Faculté des sciences et technologies
Master 2 Sécurité des systèmes informatiques

Compte rendu de TP

Stéganographie

Mamadou Aliou DIALLO

Chargé du cours : Mr Habib BOUDJEMA

Introduction :

L'objectif du TP est de faire de la stéganographie. Ce qui consiste à cacher une information, un message dans un support sans qu'on puisse s'en rendre compte. On utilise particulièrement comme porteur les vidéos, les sons audio et les images. Dans notre cas nous allons travailler avec une image au format **.bmp** pour dissimuler un message afin de le rendre inintelligible auprès de toute personne n'ayant pas le droit.
Contrairement à la cryptographie qui fait en sorte que le message ne soit pas compris, la stéganographie repose sur le fait que le message ne soit pas compris.

1) Fonctionnement générale :

Une image est constituée d'un ensemble de pixels sous forme de matrice enregistré dans un fichier. Chaque pixel est constitué de trois octets (quatre si on compte l'Alpha) chacun représenté sur un octet. RVB (Rouge, Vert, Bleu).

Alpha (8 bits)	8 bits	8 bits	8 bits

Soit la clé **m** qu'on veut dissimuler dans notre image, on va la transformer sous forme d'octets.

m = 'M2SSI'' sera représenté par : **10011011100101010011101001110010 01**
Pour chaque pixel, on va modifier le Least Significant beat (bit de poids faible) par un bit du message **m** jusqu'à épuiser toutes les occurrences.

Si on choisit **m [3] = [0, 1, 0]**

Et un pixel **p (i, j)** =

1111111110110111110101100100011 1

On va obtenir le pixel **p' (i, j)** qui est égale à :

111111111011011**0**1101011**1**0100011**0**

2) Implémentation en java :

Nous allons implémenter en langage java la solution décrite au point ci-dessus.
Le projet se nomme **Stegano** et les classes se trouvent dans le package **hide.**

- **Classe HideMessage :**
  - La méthode **steganographie** qui retourne un entier sur le nombre d'octets dissimulé.
  - On charge l'image dans deux **BufferedReader** avec la classe **ImageIO** (image : à traiter et imageFinal : en sortie).
  - La clé en Ascii est enregistrée sous forme binaire dans un tableau.
  - On parcourt l'image et pour chaque pixel **image.getRGB (i, j)** stocké dans un tableau de 32, on récupère les trois premiers octets de la clé qu'on remplace aux positions :

    pixelNew [15] = tab [0]
    pixelNew [23] = tab [1]
    pixelNew [31] = tab [2]
  - On met à jour le pixel de l'image en sortie **imageFinal.Set (i, j, pixelNew)**
  - Au moment où les octets de la clé sont lus, on a terminé la dissimulation.
  - L'image est enregistrée dans un fichier et on retourne la taille de la clé nécessaire pour la reconstruction.

- **Classe ShowMessage :**
  - La méthode **deStegano** prend en paramètre la taille de clé dissimulé
  - Sur le même principe que la dissimulation on effectue l'opération dans le sens inverse.
  - L'image contenant le message caché est chargée et on boucle sur le contenu des pixels tant que la taille de la clé n'est pas atteinte.
  - A chaque étape on reconstitue la clé en prenant les octets des positions **15, 23 et 31** des pixels**.**
- **Classe UseFul :**
  - Contient les méthodes
    - **copyToTab3(char t1 [], char t2[], int pos)** qui copie trois éléments d'un tableau de char vers un tableau de taille 3.
    - **copyTo(String t1, char t2 [])** qui copie le contenu d'une chaine dans un tableau de char [].
- **Classe Main :**
  - La classe principale dans laquelle on lance les deux classes de notre programme
    - **HideMessage**
    - **ShowMessage**

3) Difficultés rencontrées :

J'ai eu à faire face à plusieurs difficultés d'ordre technique. Notamment à des erreurs de types **ArrayIndexOutOfBoundsException** dû la gestion des tailles des tableaux. Ce qui m'a amené à réfléchir à un algorithme plus efficace qui est capable de traiter un message de taille aussi grande que le nombre de pixels de l'image.

Conclusion :

La réalisation de ce TP a été pour moi une expérience passionnante. La stéganographie est utilisée dans l'industrie pour faire du watermarking : avantage ! Pas que car des personnes mal intentionnées peuvent utiliser ce procédé pour transmettre des informations sensibles (Employé d'une entreprise). Sa limitation dépend du procédé utilisé, dans notre cas une information dissimulée dans le format **.bmp** transmise et compressé dans un format **.jpeg** va être altérée.