Criptografía y Seguridad *(72.44)*

TRABAJO PRÁCTICO: ESTEGANOGRAFÍA

1. Objetivos

- Introducirlos en el campo de la esteganografía y sus aplicaciones.
- Experimentar con métodos de ocultamiento de información en archivos bmp, analizando ventajas y desventajas de cada uno.

2. Introducción

La ***esteganografía*** (del griego στεγανοζ *steganos, encubierto u oculto* y γραπηοζ *graphos, escritura*) es la ciencia que se ocupa de la manera de **ocultar** un mensaje.

La existencia de un mensaje u objeto es ocultada dentro de otro, llamado **portador**. El objetivo es proteger información sensible, pero a diferencia de la criptografía que hace ininteligible dicha información, la esteganografía logra que la información pase completamente desapercibida al ocultar su existencia misma.

La criptografía y la esteganografía se complementan. Un mensaje cifrado mediante algoritmos criptográficos puede ser advertido por un intruso. Un mensaje cifrado que, además, ha sido ocultado mediante algún método de esteganografía, tiene un nivel de seguridad mucho mayor ya que los intrusos no pueden detectar su existencia. Y si por algún motivo un intruso detectara la existencia del mensaje, encontrarían la información cifrada.

La estenografía tiene un origen muy antiguo. Ya Heródoto en el año 440 aC narra la historia de un mensaje escrito en una tablilla que es cubierto con cera para pasar desapercibido ante el enemigo. El mensaje puede así llegar a su destino, siendo develado por sus receptores al quitar la cera.

En la era digital, el interés se renueva por sus múltiples aplicaciones, entre otras:

- Protección de derechos de autor (watermarking y fingerprinting)
- Técnicas de anonimato
- Voto electrónico

Los algoritmos de ocultamiento dependen del archivo portador, ya que la alteración del mismo de manera profunda puede despertar sospechas respecto de la existencia de un mensaje. En general se eligen como archivos portadores algún tipo de archivo multimedial: imagen, video o archivo de audio. Pero también existen otras posibilidades como archivos zip o archivos ejecutables.

El ***estegoanálisis*** se ocupa de estudiar métodos para detectar si un archivo ha sido ocultado en otro. Este campo de estudio está teniendo un desarrollo muy importante especialmente por agencias de investigación criminales debido a los alcances que la esteganografía con malos propósitos puede llegar a tener (por ejemplo, ataques terroristas, pedofilia, etc)

3. Consigna

- Realizar un programa **stegobmp** en lenguaje C o Java que efectúe las siguientes operaciones:
  - Oculte un archivo cualquiera en un archivo .bmp, mediante un método de esteganografiado elegido, con o sin password.
  - Descubra un archivo oculto en un archivo .bmp que haya sido previamente esteganografiado con uno de los métodos provistos.
- Estegoanalice un archivo .bmp para determinar si tiene un archivo incrustado, con qué algoritmo y lo extraiga correctamente.

4. Archivos .BMP

El formato BMP es un formato de archivos de imagen bastante simple. Consta de dos partes:

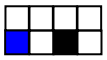
1. Encabezado → de 54 bytes
2. Cuerpo → de tamaño variable.

El encabezado contiene información acerca del archivo: tamaño de archivo, ancho de imagen, alto de imagen, bits por píxel, si está comprimido, etc

IMPORTANTE: Considerar la versión V3 de archivos BMP. Es la más común. No hace falta considerar otras versiones que puedan tener otro tamaño y datos de encabezados.

En el cuerpo del archivo bmp, están los bits que definen la imagen propiamente dicha. La imagen se lee de abajo hacia arriba y de izquierda a derecha. Si la imagen es de 24 bits por píxel, la distribución

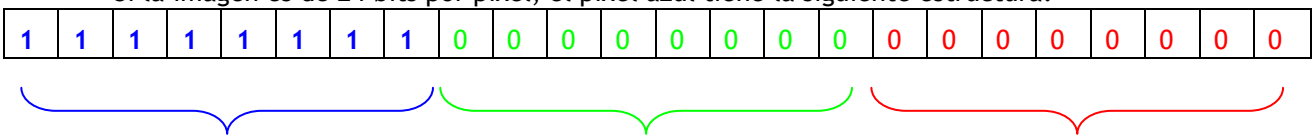
es: 8 primeros bits para azul, 8 bits para verde, y 8 bits para rojo.



Ejemplo: La imagen de la izquierda representa un bmp de 4 pixeles por 2 pixeles.

El primer píxel leido es el azul, luego el blanco, luego el negro.

Si la imagen es de 24 bits por píxel, el píxel azul tiene la siguiente estructura:



En este trabajo práctico sólo interesa usar el cuerpo del archivo bmp como portador, para ocultar allí la información a esteganografiar. Por este motivo, sólo se utilizarán archivos .bmp que tengan las siguientes características:

- Imagen de 24 bits por pixel. Esto es, por cada pixel, se tienen 3 bytes donde ocultar información.
- Imagen bmp sin compresión. Para ello controlar el parámetro de compresión del encabezado.

5. Detalles del programa stegobmp

5.1. Ocultamiento de un archivo en un .bmp.

El programa debe recibir como parámetros:

- **`-embed`**

  Indica que se va a ocultar información.

- **`-in`** file

  Archivo que se va a ocultar.

- **`-p`** bitmapfile

  Archivo bmp que será el portador.

- **`-out`** bitmapfile

  Archivo bmp de salida, es decir, el archivo bitmapfile con la información de file incrustada.

- **`-steg <LSB1 | LSB4 | LSBE>`**

  algoritmo de esteganografiado: LSB de 1bit, LSB de 4 bits, LSB Enhanced

Y los siguientes parámetros opcionales:

- **`-a <aes128 | aes192 | aes256 | des>`**
- **`-m <ecb | cfb | ofb | cbc>`**
- **`-pass password`** (password de encripcion)

Ejemplo 1:

Esteganografiar el archivo de texto “mensaje1.txt” en el archivo portador “imagen1.bmp” obteniendo un archivo “imagenmas1 .bmp” mediante el algoritmo LSB Enhanced, con encripción DES en modo CBC con password “oculto”

```
$stegobmp -embed –in “mensaje1.txt” –p “imagen1.bmp” -out “imagenmas1.bmp” –steg LSBE –a des –m cbc -pass “oculto”
```

Ejemplo 2:

Esteganografiar el archivo de imagen “mensaje1.txt” en el archivo portador “imagen1 .bmp” obteniendo un archivo “imagenmas1.bmp” mediante el algoritmo LSB Enhanced, **sin encripción**

```
$stegobmp -embed –in “mensaje1.txt” –p “imagen1.bmp” -out “imagenmas1.bmp” –steg LSBE
```

Importante:

No se puede encriptar/desencriptar sin **password**. Si este dato no está, sólo se esteganografia.

Son válidas en cambio las siguientes opciones:

- indicar **algoritmo** y **password** pero no modo: Se asume CBC por default.
- Indicar **modo** y **password** pero no algoritmo: Se asume aes128 por default.
- Indicar **sólo password:** Se asume algoritmo aes128 en modo CBC por default.

5.2.Extraer de un archivo .bmp un archivo oculto.

- **`-extract`**

  Indica que se va a extraer información.

- **`-p bitmapfile`**

  Archivo bmp portador

- **`-out file`**

  Archivo de salida obtenido

- **`-steg <LSB1 | LSB4 | LSBE>`**

  algoritmo de esteganografiado: LSB de 1bit, LSB de 4 bits, LSB Enhanced

Y los siguientes parámetros opcionales:

- **`-a <aes128 | aes192 | aes256 | des>`**
- **`-m <ecb | cfb | ofb | cbc>`**
- **`-pass password (password de encripcion)`**

Ejemplo:

Extraer el archivo de texto “mensaje1.txt” del archivo portador “imagenmas1.bmp” ocultado mediante el algoritmo LSB Enhanced, con encripción DES en modo CBC con password “oculto”

```
$stegobmp –extract –p “imagenmas1 .bmp” -out “mensaje1” –steg LSBE –a des –m cbc -pass “oculto”
```

5.3.Algoritmos de Esteganografiado.

Ocultamiento sin encripción:
Antes de ocultar el archivo propiamente dicho, con cualquiera de los algoritmos, ocultar su tamaño.
Después de ocultar el tamaño y el archivo propiamente dicho, con cualquiera de los algoritmos, ocultar su extensión (“.png”, “.jpg”, “.txt”, “.html”, etc)
La extensión debe comenzar con ‘.’ Y terminar con ‘\0’.

Es decir, se esteganografía:

Tamaño real || datos archivo || extensión

Y el total de datos a esteganografiar es:

4 (del tamaño) + longitud archivo + e (extensión)

Siempre se sabe que los primeros 4 bytes (DWORD size) corresponden al tamaño de archivo.
Siempre se sabe que después de los n bytes del archivo viene un punto y los n ascii de la extensión, terminando en ‘\0’.

Ocultamiento con encripción:
Si se eligió encriptar antes, se procede de la siguiente manera:

- Se obtiene la secuencia de bytes correspondiente a **Tamaño real || datos archivo || extensión.**
- Dicha secuencia se encripta con el algoritmo , modo y password.
- Se esteganografia el tamaño del cifrado y a continuación la secuencia cifrada.

Es decir, se esteganografía:

Tamaño cifrado || encripcion(tamaño real || datos archivo || extensión**)**

Y el total de datos a esteganografiar es:

4 (del tamaño del cifrado) + tamaño del cifrado

Para extraer, siempre se sabe que los primeros 4 bytes (DWORD size) corresponden al tamaño de cifrado. Con dicho tamaño, y algoritmo modo y password que se envian por argumento al programa se descifra. Una vez hecha la desencripción se obtiene el tamaño real del archivo, se lee esa cantidad de bytes y se toma la extensión (hasta el '\0').

Ejemplo:

Si el tamaño real de "saco.txt" es 414 bytes, y se eligio encripcion en Des Cbc.
Entonces se encripta:

414 || datos archivo || ".txt\0"

Que da un total de 4 + 414 + 5 = 423 bytes.
No es múltiplo de bloque así que requiere padding. Por lo tanto se encripta 423 + padding.
Luego, se esteganografia:

(423+padding) || encripcion(414||datos archivo||".txt\0")

Es decir, se ocultan **4 (del tamaño del cifrado) + (423 + padding)** bytes.

1. LSB1

Inserción en el bit menos significativo (Least Significant Bit Insertion - LSB)

Es el método más simple y consiste en insertar información sustituyendo el bit menos significativo de cada byte del archivo portador por un bit del mensaje.

En el caso de archivos de imagen ".bmp" consideraremos que cada bit del mensaje se inserta en el bit menos significativo de la **componente** (cada pixel tiene un componente rojo, un componente verde y un componente azul, cada uno de 8bits).

Como se observa, el primer bit del mensaje a ocultar se inserta en el bit menos significativo de la primer componente del primer pixel, el segundo bit del mensaje a ocultar se inserta en el bit menos significativo de la segunda componente del primer pixel, el siguiente, en la tercer componente del primer pixel. Entonces para guardar un byte, se requieren 2 pixeles completos y 2 componentes del siguiente pixel.

Importante: Si el archivo bmp no puede albergar el archivo a ocultar completo, deberá mostrarse un mensaje de error, en el que se indique cuál es la capacidad máxima del archivo bmp portador.

2. LSB4

Inserción en los cuatro bits menos significativos

Es una variante del anterior, donde 4 bits del mensaje se ocultan en los 4 bits menos significativos de la componente del pixel que corresponda.

Importante: Si el archivo bmp no puede albergar el archivo a ocultar completo, deberá mostrarse un mensaje de error, en el que se indique cuál es la capacidad máxima del archivo bmp portador.

3. LSB Enhanced

Es una variante de los anteriores propuesta por Sridevi, Damodaram y Narasimham en el documento "**Efficient method of audio steganography by modified LSB algorithm and strong encryption key with enhanced security**". Esta variante considera como portador un archivo de audio (.wav). En nuestro caso, se adaptará la idea a archivos de imagen (.bmp)

Dicho documento se encuentra en:
http://www.jatit.org/volumes/research-papers/Vol5No6/15Vol5No6.pdf

Importante: No hace falta escribir la marca de fin de archivo que propone el documento, ya que estamos guardando siempre el tamaño del archivo en los primeros 4 bytes del mensaje.

Importante: Si el archivo bmp no puede albergar el archivo a ocultar completo, deberá mostrarse un mensaje de error, en el que se indique cuál es la capacidad máxima del archivo bmp portador.

5.4. Otras consideraciones.

- Si el archivo a ocultar requiriera padding usar el padding por defecto de openssl que es PKCS5.
- El archivo que se encripta es el que se quiere ocultar. Se encripta completo. Es decir, aún si

fuera un bmp, se encripta desde el byte 0 hasta el último, cabecera incluida.

- Para la generación de clave a partir de una password, asumir que la función de hash usada es md5, y que no se usa SALT.
- Para modos de feedback considerar una cantidad de 8 bits
- Los bmps utilizados deben ser de **24 bits** (8bits para rojo, 8bits para verde y 8bits para azul).
- Los bmps utilizados **no deben** tener compresión.

6. Estegoanálisis

La cátedra proveerá de 4 archivos con información oculta. De ellos, uno de los archivos contendrá un archivo ocultado mediante LSB1, otro mediante LSB4 y otro mediante LSBE. Un cuarto archivo ocultará información de otra forma que habrá que descubrir.

En los que están esteganografiados con los métodos LSB1,LSB4, LSBE, se sigue el formato especificado previamente:

tamaño + archivo + extensión

Habrá que obtener los archivos ocultos (estegoanálisis) descubriendo de qué manera fueron ocultados.

En general, el análisis que se puede hacer es:

- por conocimiento de archivo portador
- por repetición de archivo portador
- estadístico
- de tamaño de los archivos
- etc.

IMPORTANTE:

Se recomienda, para este análisis, usar algún editor de archivos binarios (hex editor neo o similar) que permita hacer comparaciones de los archivos.

7. Cuestiones a analizar

Una vez realizado el programa, se resolverán las siguientes **9** cuestiones:

1. Para la implementación del programa stegobmp se pide que la ocultación comience en el primer componente del primer pixel. ¿Sería mejor empezar en otra ubicación? ¿Por qué?
2. ¿Qué ventajas podría tener ocultar siempre en una misma componente? Por ejemplo, siempre en el bit menos significativo de la componente azul.
3. Esteganografiar un mismo archivo en un .bmp con cada uno de los tres algoritmos, y comparar los resultados obtenidos. Hacer un cuadro comparativo de los tres algoritmos estableciendo ventajas y desventajas.
4. Para la implementación del programa stegobmp se pide que la extensión del archivo se oculte después del contenido completo del archivo. ¿por qué no conviene ponerla al comienzo, después del tamaño de archivo?
5. Explicar detalladamente el procedimiento realizado para descubrir qué se había ocultado en cada archivo y de qué modo.
6. ¿Qué se encontró en cada archivo?
7. Algunos mensajes ocultos tenían, a su vez, otros mensajes ocultos. Indica cuál era ese mensaje y cómo se había ocultado.
8. Uno de los archivos ocultos era una porción de un video, donde se ve ejemplificado una manera de ocultar información ¿cuál fue el portador?
9. ¿De qué se trató el método de estenografiado que no era LSB? ¿Es un método eficaz? ¿Por qué?
10. ¿Qué mejoras o futuras extensiones harías al programa stegobmp?

Se recomienda una lectura del material de consulta recomendado para hacer un análisis apropiado.

8. Organización de los grupos

El trabajo será realizado en grupos de máximo 4 integrantes.

Deberán comunicar lo antes posible cómo están conformados los grupos.

9. Entrega

La fecha de entrega es el día **25 de junio**

Cada grupo entregará el ejecutable y el código en C o Java, junto con los archivos para obtener el ejecutable y la documentación correspondiente al uso del programa, a través del svn, comunicando el número de revision en un mail que se enviará a mroig@itba.edu.ar y a contiver@itba.edu.ar

Además presentarán un informe **impreso** con la solución correspondiente al estegoanálisis de los archivos que se le entregarán oportunamente al grupo y con el análisis de las cuestiones planteadas en el punto 7.

10. Material de consulta recomendado

Hay mucho material en internet sobre el tema.

Es obligatorio leer:
Sobre el método que para este TP denominamos LSBE, Sridevi, Damodaram y Narasimham **"Efficient method of audio steganography by modified LSB algorithm and strong encryption key with enhanced security"**. Disponible en: **http://www.jatit.org/volumes/research-papers/Vol5No6/15Vol5No6.pdf**

Sobre **archivos bmp**: http://msdn.microsoft.com/en-us/library/windows/desktop/dd183374(v=vs.85).aspx

Otros documentos que son útiles para la comprensión del tema o para contestar las preguntas:

- Cummings, Jonathan y otros: **Steganography and Digital Watermarking**. Disponible en: http://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.pdf
- Isaza, Gustavo A. y otros: **Análisis de técnicas esteganográficas y estegoanálisis en canales encubiertos, imágenes y archivos de sonido**. En http://vector.ucaldas.edu.co/downloads/Vector1_3.pdf
- Gómez Cárdenas, Roberto: **Esteganografía**. Disponible en: http://www.cryptomex.org/SlidesCripto/Estegano.pdf
- Johnson, Neil F. y Jajodia, Sushil : **Exploring Steganography. Seeing the Unseen**. Disponible en: http://www.creangel.com/papers/steganografia.pdf
- Gupta, Shilpa; Gujral, Geeta y Aggarwal, Neha: **Enhanced Least Significant Bit algorithm For Image Steganography**. Disponible en: www.ijcem.org/papers072012/ijcem_072012_08.pdf