquelles il n'y a pas de solution. D'autre part, soit nous supposons que nous connaissons un code avec une matrice de contrôle H telle que pour chaque I de taille k et pour chaque m cette équation a une solution, donc H_I est inversible. Cela implique que chaque sous matrice de H de taille $(n-k) \times (n-k)$ est inversible. Or cela est équivalent à la définition des codes MDS.

Conclusion: Les codes GRS sont donc optimaux dans la mesure où nous pouvons verrouiller autant de position que possible pour un message de longueur n - k.