Université F. Rabelais de Tours

IUP Blois

Master 2

Sécurité des réseaux :

Stéganographie et tatouage numérique

TRAVAUX PRATIQUES

Enseignant **Jean-Yves ANTOINE**

TP 1 : Etude de la robustesse des techniques de stéganographie

1. Présentation

Au cours de ce TP, nous allons étudier différent logiciels de stéganographie visuelle (i.e. d'insertion d'une information cachée dans une image) pour étudier la robustesse des différentes techniques utilisées à l'heure actuelle :

- insertion dans une image bitmap (méthode LSB)
- insertion dans une image au format GIF (modification de la palette de couleurs)
- insertion dans une image au format compressé (insertion dans l'espace de transformation fréquentiel)

Ce TP portera exclusivement sur l'étude de gratuiciels (*freeware*) que vous pourrez ainsi réutiliser en dehors des cours. Notez qu'il existe cependant de nombreux utilitaires équivalents commercialisés. Du point de vue des techniques de stéganographie utilisés, ces derniers diffèrent rarement des logiciels libres. Notre étude de robustesse restera de ce point de vue totalement pertinente. On notera toutefois que les gratuiciels ne permettent généralement pas un paramétrage des méthodes utilisées. Cette situation limitera parfois nos possibilités d'investigation. A titre d'exemple :

- on ne pourra pas faire varier le nombre de bits d'encodage dans la méthode LSB
- nous serons toujours obligés de donner une stego-clé (mot de passe) lors de l'encodage : nous ne pourrons donc pas faire de tatouage asymétrique (tatouage à clé publique) au cours du TP. Bien souvent, les utilitaires libres ou commercialisés pallient aux insuffisances actuelles de la stéganographie par un cryptage de la donnée cachée… qui facilite souvent sa détection !

Au cours de ce TP, nous nous placerons généralement dans une situation d'application de tatouage numérique (*digital copyright marking*) et non pas de stéganographie pure. La robustesse consistera donc avant tout à éviter l'effacement de la marque de copyright, même si nous étudierons parfois la « simple » détection de la présence d'une information cachée dans une image, objectif qui concerne plutôt la stéganographie pure.

Au cours du TP, nous utiliserons différentes images que vous trouverez sur la page WWW du cours.

2. Tatouage d'une image bitmap : *PUFF*

Dans cette partie, nous allons étudier l'utilitaire de stéganographie bitmap *Puff Steganografia & Authoring*, qui a été développé par Oliboni Cosimo. Celui-ci permet d'intégrer une marque de copyright ou toute autre information dans une image ou des fichiers son.

2.1. Marque textuelle de copyright

L'intégration d'une marque textuelle de copyright sous PUFF (icône ©→ de la fenêtre d'accueil est très simple) :

- définition de la marque sous la forme d'un texte d'au moins 16 caractères,
- sélection du (ou des) fichiers à marquer (*carrier files*),
- choix du répertoire de création des fichiers marqués (ceux-ci gardent le même nom que le fichier original !).

1. Choisissez une des images sur le serveur (par exemple `logo_R.bmp`) et insérez une marque de copyright textuel de votre choix. Cette marque est-elle détectable visuellement ?

2. Comparez la taille du fichier obtenu avec celle du fichier de couverture initial. Conclusion sur la

discression du processus de marquage ?

On se place maintenant dans une situation d'utilisation « normale » en tatouage numérique : on (police, justice ou propriétaire du copyright…) cherche à vérifier la présence de ce tatouage sur deux fichiers que l'on supposera suspects : le fichier original et, précisément, le fichier marqué. Dans ce type de configuration, la marque de copyright est bien entendue connue (marquage semi-public) de la personne qui réalise cette vérification.

3. Utilisez PUFF en mode de détection de copyright semi-public (icône →© de l'accueil) sur les deux fichiers incriminés : qu'observez-vous ?

4. Sans utiliser PUFF, on peut retrouver cette marque : à l'aide de l'utilitaire DOS `comp`, comparez ces deux fichiers. Retrouvez-vous la présence de la marque de copyright ? De quelle manière les bits ont-ils été insérés : à la suite ou de manière aléatoire dans le fichier de couverture ?

2.2. Stéganographie ou marquage digital par une image

La marque de copyright que nous voulons associer à notre document peut également être une image accompagnée par exemple d'un numéro de série. Dans ce cas, le mode d'utilisation de PUFF est également très simple (icône **H→**, pour *Hiding*, dans la fenêtre d'acceuil)

- choix du fichier (image mais également de tout autre type) à cacher,
- définition de la marque sous la forme d'un texte d'au moins 16 caractères,
- sélection du (ou des) fichiers de couverture (*carrier files*) : PUFF vérifie ici que la taille du fichier de couverture (ou à marquer) est suffisamment importante pour permettre une intégration non détectable (un seul bit sur les 8 bits de codage).
- choix du répertoire de création des fichiers marqués (ceux-ci gardent le même nom que le fichier original, il est donc important que vous précisiez un autre répertoire).

1. Tentez maintenant d'intégrer le logo du Master SIR (`logo_MSIR.bmp`) dans votre image originale. Cette marque est-elle détectable visuellement ?

2. Appliquez PUFF en mode inverse sur le fichier marqué et le fichier orginal (là encore, on se place en situation de tatouage semi-public : la stego-clé est connue). Qu'observez-vous ?

3. Essayez maintenant d'intégrer (puis d'extraire) un autre type de document, par exemple un fichier Word : quelle quantité maximale d'information pouvez-vous intégrer (en pourcentage de l'image de couverture) pouvez-vous intégrer dans le fichier `logo_R.bmp` avec LSB ?

2.3. Etude de robustesse du marquage LSB

Nous allons maintenant nous intéresser à la robustesse de la méthode LSB. On rappelle que la robustesse caractérise avant tout le fait que le tatouage digital reste perceptible après des manipulations de l'image volontaires (attaque active) ou involontaire.

1. Reprenez l'image que vous venez de créer avec insertion du logo du Master dans celui de l'Université.

2. A l'aide de l'utilitaire graphique de votre choix (*Paint, Microsoft Photo Editor*…), créer un ensemble de nouvelles images obtenues à partir des modifications suivantes de la stego image :

- rotation de l'image (même imperceptible) puis retour à l'original,
- inversion des couleurs (image en négatif) puis retour à l'original,
- transformation au format TIF puis retour au format BMP,
- transformation au format compressé GIF puis retour au format BMP,
- transformation au format compressé JPEG puis retour au format BMP.

Vérifiez à chaque fois si le tatouage est resté présent. Ces résultats étaient-ils prévisibles ?

2.4. Attaque passive : détection d'information cachée

Les problèmes de robustesse que nous venons de détecter peuvent être le fait de manipulations involontaires. A l'inverse, chercher à détecter la présence d'information cachée dans un fichier ne peut être que le fait d'une attaque délibérée. Là encore, les techniques LSB sont peu fiables face à ces attaques passives. Celles-ci se traduisent en effet par une signature statistique claire : après insertion

des données masquées, l'information sur les bits de poids faibles se rapproche d'une distribution gaussienne. Nous allons étudier plusieurs techniques pour détecter cet artefact statistique.

1. Prenez tout d'abord l'image originelle `logo_R.bmp` ainsi que sa stego image obtenue par insertion du logo du Master. Comparer leur taille : conclusion ? Compressez ensuite ces deux fichiers avec l'utilitaire WinZip. Comparez la taille des fichiers ainsi compressés. Qu'observez-vous ? Pouvez-vous expliquer ce résultat ?

Cette détection par compression suffit à éveiller l'attention dans le cas d'une insertion importante de données. Pour cela, il faut toutefois disposer de l'image originale. Une étude statistique pure de la distribution des bits de poids faible de la seule stego image suffit pourtant à détecter une présence suspecte. Le gratuiciel `Chi-square`, développé par Gillermito, repose sur la comparaison statistique de la distribution des trois LSB avec une distribution aléatoire, suivant la technique des paires de valeurs étudiée en cours [Westfeld, Pfitzmann, 2000].

L'utilisation de `Chi-square` est simplissime : vous choisissez le fichier que vous voulez analyser et le logiciel vous fournit l'analyse en $\chi 2$ de l'image versus une distribution aléatoire.

2. Reprenez les istego images réalisées à partir de l'image `logo_R.bmp` en inserant une petite marque de copyright, puis une image de taille importante (logo Master). Pour chacune de ces images, ainsi que pour l'image originelle, visualisez la courbe du $\chi 2$ donnée l'utilitaire `Chi-square`. L'insertion d'information est-elle détectable ?

3. Nous avons vu que des manipulations relativement simple des stego images suffisent pour effacer une marque de copyright. Peut-on cependant tracer cet effacement : en étudiant un des fichiers crées au §2.3, pouvez-vous dire si l'analyse statistique révèle encore la présence d'une opération antérieure de tatouage ?

2.5. Attaque active : surimpression de marque

Comme nous l'avons vu, une premier méthode d'attaque active consiste à effacer le tatouage présent dans l'image. Une autre technique consiste à rajouter une nouvelle marque sur la stego-image, soit pour effacer la première, soit simplement pour qu'il ne soit plus possible de détecter quel était le premier propriétaire de l'objet originel.

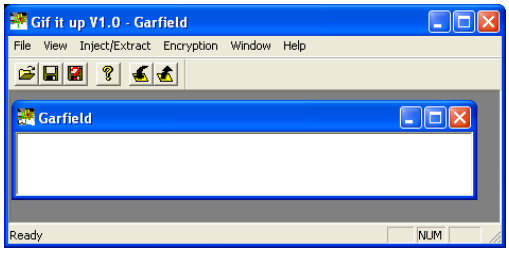
1. Reprenez une stego image contenant un tatouage important (le logo du Master par exemple) et insérez à l'aide de PUFF une simple marque de copyright textuelle très limitée (16 caractères).

2. Lancez PUFF en analyse sur cette nouvelle stego image : qu'observez-vous ?

3. Tatouage d'une image GIF : *GifItUp*

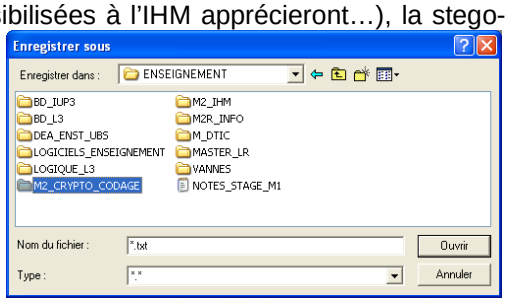
Le format GIF est un format compressé qui ne travaille pas dans l'espace des fréquences comme JPEG, mais s'appuie simplement sur une palette de couleur réduite. Une image GIF n'est donc rien d'autres qu'une image bitmap à laquelle est ajoutée une entête décrivant la palette de couleur utilisée. Les logiciels de steganographie travaillant sur le format GIF auront donc des faiblesses assez comparables aux méthodes LSB. Pour le vérifier, nous allons utiliser cette fois le gratuiciel *Git It Up* qui travaille sur la palette de couleur de l'image GIF de couverture.

Gif It Up autorise l'insertion de tout type d'information dans une image GIF, du moment que cette information cachée est intégrée dans un fichier. Pour l'utiliser en marquage textuel (*copyright watermarking* ou *fingerprinting*), vous devez donc créer un fichier ASCII contenant cette marque. L'insertion du fichier caché suit la procédure ci-dessous :



- On ouvre tout d'abord le fichier de couverture (icône de la barre d'outils ou menu File)
- On « injecte » ensuite le fichier caché (icône ou menu `Inject/Extract`) . Le logiciel peut vous demander d'étendre la palette de couleur au cas où moins de 256 couleurs sont utilisées par l'image de couverture. Cette extension augmentera la taille du fichier mais permet de cacher plus d'information : tout dépend donc de votre contexte d'utilisation.

- Après cette étape sans feedback (les personnes sensibilisées à l'IHM apprécieront…), la stego-image doit être sauvegardée à l'aide du menu File (option Save ou Save As) ou directement à l'aide de la barre d'outils.



  Pour extraire une information cachée à partir d'une stego image, on réalise la procédure inverse d'extraction après avoir ouvert le stego fichier correspondant. Une boite de dialogue vous demande cette fois sous quel nom sauvegarder le fichier cacher. On remarquera que *Gif It Up* n'analyse pas le contenu du fichier caché : c'est à vous de savoir quel est le type (ASCII, image,…) du fichier caché.

1. Prenez l'image originale chambord`.gif` et intégrez-y une petite image (logo Master par exemple) : cette insertion est-elle détectable visuellement ?
2. Comparez la taille de la stego image obtenue avec celle de l'image de couverture. L'insertion est-elle détectable ? Expliquez ce qui se passe sachant que le format GIF fait l'objet d'une compression dans tous les cas de figure (analyse des pixels redondants).

4. **Etude de robustesse des techniques de modification de palette GIF** — Testez maintenant la robustesse de cette technique d'insertion par rapport aux transformations usuelles suivantes :
   - rotation de l'image (même imperceptible), puis retour à l'original,
   - inversion des couleurs (image en négatif) puis retour à l'image initiale,
   - transformation au format BMP puis retour au format GIF,
   - transformation au format TIF puis retour au format GIF,
   - transformation au format compressé JPEG puis retour au format GIF.

   Ces observations étaient-elles prévisibles ? Comparez cette robustesse à celle des méthodes précédentes.

5. Attaque passive : détection d'information cachée

Les techniques statistiques de détection passives reprennent, en les adaptant, celles vues avec les images bitmap. Nous ne les étudierons donc pas ici.

6. Attaque active : surimpression de marque

Nous allons enfin étudier la robustesse de la stéganographie par modification de palette en tentant de surimposer un tatouage à une image déjà marquée.

1. Réalisez une stego image contenant un tatouage important (le logo du Master par exemple) et insérer à l'aide de *Gif It Up* une marque de copyright textuelle très limitée (16 caractères).
2. Lancez *Gif It Up* en analyse sur cette nouvelle stego image : qu'observez-vous ?

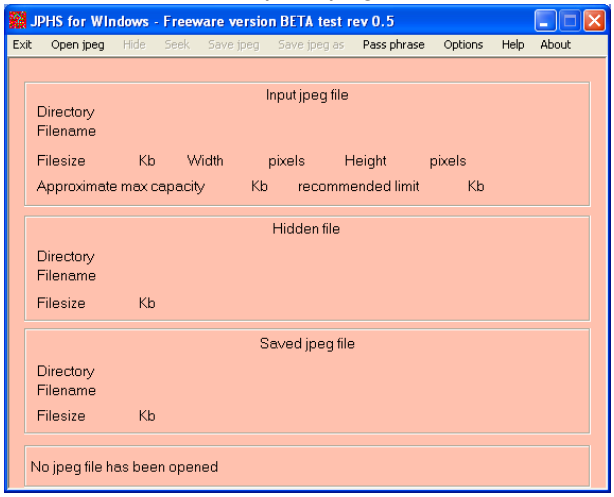
4. Tatouage dans l'espace des fréquences : images JPEG

Pour terminer cette étude des différentes techniques de stéganographie, nous allons nous intéresser aux méthodes de tatouage dans l'espace fréquentiel (coefficients DCT de la compression JPEG, par exemple). Ces techniques sont très en vogue car elle s'avèrent plus robustes que celles étudiées précédemment et résistent aux techniques de compression actuelles. Il n'en reste pas moins que *F5* et *Outguess*, qui sont considérés comme les utilitaires les plus sûrs en matière de « stéganographie fréquentielle », ne résiste pas à une attaque systématique de type StirMark. C'est ce que nous allons voir dans cette dernière partie en nous intéressant à *JPHS*, un gratuiciel développé par Allan Latham.

4.1. Tatouage numérique avec JPHS

Comme *Gif It Up*, *JPHS* n'autorise que l'insertion d'un fichier caché dans une image de couverture. Si vous voulez intégrer une simple marque de copyright ou un numéro de série, il vous faudra donc encapsuler cette information dans un fichier ASCII.

1. On ouvre tout d'abord le fichier de couverture à l'aide du menu `OpenJpeg`.
2. On sélectionne ensuite le fichier à cacher à l'aide du menu `Hide`. Une fenêtre de dialogue apparaît, qui vous demande de préciser une phrase de code. *JPHS* utilise ce code comme stego-clé. Vous pouvez néanmoins ne pas utiliser de code pour réaliser un marquage à clé publique : pour cela, il vous suffit de ne rien préciser dans la boite de dialogue. C'est ce mode que nous utiliserons au cours de ce TP. Après sélection, le fichier est intégré dans l'image de couverture.
3. Il reste à sauver la stego-image ainsi obtenue à l'aide du menu `Save Jpeg` (original écrasé) ou `Save Jpeg As`.



La récupération suit le même mode opération, à la différence qu'on utilise le menu `Seek` à la place de Hide. En cas de marquage à clé publique, on ne précisera aucune phrase de code.

4.2. Limites de la stéganographie fréquentielle : JPHS

A la manière des études précédentes, il vous est demandé de mener une étude expérimentale de *JPHS* pour décrire les caractéristiques et limitations de cet utilitaire en terme de discrétion, capacité de masquage et robustesse.

4.3. Attaque passive : détection d'information cachée

Les techniques statistiques de détection passives reprennent, en les adaptant au domaine fréquenhtiel, celles vues avec les images bitmap. Nous ne les étudierons donc pas ici. En pratique, chaque utilitaire de tatouage dans l'espace des fréquences rajoute des petits raffinements (cryptage, stégo-clé, méthode d'insertion LSB fréquentielle) qui lui est propre. Les logiciels de stéganalyse sont alors obligés de développer des techniques d'attaques spécifique à chacun, par delà les principes généraux de l'attaque statistique. Au final, tous les logiciels actuels de stéganographie commercialisés sont attaquables.