Criptografía y Seguridad - 72.44

Trabajo Práctico de Implementación

Informe

Profesores:

Pablo Eduardo Abad
Ana María Arias Roig
Federico Eduardo Castañeda
Rodrigo Ramele

Autores:

Gastón Ariel Francois, 62500
Andrés Carro Wetzel, 61655
Alejo Flores Lucey, 62622
Nehuén Gabriel Llanos 62511

Lunes, 4 de noviembre

Índice general

1. Introducción 1
2. Análisis del documento sobre LSBI 2
   2.1. Organización del documento 2
   2.2. Descripción del algoritmo 2
   2.3. Notación utilizada 2
3. Comparación de algoritmos de esteganografía 3
4. Obtención de mensaje oculto 5
   4.1. Mensaje oculto dentro de otro mensaje oculto 7
   4.2. Explicación del fragmento de video 8
   4.3. Método alternativo a LSB1, LSB4 y LSBI 8
5. Análisis de LSBI 9
   5.1. Almacenamiento de patrones 9
   5.2. Dificultades en la implementación 10
6. Mejoras o futuras extensiones 11
7. Conclusiones 12

1 Introducción

En este informe se hará el análisis de la realización de un programa de esteganografiado sobre archivos de tipo *.bmp*. Además se detallará sobre las implementaciones realizadas de los algoritmos de *Least Significant Bit* (LSB) en sus distintas variaciones LSB1, LSB4 y LSBI. Adicionalmente, se analizará la variante LSBI en particular *(Least Significant Bit Improved)* que es extraída de un paper escrito por Majeed y Sulaiman [1].

Ultimamente, pondremos a prueba las implementaciónes realizadas de los algoritimos, sobre los archivos *.bmp* otorgados por la cátedra que contienen información oculta.

2 Análisis del documento sobre LSBI

Para la implementación del algoritmo de esteganografiado LSBI se proveyó por la cátedra un paper sobre la variante al método propuesta por Mohammed Abdul Majeed y Rossilawati Sulaiman [1]. Como análisis posterior a la implementación, surgen algunos puntos sobre los cuales vale la pena hacer hincapié.

2.1. Organización del documento

En términos generales el documento se encuentra estructurado en 8 secciones, las cuales abarcan desde la introducción hasta la discusión y conclusiones. En este sentido, si bien la separación en secciones favorece a la lectura fluida, resultaría mejor agrupar las propuestas del método en una sección diferenciada respecto a la explicación de contenidos previos, es decir, agrupar todo lo referido a conocimientos previos en una sección de marco teórico por un lado y todo lo referido al nuevo método por otro. De esta forma, quienes ya cuentan con el conocimiento previo pueden dirigirse directamente a la sección diferencial del paper.

2.2. Descripción del algoritmo

En cuanto a claridad del algoritmo, la explicación es clara y con el largo adecuado para no resultar extensa. Se hace un punteo paso a paso lo que favorece el entendimiento y se proveen recursos explicativos como imágenes y cuadros lo cual agiliza la compresión.

Algo negativo a destacar recae en la sección de resultados. Donde al mostrar los resultados del algoritmo se citan imágenes como *Lena.jpg* o *Peppers.jpg*, pero nunca se brinda acceso al antes y al después de las imágenes, por lo que se encuentra un vacío notable en la explicación del algoritmo y de los resultados.

2.3. Notación utilizada

La notación y el lenguaje a lo largo del documento resulta correcto pero no un aspecto a destacar. En primer lugar, notamos una constate repetición en definir el término PSNR *(Peak Signal-to-Noise Ratio)*, lo cual resulta una falla menor. Por otra parte, se nota un error en la quinta página del documento donde se menciona *"The following text includes 226 characters or 2881 bits"*, esto claramente es un error si asumimos que cada caracter es un byte y cada byte posee 8 bits; en este sentido, se debería enunciar *"The following text includes 226 characters or **1808** bits."*.

3 Comparación de algoritmos de esteganografía

Como sus nombres sugieren, estos métodos de esteganografía se basan en la modificación de los bits menos significativos de cada byte dentro de un pixel de una imagen de tipo *Windows Bitmap*, para ir almacenando un mensaje o archivo oculto. Cada pixel se representa con tres bytes: uno para el color azul, otro para el color verde y el último para el color rojo.

El método de LSB1, modifica únicamente el ultimo bit menos significativo de cada byte, mientras que el LSB4 modifica los ultimos cuatro bits menos significativos. Estos algoritmos se pueden generalizar a la familia LSB-N donde el N determina la cantidad de bits menos significativos que se modifican. Dentro de la misma familia es importante destacar la elección del N, mientras mayor sea el valor que tome N, más información podrá abarcar el pixel, pero más se notará visualmente la alteración de la imagen con respecto a la original, dado que estamos modificando cada vez mas los colores. En cambio, con un N menor, se podrá ocultar menos información en cada pixel, pero visualmente es más complicado de diferenciar la imágen esteganografiada y la original. En cuanto al algoritmo LSBI, se modifican incluso menos bytes que la implementación de LSB1 dado que no se alteran los bits del color rojo dentro del pixel. Esto hace que sea incluso más díficil de notar la alteración sobre la imágen original, pero a contraparte, se requiere de un archivo portador más grande para esconder el mensaje oculto.

Para demostrar lo que argumentamos anteriormente, mostraremos una imagen con un fondo verde y para comparación la misma imágen esteganografiada por los distintos métodos. Para apreciar bien las diferencia entre las imagenes, recomendamos subir el brillo de la pantalla.

Figura 3.1 *Imagen original*

Figura 3.2 *Imagen esteganografiada con LSB1*

Como se puede ver, en la imagen esteganografiada por el algoritmo LSB4 se nota en la parte inferior de la imagen que el verde se destiñe minimamente, evidenciando que la imagen ha sido alterada. En cambio, en LSB1 y LSBI es más díficil diferenciarla de la imagen original.

A continuación, a modo de resumen, realizamos una tabla comparativa entre los

Figura 3.3 *Imagen esteganografiada con LSB4*

Figura 3.4 *Imagen esteganografiada con LSBI*

tres algoritmos discutidos en este informe:

Método	Ventajas	Desventajas
LSB1	Díficil de detectar cambios visualmente sobre el archivo.	Al usar únicamente un bit por byte, se requiere de un portador 8 veces más grande que el mensaje oculto.
LSB4	Se usan 4 bits por cada byte del portador, solo necesito un portador 2 veces mas grande que el mensaje oculto.	Es fácil de detectar los cambios visualmente sobre el archivo.
LSBI	Es más dificil de detectar cambios en el archivo en comparación con LSB1 dado que no cambia los bytes del rojo.	Como solo se usa el bit menos significativo del byte azul y el byte verde, se necesita de un portador doce veces mas grande que el mensaje oculto.

Cuadro 3.1 *Cuadro Comparativo sobre los algoritmos estudiados*

4 Obtención de mensaje oculto

A modo de desafío, la cátedra otorgó al grupo 4 imágenes *.bmp* esteganografiadas con diferentes métodos con la idea de utilizar las implementaciones propias para descifrar todos los mensajes.

En un primer momento, como no contábamos con ninguna información adicional, probamos todos los métodos de esteganografía con todas las imágenes provistas. De todos estos intentos obtuvimos dos resultados notables.

El primer resultado notable resultó al desesteganografiar **montevideo.bmp** con LSBI, obteniendo un archivo *.pdf* con el contenido *al .png cambiarle la extensión por .zip y descomprimir*

```
./stegobmp -extract -p montevideo.bmp -out motevideo -steg LSBI
```

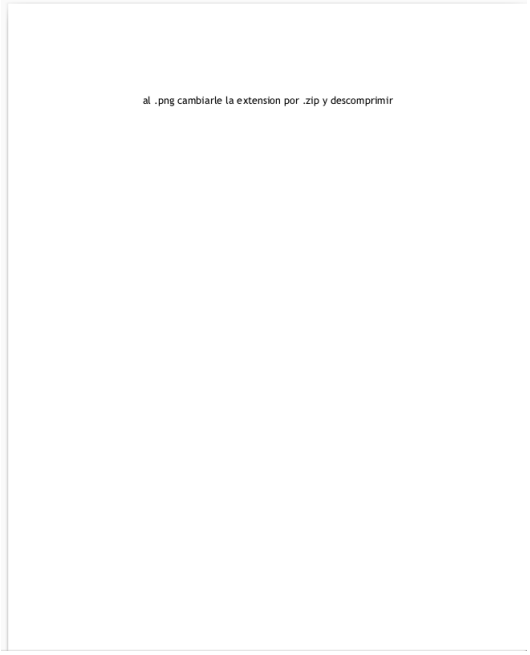

Figura 4.1 *Imagen ilustrativa del PDF extraído de montevideo.bmp*

Por otra parte, al continuar intentando con otras imágenes y utilizar el método LSB1 con el archivo **kings1.bmp** obtuvimos una imagen *.png* alusiva a un juego de buscaminas en proceso de resolución.

```
./stegobmp -extract -p kings1.bmp -out kings1 -steg LSB1
```

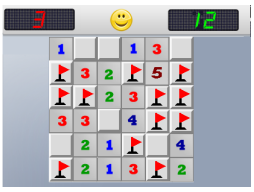

Figura 4.2 *Imagen PNG extraida de Kings1.bmp*

Como previamente contábamos con las instrucciones de **montevideo.pdf**, decidimos cambiar la extensión de la imagen obtenida y descomprimir. Al realizar esto, obtuvimos un segundo mensaje que indicaba

> 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 192 bits y el modo
> La password esta en otro archivo
> Con algoritmo, modo y password
> hay un .wmv encriptado y oculto.

Por lo tanto, procedimos a resolver el buscaminas para luego, utilizando las instrucciones encontradas, revelar el mensaje oculto dentro del juego.

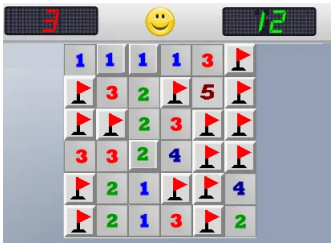

Figura 4.3 *Imagen del buscaminas resuelto*

> 0100 0001 0110 0101 0111 0011
> 0100 0011 0110 0110 0110 0010

> 41 65 73 43 66 62

> AesCfb

Por todo esto, podemos deducir que el mensaje oculto en algún otro archivo *.bmp* tendrá como algoritmo de encriptación a AES 192 y a CFB como el modo de encriptación.

Llegados a este punto, con los métodos de esteganografía implementados no podíamos encontrar ningún indicio más en los dos archivos restantes, por lo que pasamos a utilizar otras técnicas en búsqueda. Entre uno de estos intentos utilizamos el comando `strings`, el cual obtiene secuencias de caracteres imprimibles dentro de un archivo y obtuvimos una pista en el final del archivo **roma.bmp**.

```
strings roma.bmp
```

(...)
}yxlhthd
uZũZ
e3la password es descubrilo

Contando con esta nueva información, ya teníamos todos los datos para realizar la desencripción del archivo faltante **frozen.bmp**.

```
./stegobmp -extract -p frozen.bmp -out frozen -steg LSB4 -m CFB -a AES_192 -pass
    descubrilo
```

De esta forma, se logró extraer de una imagen un video *.wmv* de 53 segundos de duración explicando un antiguo método de esteganografía en paneles tramados con un telar.

Figura 4.4 *Fragmento del video obtenido*

4.1. Mensaje oculto dentro de otro mensaje oculto

Como se mostró previamente, de **montevideo.bmp** se pudo extraer un archivo PDF con la instrucción *al .png cambiarle la extensión por .zip y descomprimir* y del archivo **kings1.bmp** se pudo extraer un archivo PNG. Por lo tanto, se dio por entendido que dentro del mensaje oculto del PNG existía un segundo mensaje oculto al cual se debía acceder descomprimiendo el archivo.

Al realizar tal extracción, notamos que efectivamente existía un segundo mensaje oculto que arrojaba lo siguiente. De esta forma determinamos que la imagen del buscamina resultaba ser a su vez un portador de un nuevo mensaje.

> 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 192 bits y el modo
> La password esta en otro archivo
> Con algoritmo, modo y password
> hay un .wmv encriptado y oculto.

4.2. Explicación del fragmento de video

Una vez obtenido el mensaje oculto en **roma.bmp** se puede ver un video de aproximadamente un minuto donde se explica un antiguo método de esteganografía descubierto por un grupo de tejedores llamado *La Hermandad*.

Este método se sostiene en la forma de entrelazar las tramas de un tejido, donde en función de la posición del hilo vertical se puede interpretar el mensaje oculto. Cuando el hilo esta enhebrado por debajo entonces se obtiene un 0 y si esta enhebrado por arriba se obtiene un 1. Mediante esta simple regla se podían intercambiar mensajes ocultos a simple vista de todos utilizando como portador el tejido realizado por el telar.

4.3. Método alternativo a LSB1, LSB4 y LSBI

Dentro de los métodos utilizados para ocultar mensajes notamos la utilización de LSBI, LSB4, LSBI y un cuarto método que recae en colocar el mensaje en formato literal al final del archivo. Como se puede notar, este método cuenta con varias falencias comenzando por colocar el contenido secreto en formato literal, logrando que con un análisis muy superficial se pueda llegar al valor y en segundo término colocando este contenido literal al final del documento, haciendo que sea aún más fácil su obtención. Sin dudas este método resulta menos efectivo en terminos de ocultamiento respecto a los demás métodos implementados.

5 Análisis de LSBI

La variante propuesta por Majeed y Sulaiman resulta una mejora si tomamos como referencia la premisa de querer dificultar la detección un mensaje oculto, pero no cumple las expectativas si tomamos como referencia otra métrica como el tiempo de cómputo.

Al analizar lo postulado en el documento, se puede notar que al revertir los cambios en las situaciones de mayor reemplazo de bits, se logra que el documento no cambie de forma innesesaria y solo reporte en bits cual fue el tratamiento provocando así, que la imagen tanto visual como estadísticamente, no sufra cambios notables como en el resto de métodos. Sumado a esto, considerar que el rojo no se modifique por ser el color más detectable para el ojo humano, resulta una gran optimización en términos de ocultamiento del mensaje.

En cualquier caso, vale también la aclaración de que si bien resulta una mejora en el ocultamiento de los datos, LSB1 no resulta un método que deje rastro perceptible de forma visual para un posible detector de mensajes. En este sentido, si bien LSBI resulta una mejora no es una mejora distinguible, al menos a simple vista.

5.1. Almacenamiento de patrones

En la implementación propuesta por Majeed y Sulaiman se opta por almacenar los patrones invertidos antes del mensaje. Esta manera genera muchas ventajas, desde aportar claridad como en reducir la complejidad computacional, ya que desde el comienzo del recorrido por el archivo se cuenta con los valores de patrones invertidos. Desde nuestro lugar sugerimos dos posibles ideas de reemplazo, no necesariamente superadoras.

En primer lugar, se postula colocar los valores de patrones almacenados al final del documento y, por supuesto, sin invertir al igual que en la implementación actual. De esta forma se podría consultar solo a los últimos 4 bits del archivo y recibir de allí la información necesaria.

En segundo lugar, se postula utilizar 4 bits del header del archivo para ocultar la información de los patrones invertidos allí. En este caso se debe ser minucioso en la elección de qué campo modificar, ya que puede resultar en una alteración al formato del archivo. Por ejemplo, se podrían usar bits reservados, que hoy en día simplemente son ignorados por los lectores de imágenes BMP.

5.2. Dificultades en la implementación

Una gran complicación a nivel de desarrollo recayó en no intentar desde un comienzo interpretar a LSB1, LSB4 y LSBI como 3 variantes con el mismo método central similar. En este sentido, pensar las 3 implementaciones como objetos diferenciados generó demoras en la implementación de los 3 métodos.

En lo puntual de LSBI, la mayor complicación recayó en tener que mantener el valor rojo del RGB inmutable. De esta forma, se tuvieron que implementar dos índices diferenciados entre información de payload y carrier, generando complicaciones en la sincronización de estos dos índices.

6 Mejoras o futuras extensiones

Como mejoras futuras al programa `stegobmp`, se le podría agregar una opción para controlar el uso del *padding* en los algoritmos de encriptación. Además, tambien se le podría agergar el manejo de más extensiones de archivo mas allá de la extensión *.bmp*. Finalmente, como extensión de `stegobmp` se podría agregar una herramienta que encuentre posibles mensajes embebidos en el archivo portador, probando con cada método conocido.

7 Conclusiones

A lo largo del trabajo se pudo no solo desarrollar, sino sobre todo entender diferentes mecanismos de esteganografía y sus principales ventajas y desventajas. Se pudo notar, a su vez, cambios visibles en función de los mecanismos utilizados y cambios en el tiempo de cómputo de los mecanismos más avanzados a la hora de ocultamiento. Al mismo tiempo, se logró implementar una solución con la utilización de librería de encriptación AES y DES de forma satisfactoria. Finalmente, contar con un desafío donde poner en práctica todo lo desarrollado resultó una meta interesante y acorde a lo esperado para el trabajo práctico.

Bibliografía

[1] T.R. Andel y A. Yasinsac. «On the credibility of manet simulations». En: *IEEE Computer* 39.7 (jul. de 2006), págs. 48-54.