Criptografía y Seguridad

Esteganografía

Grupo 6

Integrantes

D’onofrio, Nicolás	54160
Fernandez Rojo, Juan Pablo	53164
Horvat, Eric	55564
Kochian, Daniel Jonás	54110

Índice

Introducción

El objetivo del trabajo práctico consiste en realizar un programa de esteganografía, que nos permita esconder o encontrar un archivo dentro de un archivo portador BMP. Para esto se utilizó los métodos de esteganografía LSB1, LSB4 y LSBE.

El archivo oculto contenido dentro del BPM, puede estar des encriptado o encriptado con los modos CBC, CFB, ECB u OFB, utilizando como función DES o AES, en modo 128, 196 o 256 bits.

En el presente informe se encuentra un análisis de esteganografía con los 4 archivos provistos por la cátedra.

Por último, se resuelven las diferentes cuestiones contenidos dentra de la consigna.

Resolución

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?**

   No es recomendable empezar en otra ubicación. De la manera propuesta se decide esconder en el próximo byte, por lo que el proceso es secuencial y por ende más rápido. Además, si se desea empezar en otro lugar esa información debe especificarse por diseño, y la problemática radica en dónde establecer ese comienzo. El principal problema de lo antes mencionado es que se pierden bytes donde esconder información si el salto es muy grande.

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

   Esto lograría que la variación de colores sea menor, sin disminuir la facilidad de la implementación, ya que solo se ignoran los demás componentes.

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.**

	Ventajas	Desventajas
LSB1	● Indistinguibilidad visual ● Portación de archivos relativamente más chicos al portador (~⅛)	● Distinguibilidad al analizar cualquier secuencia de bytes
LSB4	● Portación de archivos de hasta casi la mitad del tamaño del portador	● Distinguibilidad visual
LSBE	● Indistinguibilidad visual ● Diferencia de bytes en únicamente aquellos que son iguales a 0xFE o 0xFF en el	● Capacidad de portación dependiente del contenido del portador

	portador	

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?**

   Esto puede permitir obtener información del archivo mismo, ya que al saber su extensión se puede tratar de interpretar los datos leídos. Además, como la extensión se encuentra al final del mismo, nos sirve de terminador de procesamiento, semejante a un EOF.

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.**

   Los puntos 5,6 y 7 se contestan a continuación.

   Se tenían 4 archivos BMP: medianoche1, roma, hugo4 y sherlock. La consigna indica que 3 de estos archivos contienen archivos oculto sin encriptación, uno por cada modo de esteganografiado. Se decidió probar con cada uno de estos archivos.

   Los resultados fueron:

   - medianoche1.bmp: Con LSB1 se obtiene un archivo .png (a partir de ahora medianoche1.png).
   - roma.bmp: Con LSB4 se obtiene un archivo .wmv que no se puede abrir.
   - hugo4.bmp : Se obtiene un error ya que el size del archivo es distinto al indicado en el header.
   - sherlock.bmp Con LSBE se obtiene un archivo .pdf (a partir de ahora sherlock.pdf).
   - sherlock.pdf: El contenido al abrirlo es la siguiente instrucción: “al .png cambiarle la extensión por .zip y descomprimir”.
   - medianoche1.png → medianoche1.zip: Contiene un archivo .txt con el siguiente contenido:

     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.

- medianoche.png: La imagen es un buscaminas, que luego de resolverlo, se toma los espacios en blanco como 0 y las bombas como uno, leyendo cada fila antepuesta con ‘01’, quedando luego de analizarlo en un editor hexadecimal, la siguiente información:

  01000100 (0x44 D)

  01100101 (0x65 e)

  01110011 (0x73 s)

  01001111 (0x4F O)

  01100110 (0x66 f)

  01100010 (0x62 b)

- hugo4.bmp: Recordando que el tamaño del archivo era mayor a lo que indicaba el header, lo abrimos con un editor hexadecimal, y al final del mismo se encuentra: “la password es descubrilo”.
- roma.bmp: Con LSB4, password indicada, DES y OFB, se obtiene un archivo .wmv que contiene un video de una porción de un capítulo de una serie televisiva.

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?**

   En el video se puede ver que el portador es una imagen.

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

   Se trata de escribir la información al final del archivo. Esto no es eficaz ya que la información es totalmente legible con un editor hexadecimal, y un control del header nos permitió darnos cuenta que algo no estaba bien.

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

    Dado que es de facil implementacion, probariamos la utilizar un LSB2, tomando los 2 bits menos significativos, ya que intuimos que sería menos notable que LSB4 pero portaria un archivo del doble de tamaño que LSB1. Ademas LSBE podria tomar los bytes que valgan 0x00 o 0x01, o de igual manera 0xFC o 0xFD, ya que creemos que podrían pasar de la misma manera con no tanta concurrencia y por ende, más difícil de discernir al leer los bytes.

11. **Indicar ventajas y desventajas de haber elegido utilizar el padding de bytes de los archivos BMP, qué otra opción podría haberse elegido, y por cada una,**

ventajas y desventajas (desde el punto de vista de implementación y desde el punto de vista del esteganografiado).

Nosotros utilizamos el padding de los archivos BMP ya que como principal ventaja nos facilitaba la implementación en cuanto a control de si este padding se encontraba activo y posterior salto de bytes. Sin embargo, una desventaja que tiene es que este padding tiene un valor fijo, y al estar modificándolo es evidente que el archivo es un portador. La otra opción que proponemos es realizar controles en el header, rechazando aquellos que tengan padding (esto nos limita en cuanto a portadores) o implementando el salto de bytes.

Conclusión

En este trabajo práctico desarrollamos un programa que nos permite ocultar un archivo dentro de un portador BMP; donde destacamos el manejo de bits para la técnica de esteganografiado, el manejo práctico de las funciones y modos de cifrado vistos en clase y el análisis de archivos según la información que contienen. Más aún, logramos el estegoanálisis de 4 archivos para encontrar diferentes archivos, y entre ellos, un video encriptado y oculto.