Criptografía y Seguridad
Trabajo práctico 2
Estenografía

Grupo 2

Alumnos:

- **Matías De Santi - 51051**
- **Cristian Pereyra - 51190**
- **Esteban Pintos - 51048**

Tabla de contenido

Introducción

La estenografía es la parte de la criptología en la que se estudian y aplican técnicas que permiten el ocultamiento de mensajes u objetos, dentro de otros, llamados portadores, de modo que no se perciba su existencia.

El objetivo del trabajo práctico especial era implementar un programa que permita ocultar y des ocultar información con algún método de estenografiado elegido, en archivos de extensión BMP. Se utilizaron archivos de este tipo ya que son muy simples, y al no estar comprimidos permiten modificar los datos a nivel bit sin corromper la imagen.

El programa debía también permitir encriptar la información a ocultar con diferentes algoritmos y modos, utilizando un string como contraseña. Para esto se utilizó como lenguaje C y se utilizó la librería *argtable2* para parsear los argumentos de entrada al programa y *openssl* para encriptar y desencriptar.

Estegoanalisis

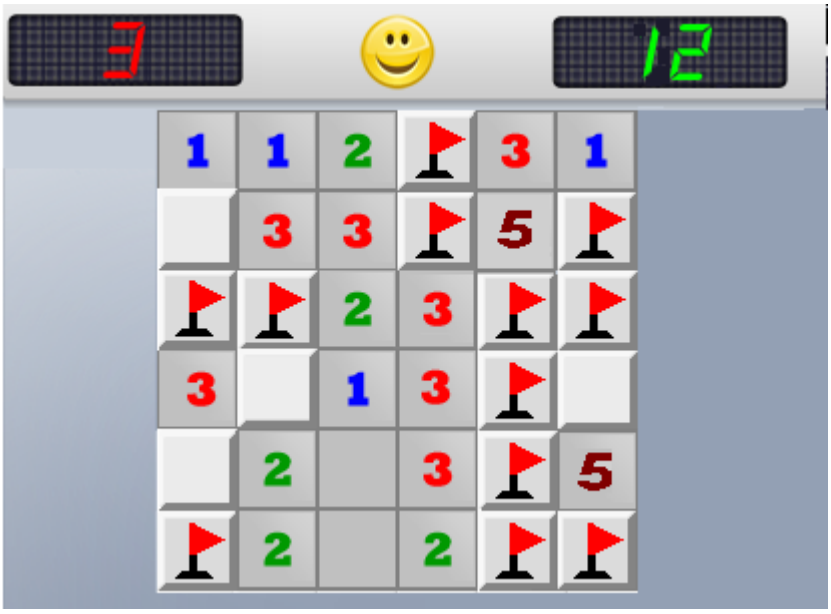
La cátedra nos dió cuatro imágenes en formato BMP de las cuales debíamos des ocultar cierta información pero no sabíamos de que manera. El objetivo era encontrar la manera y la información oculta.

Para poder descubrir que se había ocultado en los archivos provistos por la cátedra se tuvo en cuenta que tres de los cuatro estaban estenografiados utilizando LSB1, LSB4 y LSBE. Por lo tanto se fueron probando diferentes combinaciones hasta que se obtuvieron resultados coherentes y no aparecían errores por tamaños leídos negativos o archivos corrompidos. De esta manera se obtuvieron los siguientes datos:

1. De la imagen *hugo.bmp* se obtuvo utilizando LSB1 un PDF con el siguiente texto:

   al .png cambiarle la extension por .zip y descomprimir

2. De la imagen *ironlady.bmp* utilizando LSB4 se obtuvo un PNG con la siguiente imagen:



y luego siguiendo las instrucciones del ítem 1, se obtuvo la siguiente salida:

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

Con lo cual se resolvió el buscaminas anterior y se obtuvieron los siguientes valores:

Bytes obtenidos	Decimal	Letra
01000100	68	D
01100101	101	e
01110011	115	s
01000011	67	C
01000010	66	B
01100011	99	c

Como se puede observar en la tabla anterior, se obtuvo el text “Des CBc”, con lo cual dedujimos que alguno de los archivos que quedaban debía ser des ocultado utilizando como algoritmo DES y modo CBC.

3. Por último nos faltaba conseguir la contraseña, para esto se nos ocurrió abrir los archivos restantes con editor hexadecimal. Al abrir la imagen *lincoln.bmp* encontramos al final del mismo lo siguiente:

```
04f3330: ffff ffff ff00 6c61 2070 6173 7377 6f72  ......la passwor
04f3340: 6420 6573 2064 6573 6166 696f            d es desafio
```

Por lo tanto ya estábamos en condiciones de encontrar el .wmv encriptado y oculto.

4. Utilizamos el método que nos faltaba LSBE con el algoritmo DES, modo CBC y contraseña "desafio" para poder des ocultar el video final.

Cuestiones a analizar

Por último se pidió responder las siguientes preguntas:

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?

   Empezar desde otra ubicación podría ser una buena implementación ya que si siempre se utiliza la del primer componente del primer pixel, un atacante podría des ocultar la información y no se desea esto. Sin embargo para empezar desde otra ubicación habría que implementar algún método para que cuando se llegue al fin de la imagen se continúe ya sea por el principio o algún otro lugar deseado.

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

   Creemos que realizar esto no traería ninguna ventaja ya que seguramente como se esta perdiendo mucho espacio al no contar con todas las componentes, haya que usar más bits de la componente y esto produciría que la imagen se estropee y se noten variaciones en el color, y de esta manera un atacante podría darse cuenta que se trata de una imagen portadora de información oculta.

3. Estenografiar 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.

Algoritmo	Espacio necesario	Detección
LSB1	8 veces el tamaño del archivo a ocultar	Detectable para el ojo humano.
LSB4	2 veces el tamaño del archivo a ocultar	Mayor detección para un ojo humano y para análisis automáticos.
LSBE	8 veces la cantidad de bytes de valor 255 y 245 del archivo a ocultar	Detectable para el ojo humano pero más difícil al comparar bit a bit.

Como se puede observar no hay un algoritmo que sea el mejor, si no que esto depende de lo que se esta buscando. Si se quiere un mejor ocultamiento, LSBE resulta mejor debido a que es el más difícil para detectar cuando se utilizan

algoritmos que comparan bit a bit. Si se quiere generar un ocultamiento que ocupe menos espacio en la imagen portadora, conviene LSB4, aunque hay que tener en cuenta que como se modifica la mitad de la componente, para el ojo humano puede resultar más detectable.

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?

   No conviene ponerla al principio por que si no un atacante podría fácilmente reconocer que tipo de archivo se esta ocultando sabiendo que los primeros 4 bytes son el tamaño del archivo y los siguientes bytes hasta un "\0" darían la extension. Al poner la extensión al final, el atacante debería leer primero toda la información del archivo, y como esta es variable, no puede acceder directamente a la extensión del mismo.

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

   Explicado en la sección estenografía.

6. ¿Qué se encontró en cada archivo?

   Explicado en la sección estenografía.

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

   El archivo PNG recuperado, al cambiar su extensión a .zip y descomprimiéndolo, se obtuvo un archivo de texto. Esto es posible ya que los archivos PNG en el header tienen el tamaño de la imagen y no importa si en el cuerpo del archivo hay más información luego de ese valor. Por lo tanto es posible agregar más información al final y así poder generar un .zip.

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?

   Explicado en la sección estenografía.

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

   Se encontraron dos métodos de estenografiado que no eran LSB. Los métodos son los siguientes:

   i. **Ocultar texto plano en un archivo binario**: Este método permite a un atacante recuperar la información fácilmente utilizando un editor hexadecimal y observar con atención las secuencias que se generan. En este

caso fue muy simple ya que la información ocultada estaba en español / inglés de forma clara, pero podría haberse ocultado ciertas secuencias que al ojo humano suena a información oculta.

ii. **Agregar información extra al final de un archivo**: Es un método un poco más seguro que el anterior, pero sin embargo no sigue siendo muy eficaz ya que si un atacante se da cuenta que el tamaño en el header de un archivo no coincide con el tamaño de la misma, podría leer la información que sobra y generar algún otro tipo de archivo. Un ejemplo de esto se puede observar en el video des ocultado.

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

Las mejoras o futuras extensiones que le haríamos al programa stegobmp son las siguientes:

- Soportar otras imágenes portadoras a demás de BMP.
- Soportar otros archivos portadores, como de video, sónido, etc.
- Detectar automáticamente con qué método LSB se está estenografiando.
- Implementar nuevas formas de ocultamiento