**Série 1(La stéganographie)**

La stéganographie (du grec steganos, couvert et graphein, écriture) est l’art de cacher un message au sein d’un autre message de caractère anodin, de sorte que l’existence même du secret en soit dissimulée. Alors qu’avec la cryptographie habituelle, la sécurité repose sur le fait que le message ne sera sans doute pas compris, avec la stéganographie, la sécurité repose sur le fait que le message ne sera sans doute pas détecté.

**Exercice 1 : (Analyse de texte )**

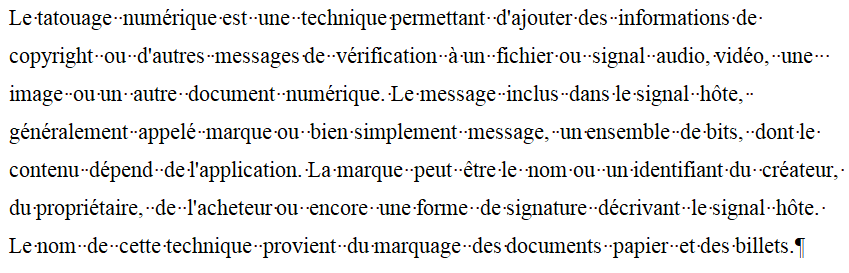
Le tatouage numérique est une technique permettant d'ajouter des informations de copyright ou d'autres messages de vérification à un fichier ou signal audio, vidéo, une image ou un autre document numérique. Le message inclus dans le signal hôte, généralement appelé marque ou bien simplement message, un ensemble de bits, dont le contenu dépend de l'application. La marque peut être le nom ou un identifiant du créateur, du propriétaire, de l'acheteur ou encore une forme de signature décrivant le signal hôte. Le nom de cette technique provient du marquage des documents papier et des billets.

« https://fr.wikipedia.org/wiki/Tatouage\_numérique »

1. Analyser le texte ci-dessus
2. Comment détecter automatiquement l’existence d’un texte dissimilé
3. Comment utiliser cette technique pour transférer un mot de passe.

**Solution 1 :**

1. Présence de blancs (espaces entre les mots).



1. On remplace l’espace par un autre symbole pour remarquer cette dissimulation.
2. Coder par exemple :

1 espace suivi de deux espaces 0

2 espaces suivis de 1 espace 1

**Exercice 2 : (Code Barn )**

Un moyen de communication assez complexe mais parfaitement efficace était donné par la dissimulation d'un court message dans une lettre ordinaire selon le "Barn Code". Nous en donnons un exemple détaillé ; supposons que l'agent reçoive par la poste la lettre ci-après :

Mon cher Pierre,

J'espère que tu voudras bien m'excuser, mais j'ai eu tellement de travail à la maison que je n'ai pas pris le temps d'écrire aux amis. Cependant je t'envoie ce petit mot d'urgence pour te faire savoir que si tu veux des pneus, tu ferais bien de te dépêcher ; en effet :

Hier, Jean est venu nous rendre visite, il descendait du train et s'est arrêté un moment chez nous pour bavarder et donner des nouvelles à mon père de son Paris. En principe, il doit rester quelques jours ici pour mettre en ordre ses affaires avant de repartir pour la capitale. A Paris, c'est calme, mais la veille il avait été dérangé en plein sommeil par les sirènes deux fois dans la nuit ! Ceci mis à part, il doit nous faire envoyer par un ami à lui des pneus neufs pour nos vélos. Il en a pour le moment, profitons-en ! A bientôt de tes nouvelles.

Trouver le message caché, si on utilise le quatrième mot du premier paragraphe comme clé.

**Solution 2 :**

Le message caché:

V o u d r a s

7 3 6 2 4 1 5

Hier, Jean est venu nous rendre visite,

il descendait du train et s'est arrêté

un moment chez nous pour bavarder et

donner des nouvelles à mon père de

son Paris. En principe, il doit rester

quelques jours ici pour mettre en ordre

ses affaires avant de repartir pour la

capitale. A Paris, c'est calme, mais la

veille il avait été dérangé en plein

sommeil par les sirènes deux fois dans

la nuit ! Ceci mis à part, il

doit nous faire envoyer par un ami

à lui des pneus neufs pour nos

vélos. Il en a pour le moment,

profitons-en ! A bientôt de tes nouvelles.

L’inconvénient de cette méthode réside dans la difficulté de trouver un texte et une clé.

**Exercice 3 : ( Discussion)**

1. Calculer la taille minimale et maximale d’un texte dissimilé dans une image couleur (RVB) de 600\*800 pixels. L’image de format BMP
2. Calculer la taille minimale et maximale d’un texte dissimilé dans une image de 256 couleurs de 600\*800 pixels. L’image de format BMP.
3. Même question pour un autre format d’image.

**Solution 3:**

1. Il faut 3 octets pour coder chaque pixel :

Taille minimale si on prend un bit pour le texte (3 bits par pixel), donc : 600\*800\*3\*1= 1 440 000‬ bits= 180 000‬ octets.

Taille maximale si on prend 4 bits pour le texte (4 bits par octet et 12 bits par pixel) donc : 600\*800\*3\*4= 5 760 000‬ bits= 720 000‬ octets.

1. Les images en 256 couleurs utilisent 8 bits par pixel, ce qui signifie qu'un octet code chaque pixel

Taille minimale si on prend un bit pour le texte (1 bit par pixel), donc : 600\*800\*1= 480 000‬ bits= 60 000‬ octets.

Taille maximale si on prend 4 bits pour le texte (4 bits par octet et par pixel) donc : 600\*800\*4= 1 920 000‬‬ bits= 240 000‬‬ octets.

**Exercice 4 :**

1. Proposer une méthode de stéganographie pour cacher un texte de longueur L dans une image couleur. (Définir l’entête, l’algorithme d’ajout du message, et la méthode de restituer le message)
2. Même question pour une image en niveau de gris.

**Solution 4 :**

Q1. Codage :

1. Le message est caché dans les pixels de l’image.
2. Le premier de ces pixels (celui de coordonnées (0,0)) contient la longueur du message en nombre de caractères. Appelons *L* cette longueur.
3. Les 8*L* pixels qui suivent contiennent le message codé en binaire par le code ASCII à raison d’un bit par pixel.
4. Chacun de ces bits est dissimulé en jouant sur la parité des composantes vertes et bleues :
   1. si le bit à dissimuler est un 0, alors on s’arrange pour que les deux composantes vertes et bleues aient la même parité. Si c’est le cas, on ne change pas la couleur. Si ce n’est pas le cas on modifie la composante bleue en lui ajoutant 1 si elle est initialement paire, ou en lui retranchant 1 dans le cas contraire ;
   2. si le bit à dissimuler est un 1, alors on s’arrange pour que les deux composantes vertes et bleues n’aient pas la même parité. Pour cela on procède de la même façon que dans le cas précédent.

Décodage :

La récupération du texte dans l’image s’opère de la manière suivante :

– Récupérer la matrice décrivant l’image.

– Remplacer les deux composantes vertes et bleues ayant la même parité par 0, dans le cas contraire par 1.

– Grouper les bits par groupe de 8.

– Convertir chaque octet en nombre décimal.

– Écrire les caractères correspondant aux codes ASCII obtenus.

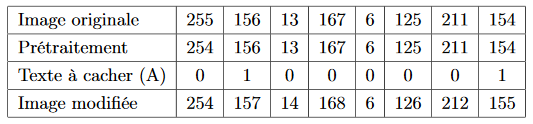
Q2. Une image en niveaux de gris peut être décrite comme une matrice composée de nombres allant de 0 à 255. A chaque nombre représentant un pixel, on associe un niveau de gris :

0→ noir, ..., 255→ blanc.

**Codage :**

À chaque nombre pair correspond un 0 du texte et à chaque nombre impair correspond un 1. Il faudra donc modifier certains pixels de l’image, mais ces altérations seront invisibles à l’œil. On va procéder ainsi : si le nombre du tableau de l’image a la parité que l’on veut, on le laisse inchangé. Si ce n’est pas le cas, on lui ajoute 1. Cette méthode nécessite un prétraitement de l’image : tous les pixels à 255 seront ramenés à 254 pour éviter les débordements.

**Ex :**



**Décodage :**

La récupération du texte dans l’image s’opère de la manière suivante :

– Récupérer la matrice décrivant l’image.

– Remplacer un nombre pair par 0, un nombre impair par 1.

– Grouper les bits par groupe de 8.

– Convertir chaque octet en nombre décimal.

– Écrire les caractères correspondant aux codes ASCII obtenus.

**Exercice 5:**

Comment peut-on améliorer la méthode LSB ?

**Solution 5 :**

**L’avantage du LSB:**

* La taille du fichier n’est pas modifiée, puisque le message est encodé dans les parties peu ou pas utilisées du fichier.
* Méthode rapide et facile à mettre en œuvre.

**Inconvénients du LSB:**

* La perte du message lorsque des changements importants ont lieu sur le support, comme par exemple une rotation, ou un redimensionnement de l’image

**Amélioration:**

* Une amélioration du LSB consiste à introduire un paramètre aléatoire permettant de distribuer les bits de poids faible utilité. Les modifications n’auront pas lieu "uniquement" dans les premiers octets de l’image, mais seront au contraire répartis aléatoirement dans l’entièreté de l’image.