

CRIPTOGRAFÍA Y SEGURIDAD

TRABAJO PRÁCTICO ESPECIAL

Esteganografía: Métodos de ocultamiento de información en archivos bmp

Grupo 3

Autores:

Alan Ezequiel Karpovsky - 49746

Federico DiNucci - 50120

Nicolás Loretí - 49479

Resumen

El objetivo del presente informe es comentar la experimentación de ocultamiento de información mediante técnicas de esteganografía, comentando las ventajas y desventajas de cada método.

10 de Junio de 2013

Índice

1. Manual de Usuario	2
1.1. Instalación	2
1.2. Uso	2
2. Análisis de resultados	4
2.1. 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?	4
2.2. ¿Qué ventajas podría tener ocultar siempre en una misma componente? Por ejemplo, siempre en el bit menos significativo de la componente azul.	4
2.3. 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
2.4. 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 del archivo?	7
2.5. Explicar detalladamente el procedimiento realizado para descubrir qué se había ocultado en cada archivo y de qué modo.	8
2.6. Extracción de información oculta	9
2.7. ¿De qué se trató el método de esteganografiado que no era LSB? ¿Es un método eficaz? ¿Por qué?	13
2.8. ¿Qué mejoras o futuras extensiones harías al programa stegobmp?	13
2.8.1. Soporte para archivos portadores WAV	13
2.8.2. Soporte para otros formatos de imágenes	14
2.8.3. Secreto compartido	14
3. Referencias	14

1. Manual de Usuario

1.1. Instalación

Para poder compilar el programa es necesario tener las librerías de criptografía. En Linux pueden ser instaladas mediante la ejecución del siguiente comando:

```
$ sudo apt-get install libmcrypt-dev
```

Luego debemos asegurarnos de que exista el directorio *obj* dentro de la raíz del proyecto donde el *makefile* creará los archivos objeto. Para esto basta con ejecutar:

```
$ make clean
```

Para finalizar compilamos el código fuente mediante:

```
$ make
```

Una vez realizados estos pasos deberíamos tener en el directorio el ejecutable **stegobmp**. En caso de que el mismo no contenga permisos de ejecución, otorgárselos mediante:

```
$ chmod +x ./stegobmp
```

1.2. Uso

Como primer argumento, podemos utilizar el *--help* que nos proporcionará una ayuda básica sobre los argumentos aceptados por el programa:

```
$ ./stegobmp --help
```

Las formas de ejecución válidas son las combinaciones de parámetros que se detallan en el enunciado provisto por la cátedra. Aclaración importante, los parámetros del programa deben ser precedidos de dos guiones (-); ejemplos de ejecución válidos:

- `./stegobmp -embed -in ultrasecret.txt -p images/lena512.bmp -out secretpicture -steg LSB1`
 - El programa ocultará dentro de la imagen portadora *lena512.bmp* el contenido del archivo *ultrasecret.txt* mediante el método de esteganografía *LSB1*. La salida del programa se guardará dentro de la carpeta *target* en un archivo de imagen con nombre *secretpicture*
- `./stegobmp -extract -p target/secretpicture -steg LSB1 -out salida`
 - El programa extraerá el archivo previamente ocultado mediante *LSB1* en la imagen *secretpicture* y creará en el directorio raíz el archivo *salida.txt* (que deberá ser igual a *ultrasecret.txt*. Notar que la extensión del archivo de salida viene dada por la extensión original del archivo previamente embebido en la imagen portadora.

A continuación se listan todos los argumentos disponibles para la acción de **embeber** información en una imagen portadora:

■ Parámetros obligatorios:

- **-embed**
Indica que se va a ocultar información
- **-in file**
Archivo que se va a ocultar
- **-p bmpfile**
Archivo bmp portador
- **-out bmpfile**
Archivo bmp de salida
- **-steg <LSB1 — LSB4 — LSBE>**
Algoritmo de esteganografiado

■ Parámetros opcionales:

- **-a <aes128 — aes192 — aes256 — des>**
Algoritmo de encripción
- **-m <ecb — cfb — ofb — cbc>**
Modo de encadenamiento
- **-pass password**
Password de encripción

Para el caso de la **extracción** de un archivo oculto dentro de un bmp, usamos los siguientes parámetros:

■ Parámetros obligatorios:

- **-extract**
Indica que se va a extraer información
- **-p bmpfile**
Archivo bmp portador
- **-out bmpfile**
Archivo de salida obtenido
- **-steg <LSB1 — LSB4 — LSBE>**
Algoritmo de esteganografiado

■ Parámetros opcionales:

- **-a <aes128 — aes192 — aes256 — des>**
Algoritmo de encripción
- **-m <ecb — cfb — ofb — cbc>**
Modo de encadenamiento
- **-pass password**
Password de encripción

2. Análisis de resultados

- 2.1. 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?**

Dado que el archivo portador utilizado para esteganografiar es una imagen, daría lo mismo que la ocultación comience en cualquier parte del archivo siempre y cuando se respete el encabezado BMP necesario para no corromperlo.

En realidad, lo ideal sería que la ocultación comience en un lugar al azar y no siempre en un punto fijo cosa de que para un atacante sea más difícil encontrar la información; el problema está en cómo comunicarse el emisor y receptor para intercambiar esta información.

- 2.2. ¿Qué ventajas podría tener ocultar siempre en una misma componente? Por ejemplo, siempre en el bit menos significativo de la componente azul.**

El algoritmo LSB oculta la información en el bit menos significativo de cada componente (ej. RGB) de la imagen portadora. El problema está en que modificar las 3 componentes (los tres colores) del pixel produce una mayor distorsión en el color resultante. Entones, un método que podría introducir mayor eficiencia y menos distorsión es el **ELSB**¹.

El *ELSB* trabaja en el dominio espacial y mejora la performance del LSB mediante el ocultamiento de la información sólo en la componente azul. El paper citado al pie de la página hace un análisis extenso sobre la distorsión de cada método y concluye que el *ELSB* es el método que produce menor distorsión entre los colores reales de la imagen portadora y los colores modificados por el ocultamiento de la información.

- 2.3. 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.**

Para realizar esta tarea se creó el archivo *bmp* todo negro con algunos puntos blancos que se muestra a continuación. La idea de generar esta imagen toda negra y con puntitos blancos vino dada por la facilidad luego a la hora de investigar resultados en un editor hexadecimal (el negro se representa como todos ceros).

¹Enhanced Least Significant Bit Algorithm For Image Steganography - http://www.ijcem.org/papers072012/ijcem_072012_08.pdf

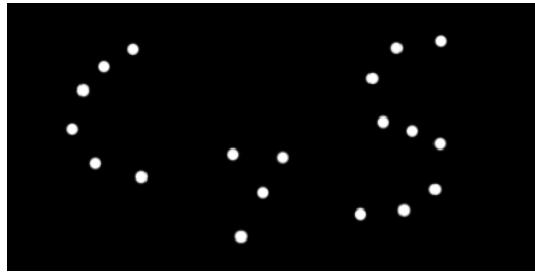


Figura 1: Archivo portador de prueba

Luego se tomó el archivo un archivo *mensaje.txt* el cual contenía el siguiente texto: *Ya termina la cursada de CyS!!*. Se procedió a ocultarlo con los distintos algoritmos de esteganografiado mediante:

```
$ ./stegobmp --embed -p blackandwhite.bmp  
--in ./mensaje.txt --out bwLSB1 --steg LSB1  
  
$ ./stegobmp --embed -p blackandwhite.bmp  
--in ./mensaje.txt --out bwLSB4 --steg LSB4  
  
$ ./stegobmp --embed -p blackandwhite.bmp  
--in ./mensaje.txt --out bwLSBE --steg LSBE
```

Una vez obtenidos los archivos de salida *bwLSBx*, se abrieron con un editor hexadecimal para compara su contenido.

Archivo original		LSB1	
0	42 4D 38 00 0C 00 00 00 00 00 36 00 00 00 28 00	0	42 4D 38 00 0C 00 00 00 00 00 36 00 00 00 28 00
16	00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00	16	00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00
32	00 00 02 00 0C 00 C3 0E 00 00 C3 0E 00 00 00 00	32	00 00 02 00 0C 00 C3 0E 00 00 C3 0E 00 00 00 00
48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
64	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	64	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
96	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	96	01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
112	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	112	01 01 00 01 00 00 00 00 01 00 00 00 01 00 00 01
128	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	128	01 01 00 00 01 00 00 00 01 00 00 01 00 00 01 00
144	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	144	01 00 01 00 00 00 01 00 00 01 00 01 01 01 00 01
160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	160	01 00 00 00 00 00 01 00 00 01 00 00 00 00 00 01
176	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	176	01 00 00 00 00 00 01 00 00 01 00 00 00 00 00 00
192	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	192	01 00 00 00 00 00 01 00 00 01 00 00 00 00 01 00
208	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	208	01 01 00 01 00 01 00 01 01 01 00 00 01 00 00 01
224	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	224	01 01 00 00 01 00 01 00 01 00 00 00 00 00 01 00
240	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	240	01 00 00 01 00 00 00 00 01 00 00 00 00 00 01 00
256	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	256	01 00 00 00 00 00 00 00 01 00 00 00 00 00 00 01
272	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	272	01 00 00 01 00 01 00 00 01 00 00 00 00 00 00 01
288	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	288	00 00 00 00 00 01 00 00 01 00 01 01 00 00 01 00
304	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	304	00 01 00 00 01 00 01 00 00 01 00 00 00 00 01 00
320	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	320	01 00 00 00 00 00 01 00 00 00 00 00 01 00 01 00
336	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	336	01 00 01 01 01 00 00 01 01 01 00 00 01 00 00 01
352	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	352	01 01 00 00 00 00 00 00 01 01 01 00 00 00 00 00
368	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	368	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
384	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	384	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	400	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
416	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	416	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Archivo original		LSB4	
0	42 4D 38 00 0C 00 00 00 00 00 36 00 00 00 28 00	0	42 4D 38 00 0C 00 00 00 00 00 36 00 00 00 28 00
16	00 00 00 02 00 00 00 00 00 00 01 00 18 00 00 00	16	00 00 00 02 00 00 00 00 00 00 01 00 18 00 00 00
32	00 00 02 00 0C 00 C3 0E 00 00 C3 0E 00 00 00 00	32	00 00 02 00 0C 00 C3 0E 00 00 C3 0E 00 00 00 00
48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
64	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	64	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	80	06 01 02 00 07 04 06 05 07 02 00 00 06 09 06 0E
96	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	96	07 03 06 01 06 04 06 01 02 00 00 04 06 05 02 00
112	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	112	04 03 07 09 05 03 02 01 02 01 00 0A 02 0E 07 04
128	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	128	07 08 07 04 00 00 00 00 00 00 00 00 00 00 00 00
144	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	144	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Archivo original		LSBE	
378888	4F C0 C0 C0 F8 F8 F8 FF FF FF FF FF FF FF FF	378888	4F C0 C0 C0 F8 F8 F8 FE FE FE FE FE FE FE FE
378896	FF FF FF F4 F4 A8 A8 39 39 39 04 04 04 00	378896	FE FE FE F4 F4 F4 A8 A8 39 39 39 04 04 04 00
378912	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	378912	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
372416	86 E8 E8 E8 FF	372416	86 E8 E8 E8 FE
372432	FF FF FF FF FF FF DE DE DE 65 65 65 00 00 00 00	372432	FE FE FE FF FF DE DE DE 65 65 65 00 00 00 00
372448	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	372448	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
373936	00 00 00 00 00 00 00 00 00 00 00 00 00 00 1D 1D	373936	00 00 00 00 00 00 00 00 00 00 00 00 00 00 1D 1D
373952	A0 F6 F6 F6 FF	373952	A0 F6 F6 F6 FF FF FE FF FE FF FE FF FF FF
373968	FF FF FF FF FF FF F2 F2 F2 7B 7B 7B 14 14 14 00	373968	FF FE FE FE FE FF F2 F2 F2 7B 7B 7B 14 14 14 00

Figura 2: Comparación de los distintos algoritmos de esteganografiado

A continuación se comparan las ventajas y desventajas encontradas al utilizar cada uno de los algoritmos:

Algoritmo	Ventajas	Desventajas
LSB1	Es un algoritmo simple. La diferencia entre el archivo portador original y el archivo con información oculta no es muy notoria ya que sólo se modifican, a lo sumo, 3 bits por pixel.	Se necesita un archivo portador 4 veces más grande que el necesario para utilizar LSB4.
LSB4	Es el algoritmo de esteganografiado que necesita el archivo portador de menor tamaño.	La información ocultada queda muy junta (o próxima) en una misma zona (se modifican de a 4 bits). Esto produce que los cambios entre el archivo portador original y el archivo con información oculta sean mucho más notorios para el atacante.
LSBE	Es altamente difícil para un atacante poder encontrar la información oculta; la misma queda distribuida a lo largo y ancho de todo el archivo bmp, siendo casi imperceptible.	Se necesita una imagen portadora de tamaño considerablemente grande para poder ocultar la información. En el caso particular en el que la imagen portadora no contenga bytes <i>FEh</i> o <i>FFh</i> , se reduce a 0 el espacio disponible para ocultar datos (ej. imagen bmp completamente negra).

Figura 3: Ventajas y desventajas de los distintos LSBy

Notar que en el caso de *LSBE*, los bits modificados están tan dispersos que no pudieron entrar en un *screenshot* del editor hexadecimal. Se colocaron sólo 3 porciones del archivo donde se visualizan los cambios pero el archivo original es muchísimo más extenso (verificar distancias entre número de líneas). En cambio, en el caso de *LSB1* y *LSB4*, se puede ver claramente que la captura comienza en la línea cero (primeros bytes del archivo) y toda la modificación ocurre antes de que salga del *scope* de la captura de pantalla. Obviamente se ha utilizado una imagen portadora grande y un archivo a ocultar muy pequeño, pero estas capturas muestran cómo los cambios del *LSBE* quedan suficientemente distribuidos a lo largo del archivo y son casi imperceptibles de detectar.

2.4. 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 del archivo?

No es conveniente poner la extensión al comienzo del BMP, luego del tamaño, por razones de seguridad. Si se pusiera la extensión de la forma antes descripta, el atacante encontraría mayor facilidad a la hora de intentar obtener la información oculta dado que la cantidad de extensiones es finita. El atacante podría proceder de la siguiente forma:

1. Obtengo una lista de todas las posibles extensiones válidas de archivos para cada SO.

2. Ubico posición de la extensión (recordar que la posición en la que se sitúa el tamaño es fija, luego del header *bmp*)
3. Realizo un ataque por fuerza bruta probando los 3 tipos de algoritmo de esteganografiado: LSB1, LSB4, LSB y voy comparando con las extensiones válidas obtenidas en el punto 1.
 - a) Notar que se debe ir levantando y comparando carácter a carácter.
 - b) Un software no demoraría demasiado tiempo en realizar esta tarea de forma automática: no se debe parsear todo el archivo, basta con recorrer hasta tener tantos caracteres como la extensión más larga encontrada en la lista del punto 1.

Lógicamente, el ataque anterior no sería válido para archivos con extensiones no estándar. Asimismo notar que el atacante, una vez que obtiene la extensión, obtiene el algoritmo de esteganografiado con el que se ocultó la información.

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

A nuestro grupo le tocaron analizar las siguientes cuatro imágenes:

- django.bmp
- django1.bmp
- kingsspeech.bmp
- miserables2.bmp

Lo primero que se hizo fue abrir todas las imágenes con un editor hexadecimal para empezar a analizar su contenido. Al hacer esto nos dimos cuenta que la imagen **django1.png** contenía al final del archivo el siguiente texto plano:

4320040	02 E4 05 03 E5 04 03 E5 04 03 E5 04 03 E5 61 6C 20 2E 70 6E	% Â Â Â Äal .pn
4320060	67 20 63 61 6D 62 69 61 72 20 65 78 74 65 6E 73 69 6F 6E 20	g cambiar extensión
4320080	70 6F 72 20 2E 7A 69 70 20 79 20 64 65 73 63 6F 6D 70 72 69	por .zip y descomprimir
4320100	6D 69 72	mir

Figura 4: Texto encontrado: *al png cambiar extensión por .zip y descomprimir*

Como la consigna aclaraba que había cuatro imágenes portadoras una que utilizaba el método LSB1, otra LSB4, otra LSBE y por último una que utilizaba una forma distinta de ocultamiento, se asumió que el archivo **django1.bmp** correspondía a este último caso.

Una vez descubierto el archivo esteganografiado por un método distinto a los enunciados en el trabajo práctico, se procedió a utilizar el desarrollo del equipo para averiguar el contenido de los archivos en cuestión. En la siguiente sección se explica en detalle el qué se encontró en cada archivo.

2.6. Extracción de información oculta

En esta sección se tratarán los siguientes temas:

- ¿Qué se encontró en cada archivo?
- Explicar detalladamente el procedimiento realizado para descubrir qué se había ocultado en cada archivo y de qué modo.
- Algunos mensajes ocultos tenían, a su vez, otros mensajes ocultos. Indica cuál era ese mensaje y cómo se había ocultado.
- Uno de los archivos ocultos era una porción de un video, donde se ve ejemplificado una manera de ocultar información, ¿cuál fue el portador?

Como ya se comentó en la sección anterior, el archivo **django1.bmp** contenía un mensaje en texto plano al final del archivo que indicaba que se le debía cambiar la extensión a un archivo png a .zip. Esto nos hizo asumir que, dentro de las imágenes portadoras, debía existir algún archivo oculto por la cátedra cuya extensión sea *png*.

Dentro del archivo **miserables1**, ejecutando la extracción mediante el algoritmo de esteganografiado **LSB1**

```
$ ./stegobmp --extract --p miserables1.bmp --steg LSB1 --out miserablesLSB1
```

se encontró el siguiente *screenshot* del juego buscaminas.



Figura 5: Screenshot del Buscaminas dentro de **miserables1.bmp**

Al ver que el archivo resultante tenía extensión *png*, recordamos el mensaje oculto en **django1.bmp** (cambiar extensión *png* por *zip* y descomprimir) por lo que se procedió a realizar dicha tarea.

Una vez descomprimido el archivo arrojó un archivo de texto llamado **sol4.txt** que se muestra a continuación:

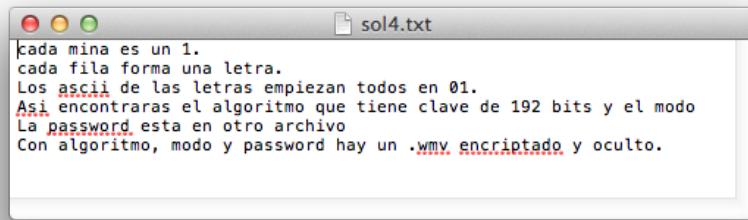


Figura 6: Texto encontrado luego de cambiar por zip el archivo obtenido de desesteganografiar la imagen portadora **miserables2.bmp**

Luego se procedió con las dos imágenes que faltaban. Dentro de la imagen **django.bmp** y mediante *LSBE* se encontró un PDF. Se ejecutó el siguiente comando:

```
$ ./stegobmp --extract --p django.bmp --steg LSBE --out djangoLSBE
```

Lo que arrojó como resultado un archivo *PDF* de una única hoja como se muestra a continuación:

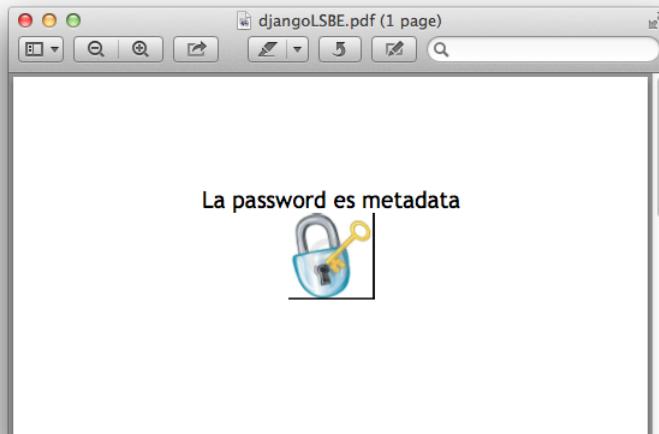


Figura 7: PDF escondido dentro de **django.bmp**

Quedaba un único método y una única imagen disponible por lo que se procedió a analizar el archivo **kingsspeech.bmp** mediante el algoritmo de esteganografiado *LSB4*.

Para este último caso, se utilizó toda la información antes obtenida de los demás archivos:

- Passowrd: metadata
- Algoritmo: AES192
- Modo: ECB

La password se había obtenido previamente dentro del arhchivo PDF, el algoritmo era parte del texto encontrado dentro de **sol4.txt** y el modo se obtuvo siguiendo dicho texto como se muestra a continuación:

Primero se marcó en el tablero del buscaminas las posiciones donde deberían estar las minas faltantes:



Figura 8: Minas dentro del buscaminas

Luego se hizo la traducción como se indicaba en el archvo de texto, obteniéndose:

Binario	Hexadecimal	ASCII
01000001	41	A
01100101	65	e
01110011	73	s
01000101	45	E
01100011	63	c
01100010	62	b

Finalmente ejecutamos el siguiente comando para obtener el video en cuestión:

```
$ ./stegobmp --extract -p ../migrupofiles/kingsspeech.bmp --steg LSB4
--out kingsspeechLSB4 -a aes192 -m ecb --pass metadat
```

El video obtenido es un fragmento de la película **Wanted** donde actúan James McAvoy, Morgan Freeman, Angelina Jolie.



Figura 9: Video escondido dentro de **kingsspeech.bmp**

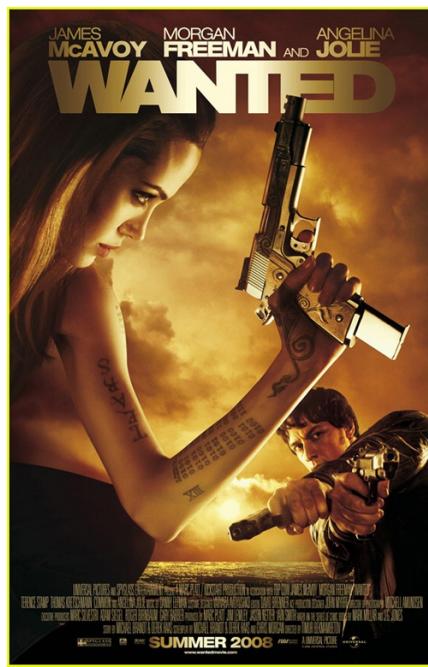


Figura 10: Portada del film **Wanted**

En dicho film Morgan Freeman explica sobre un método de ocultamiento de información utilizado por *La Hermandad* cuyo elemento portador era la tela. El método consiste en codificar en binario nombres y objetivos de la siguiente forma:

- Si el hilo vertical está por encima, representa un 1.
- Si el hilo vertical está por debajo, representa un 0.

La tela debe ser analizada bajo el microscopio (o una lupa con suficiente aumento) para poder detectar la posición de los hilos.

2.7. ¿De qué se trató el método de esteganografiado que no era LSB? ¿Es un método eficaz? ¿Por qué?

El método de esteganografiado que no era LSB fue el de insertar texto plano al final del archivo (descubierto mediante la apertura del mismo con un editor hexadecimal). Respecto a la efectividad del método, si nadie conoce la existencia de la información a ocultar y sólo interesa proteger el contenido, puede llegar a ser un buen método.

Una posible ventaja es que el ocultador cuenta con espacio ilimitado (o suficientemente grande) para guardar la información. Es decir, en los métodos LSB la cantidad de bits que el ocultador puede alterar es limitada; en cambio, concatenando al final del archivo datos, uno puede hacer crecer el archivo tanto como se quiera. Lógicamente, puede llegar a ser sospechoso tener una imagen BMP de 200 MB, pero la posibilidad está.

Ahora bien, en el caso de que el atacante conozca la existencia de la información oculta, el método resulta muy pobre. Basta con abrir el archivo con un editor hexadecimal y el atacante tendrá acceso directo a los datos ocultos.

2.8. ¿Qué mejoras o futuras extensiones harías al programa stegobmp?

2.8.1. Soporte para archivos portadores WAV

Dado que los métodos LSB1, LSB4 y LSBE no tienen en consideración el formato del archivo que están modificando ya que simplemente son distintas formas de decidir qué bits se deben modificar para ocultar la información, sería sencillo migrar el desarrollo de **stegobmp** para que soporte archivos de audio como portadores.

Siendo la modificación tan sutil, se espera que el oído humano casi ni pueda percibir la diferencia entre el archivo original de audio y el archivo con información oculta.

Lo que sí sería interesante analizar es el caso extremo: qué sucedería si se toma como portador una canción que tiene muchos silencios. Así como con las imágenes se puede analizar el caso de una imagen toda blanca o toda negra, suponemos que en el caso extremo del archivo de audio mudo, sí se notaría una diferencia y se escucharía lluvia o ruido aleatorio.

2.8.2. Soporte para otros formatos de imágenes

Como se muestra en la siguiente tabla, dentro de los formatos de imágenes válidos, *BMP* no es el mejor de ellos para utilizar en el campo de la esteganografía de imágenes.

	LSB in BMP	LSB in GIF	JPEG compression	Patchwork	Spread spectrum
Invisibility	High*	Medium*	High	High	High
Payload capacity	High	Medium	Medium	Low	Medium
Robustness against statistical attacks	Low	Low	Medium	High	High
Robustness against image manipulation	Low	Low	Medium	High	Medium
Independent of file format	Low	Low	Low	High	High
Unsuspicious files	Low	Low	High	High	High

* - Depends on cover image used

Figura 11: Formatos de imágenes para esteganografía

Cuando embebemos información dentro de un *BMP* existe un trade-off entre la invisibilidad del mensaje y la cantidad de información que podemos embeber. Es decir, el *BMP* nos ofrece una gran cantidad de bytes para ocultar información, pero el hecho de que muchos bytes resulten alterados le permite a un atacante darse cuenta a simple vista (ojo humano) que hay información escondida. Asimismo debe considerarse que el *BMP* es un formato en desuso y puede resultar sospechoso para alguien con conocimiento del tema interceptar una comunicación en la que se intercambien *BMPs*.

Para analizar más en detalle esta alternativa, se recomienda la lectura del paper de T. Morkel, J.H.P. Eloff, M.S. Olivier: **An overview of image steganography** (<http://www.prest-it.fr/Cryptographie%20et%20Steganographie/Sym%C3%A9trie/EN%20-%20Image%20Steganography%20Overview.pdf>)

2.8.3. Secreto compartido

No es muy difícil para un atacante generar un algoritmo que automáticamente extraiga información oculta de los archivos esteganografiados. Esto se debe a que los bytes que se alteran de la imagen portadora original ya están prefijados por el método utilizado (LSB1, LSB4 y LSBE). Una solución un poco más inteligente y segura sería que el emisor y el receptor se pongan de acuerdo previamente en cuáles deben ser los bytes que se alterarán (secreto compartido) y recién ahí se proceda con el ocultamiento. Esto hará que para el atacante sea muchísimo más complicada la tarea de extraer la información oculta.

3. Referencias

Se utilizaron los siguientes artículos provistos por la cátedra como material de consulta:

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

- Sridevi, Damodaram y Narasimham: **Efficient method of audio steganography by modified LSB algorithm and strong encryption key with enhanced security** (<http://www.jatit.org/volumes/research-papers/Vol15No6/15Vol15No6.pdf>)
- Cummings, Jonathan y otros: **Steganography and Digital Watermarking** (<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** (http://vector.ucaldas.edu.co/downloads/Vector1_3.pdf)
- Gómez Cárdenas, Roberto: **Esteganografía** (<http://www.cryptomex.org/SlidesCripto/Estegano.pdf>)
- Johnson, Neil F. y Jajodia, Sushil: **Exploring Steganography. Seeing the Unseen** (<http://www.creangel.com/papers/steganografia.pdf>)
- Gupta, Shilpa; Gujral, Geeta y Aggarwal, Neha: **Enhanced Least Significant Bit algorithm For Image Steganography** (www.ijcem.org/papers072012/ijcem_072012_08.pdf)