TRABAJO PRÁCTICO: ESTEGANOGRAFÍA

INFORME

72.44 - Criptografía y Seguridad

Grupo 20
Felipe Oliver, 58439
Santiago Reyes, 58148
Manuel Luque Meijide, 57386

Índice

1. Introducción

El siguiente informe es de esteganografía en imágenes para la materia Criptografía y Seguridad.

1.1 Objetivos

El objetivo del trabajo fue el desarrollo de un sistema informático capaz de realizar técnicas de estenografía y estegoanálisis sobre distintos archivos, contando con la capacidad de encriptar/desencriptar la información durante el proceso.

El presente informe busca estructurar y presentar los conocimientos, las pruebas, los resultados y los análisis obtenidos durante el desarrollo del sistema.

1.2 Esteganografía

La esteganografía 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, encontrarán la información cifrada.

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).

2. Cuestiones a Analizar

2.1 Sobre el documento **“An Improved Inverted LSB Image Steganography”**

a. Organización del documento

En cuanto al documento, su organización es bien estructurada. Está separada en introducción, implementación, resultados y análisis, conclusión y finalmente referencias lo cual parece apropiado. El texto en dos columnas si dificulta un poco el seguimiento pero no lo consideraría un problema.

Por su parte la introducción deja sin lugar a dudas de que se trata la estenografía, mostrando sus diferencias con la criptografía. La implementación, fue elocuente en como describió el algoritmo común y como introdujo al LSBI.

Luego la sección de resultados y análisis estaba regularmente explicada pero hubiese sido mejor si los gráficos estaban intercalados con el texto para una mejor comprensión del mismo.

b. La descripción del algoritmo

En cuanto al algoritmo, logra comunicar la idea de una manera relativamente clara, pero hay algunas cuestiones que quedan ambiguas a la hora de implementarlo. Podría aclarar un poco mejor el paso a paso del algoritmo y posiblemente mostrar un ejemplo que consista de un caso donde se debe invertir y otro que no.

El LSBI es más complejo que el LSB común, y un conocimiento previo sobre el sistema binario es necesario.

c. La notación utilizada, ¿es clara? ¿Hay algún error o contradicción?

No se detectó error o contradicción alguna. La notación utilizada en su gran mayoría es clara pero una aclaración previa explicando qué significa cada una sería más claro.

2.2 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.

Para analizar el comportamiento de nuestro sistema, se va a estenografiar un mismo archivo (secret.jpg) en un en otro (lado.bmp) utilizando 3 algoritmos diferentes: LSB1, LSB4 y LSBI.
El archivo portador lado.bmp es el provisto por la cátedra.

Resultados

En la *tabla 1* se muestran los resultados obtenidos por cada uno de los 3 algoritmos de estenografiado, detallando tanto ventajas como desventajas de su utilización.

Algoritmo	Ventajas	Desventajas
LSB1	Menor distorsión de la imagen final.	Requiere un mayor tamaño del portador (solo se utiliza un bit por pixel)
LSB4	Reduce el tamaño requerido para el portador (se aprovechan más bits)	Más distorsión en la imagen final.
LSBI	Menor distorsión que lsb1 ya que se reemplazan menos pixeles	Hay que utilizar 4 bytes más para almacenar el patrón

Tabla 1: tabla comparativa de los algoritmos

2.3 Explicar detalladamente el procedimiento realizado para descubrir qué se había ocultado en cada archivo y de qué modo. Indicar qué se encontró en cada archivo.

Condiciones Iniciales

Se conoce que los 4 archivos cuentan con algún tipo de información oculta. Además se sabe que hay un archivo utilizando LSB1, otro LSB4, otro LSBI y el último no utiliza una estenografía de tipo LSB.

Análisis

Para cada archivo se intenta abrirlo con todos los métodos de estenografía que no se hayan utilizado exitosamente todavía. Primero se prueba levantar todos los archivos sin ningún tipo de encripción.

buenosaires.bmp

LSB1

./stegobmp --extract -p grupo20/buenosaires.bmp -o extracted_buenosaires --steg=LSB1

Utilizando LSB1 no se tuvo éxito.

LSB14

./stegobmp --extract -p grupo20/buenosaires.bmp -o extracted_buenosaires --steg=LSB4

Utilizando LSB4 sí se tuvo éxito y se encontró la siguiente imagen escondida:

Figura 1: extracted_buenosaires.png

buenosaires0.bmp

LSB1

./stegobmp --extract -p grupo20/buenosaires0.bmp -o extracted_buenosaires0 --steg=LSB1

Utilizando LSB1 no se tuvo éxito.

LSBI

./stegobmp --extract -p grupo20/buenosaires0.bmp -o extracted_buenosaires0 --steg=LSBI

Utilizando LSBI no se tuvo éxito. Es posible que el archivo se encuentre encriptado o corresponda al que no utiliza un método LSB.

topgun0.bmp

LSB1

./stegobmp --extract -p grupo20/topgun0.bmp -o extracted_topgun0 --steg=LSB1

Utilizando LSB1 no se tuvo éxito.

LSBI

./stegobmp --extract -p grupo20/topgun0.bmp -o extracted_topgun0 --steg=LSBI

Utilizando LSBI se obtuvo un archivo **extracted_topgun0.pdf** con el siguiente mensaje:

“al .png cambiarle la extension por .zip y descomprimir”

Parecería ser que se refiere al archivo que se obtuvo a partir de la imagen buenosaires.bmp (la imágen del buscaminas).

extracted_buenosaires0.png

Siguiendo las instrucciones del pdf anterior, se ejecuta lo siguiente:
cp extracted_buenosaires.png bsas.zip && unzip bsas.zip

y de esta manera se obtiene un archivo **sol20.txt** que dice lo siguiente:

“cada mina es un 1. cada fila forma una letra. Los ascii de las letras empiezan todos en 01. Asi encontraras el algoritmo que tiene clave de 128 bits y el modo La password esta en otro archivo Con algoritmo, modo y password hay un .wmv encriptado y oculto.”

Buscaminas a binario								**ASCII**
0	1	0	0	0	0	0	1	***A***
0	1	1	0	0	1	0	1	***E***
0	1	1	1	0	0	1	1	***S***
0	1	0	0	0	1	0	1	***E***
0	1	1	0	0	0	1	1	***C***
0	1	1	0	0	0	1	0	***B***

Siguiendo las instrucciones obtenemos que hay un archivo que está encriptado con AES128 y modo ECB. Además nos dicen que este archivo es un video (wmv).

Teniendo en cuenta que el archivo frozen.bmp es **significativamente** más grande que los demás, se sospecha que este es el archivo encriptado que contiene el video. Por lo tanto es probable que el archivo **buenosaires0.bmp** sea el que no contiene un mensaje utilizando un método LSB y por lo tanto sería el que contiene la contraseña como indica **sol20.txt.**

buenosaires0.bmp (devuelta)

Para investigar qué podría estar pasando en este archivo se utilizó binwalk para ver la entropía del archivo y generar un gráfico correspondiente:

binwalk -E -J grupo20/buenosaires0.bmp

Con esto obtenemos el siguiente gráfico:

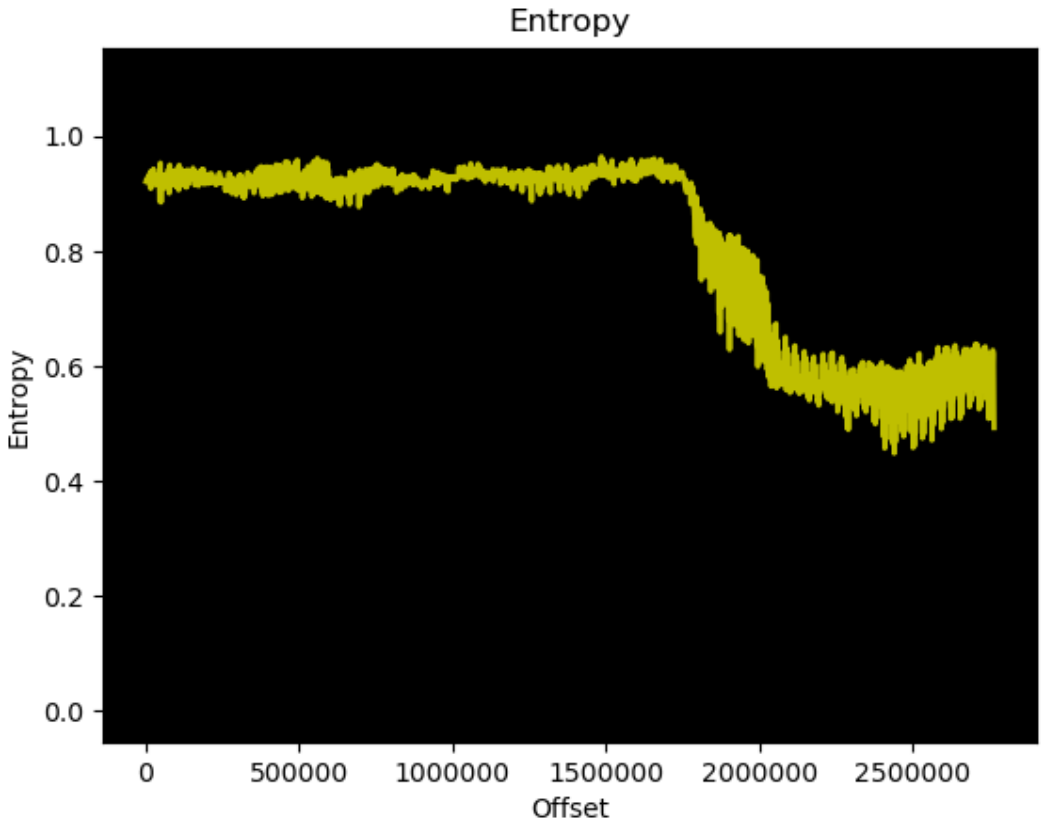

Figura 2: gráfico de entropía de buenosaires0.bmp

Parecería indicar que puede llegar a haber información al final del archivo.

Abriéndolo con un hex editor se encuentra lo siguiente:

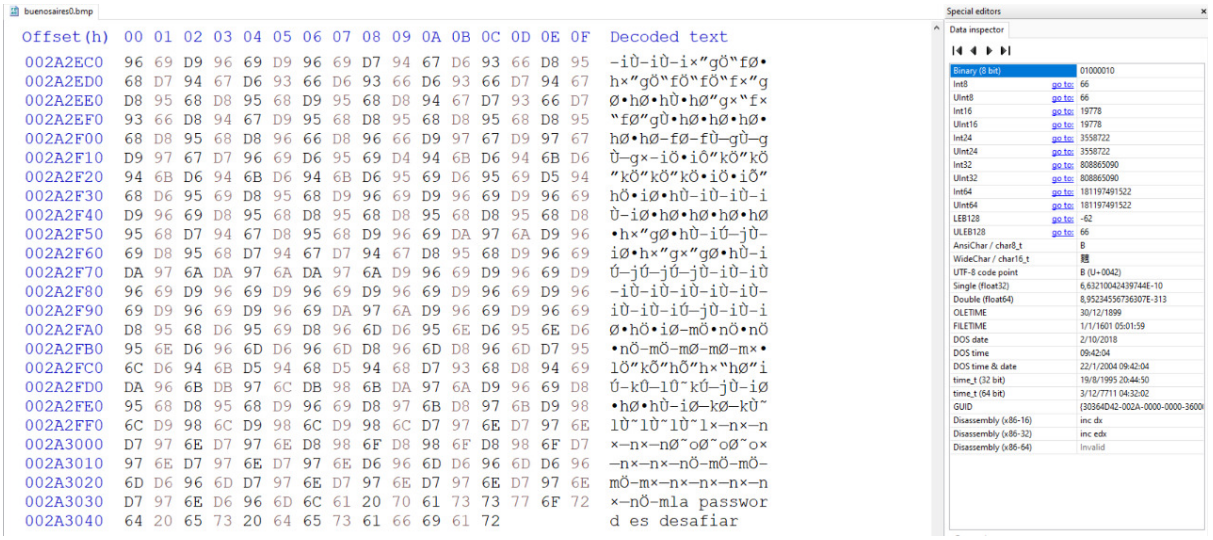

Figura 3: Hexdump del archivo buenosaires0.bmp

Se puede ver que al final del archivo dice:

"la password es desafiar"

Teniendo en cuenta esto usamos las configuraciones obtenidas previamente para extraer el archivo de frozen.bmp.

frozen.bmp

Como ya se encontró un archivo con LSB4 y otro con LSBI este debería ser LSB1, encriptado utilizando AES 128 en modo ECB con contraseña: *desafiar*

Corriendo lo siguiente:

*./stegobmp --extract -p grupo20/frozen.bmp -o extracted_frozen *
--steg=LSB1 -a=AES12 -m=ECB --pass=desafiar

Obtenemos el archivo extracted_frozen.wmv que es un video de 53s de una escena de la película Wanted (2008).

2.4 Algunos mensajes ocultos tenían, a su vez, otros mensajes ocultos. Indica cuál era ese mensaje y cómo se había ocultado.

En el archivo **buenosaires.bmp** se encontraba oculto una imagen. Esta imagen además era portadora de otro archivo oculto **sol20.txt**. Para obtener el segundo archivo oculto hubo que tratar a la imagen como un zip y descomprimir. Siguiendo las instrucciones del archivo **sol20.txt** y mirando a la imagen obtenida a partir de **buenosaires.bmp** se pudo obtener el algoritmo y modo de encriptación de otro archivo (AES 128 y ECB). El archivo con instrucciones además aporta información del tipo de archivo que está ocultado (un video).

2.5 Uno de los archivos ocultos era una porción de un video, donde se ve ejemplificado una manera de ocultar información ¿qué se ocultaba según el video y sobre qué portador?

En el fragmento del video (correspondiente a la película Wanted (2008)) se muestra una aplicación de la estenografía muy poco convencional: se oculta un código binario, donde en este caso el portador es el tejido de una tela, que corresponde a una palabra. Particularmente, si el hilo vertical del tejido pasaba por encima del hilo horizontal del mismo, se consideraba como 1, de otra manera era un 0. Siguiendo este hilo se anotaban los valores y se determinaba la palabra. Esa palabra era el nombre de una persona que debía ser asesinada.

2.6 ¿De qué se trató el método de estenografiado que no era LSB1 ni LSB4 ni LSBI? ¿Es un método eficaz? ¿Por qué?

Se trataba de concatenar el mensaje al final del archivo en texto plano. Por eso mismo se pudo ver al abrirlo con un editor de hex. La razón por la cual no se podía percibir el mensaje al abrirlo como una imágen es porque el header de la imagen indica la cantidad de bytes correspondientes a la imagen en sí, los programas usan esta información para representar la imagen. Como el texto estaba después, los programas de visualización de imágenes ignoran esa parte.

Este método no es seguro desde ninguna perspectiva, pero es una manera muy simple de embeber un mensaje. En linux por ejemplo bastaría con correr

echo “contenido del mensaje” > archivo.bmp

y con eso ya se cuenta con un mensaje oculto.

2.7 ¿Por qué la propuesta del documento de Akhtar, Khan y Johri es realmente una mejora respecto de LSB común?

La propuesta por Akhtar, Khan y Johri es realmente una mejora ya que apunta a mejorar la calidad de la esteganografía. El método es una técnica de inversión del bit. La idea es que la cantidad de bits modificados sean menores respecto al LSB1 común. Este nuevo método apunta a analizar la cantidad de bits a cambiar y compara con los bits actuales para ver si se cambió más del cincuenta por ciento o menos, en caso de ser mayor invierte los bits para que la cantidad modificada sea menor a la mitad de bits totales. De esta manera se garantiza que el mensaje original varía en menos que un cincuenta por ciento. En el LSB común, no se analiza el cambio de bits totales, lo que aumenta la probabilidad de empeorar la calidad de la estenografía.

2.8 ¿De qué otra manera o en qué otro lugar podría guardarse el registro de los patrones invertidos?

Los patrones en el momento se están guardando en el último bit de los primeros 4 bytes. Se podrían guardar también en 1 solo byte con LSB4, al estar cambiando un solo byte en ese caso se estaría modificando el valor de la componente azul (BGR) del primer píxel nada más lo cual no es mucho.

2.9 Leer el Segundo esquema y analizar (sin implementar) cuáles serían las ventajas que pueden verse.

La ventaja más evidente es que se verifican más patrones en los bytes correspondientes a la imagen con respecto al primer esquema, lo cual es posible utilizando también como referencia la imagen original. Esto permite modificar una menor cantidad de píxeles de la imagen, dando como resultado una imagen portadora más nítida y de mayor calidad.

2.10 Leer el Segundo esquema e indicar qué desventajas o inconvenientes podría tener su implementación.

Una posible desventaja, es que para poder implementar correctamente la técnica se asume que el receptor cuenta de antemano con la imagen original que se utilizará de portador, que se utilizará como parte del método de des-estenografiado.

2.11 ¿Qué dificultades encontraron en la implementación del algoritmo del paper?

Luego de entender el concepto del algoritmo, la implementación no fue particularmente difícil. Se llevó a cabo de la misma manera que el LSB1, pero a medida que se recorre el portador byte a byte (empezando 4 bytes adelante) se lleva la cuenta de la cantidad de bytes pertenecientes a cada grupo (00, 01, 10, 11) y la cantidad de bits que difieren del portador para cada grupo. Luego para cada grupo se calcula si hay que invertir sus píxeles de la siguiente manera:

$$group\_invert = group\_count / 2 \ < group\_diff$$

Es decir, si para un determinado grupo más de la mitad de los bits difieren, se los tendrá que invertir.
Luego, se recorre de vuelta el resultado anterior byte a byte, levantando el grupo al que pertenece ese byte e invirtiendo el último bit si corresponde.

Finalmente se cambia el último bit de los primeros cuatro bytes para representar si hay que invertir o no (1 si hay que invertir y 0 si no).

La mayor dificultad que presentó esta implementación es que originalmente se estaba guardando el patrón en el orden opuesto al de la cátedra (11, 10, 01, 00).

2.12 ¿Qué mejoras o futuras extensiones harías al programa stegobmp?

Por el momento el programa solo sirve para estenografiar archivos bmp (por ende el nombre), pero se podría extender para soportar otros formatos además de bmp. Una limitación del programa en el momento es el manejo de errores en algunos casos, ya que se confía en los parámetros del usuario para extraer información. Ahora si el usuario intenta extraer utilizando LSB1 un archivo que fue escondido con LSB4 por ejemplo, el archivo que se produce no es bueno. Se pueden hacer más validaciones para tener en cuenta estos casos y no producir archivos corruptos o con mal formato .

3. Apéndice

Nota: Las imágenes están como png ya que no se pudo cargarlas al documento como bmp.

Figura 4: Porter Image

Figura 5: stego-image - LSB1

Figura 6: stego-image - LSB4

Figura 5: stego-image - LSBI