72.44 Criptografía y Seguridad
Trabajo practico de implementación - Informe
Primer cuatrimestre 2022

Grupo : 7
Integrantes :

- Igal Leonel Revich - Legajo : 60390
- Salustiano Jose Zavalia Pangaro - Legajo : 60312
- Agustin Tormakh - Legajo : 60041

Cuestiones a analizar

1. ***Discutir los siguientes aspectos relativos al documento***
    a. ***Organización formal del documento***

El documento está bastante organizado en cuanto a sus secciones (“Abstract”,”Introduction”,”Implementation”,”Conclusión”), acorde a la estructura de los papers científicos habituales.

b. La descripción del algoritmo

La descripción del algoritmo está bastante completa, pero creemos que hubiera estado bueno algún ejemplo de implementación (en c o en algún otro lenguaje), para visualizarlo más en detalle, y hacer más sencilla la implementación del mismo.

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

La notación utilizada es buena, pero ciertas cosas pueden dar a confusión, por ejemplo en la explicacion del segundo esquema de LSBI, denota A,B,C y D como pixeles, pero al ver dicha explicación más en detalle, entendimos que se refería a los diversos conjuntos de bytes a analizar por cada patrón (con bit menos significativo en 0 y cambio a 1, con bit menos significativo en 0 y permanece en 0, con bit menos significativo en 1 y cambió a 0, con bit menos significativo en 1 y permanece en 1). En cuanto a errores, uno pequeño que descubrimos es que en la explicación del primer esquema del LSBI, cuando nombra los patrones a tener en cuenta repite 2 veces 10 en vez de poner 10 y 01. A su vez, por lo que vimos no especifica en ningún lado donde insertar el registro de patrones invertidos en ninguno de los 2 esquemas.

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.***

Metodo\Categorías	*Tiempo de ejecución (s)*	*Diferencia en bits con el archivo portador*	*Ventajas*	*Desventajas*
LSB1	0.004784	190180	Es rapido y es sencilla la manipulación de bits	El tamano del archivo a ocultar debe ser bastante menor al del portador (Por cada 1 byte a ocultar del archivo, se requieren 8 del portador)
LSB4	0.002536	205957	Es mas rapido que LSB1, y puede ocultar archivos mas grandes que este (por cada 1 byte a ocultar del archivo, se requieren 2 bytes del portador)	Altera en gran medida la imagen
LSBI	0.009836	169083	Deja la imagen nueva muy parecida a la original	Es mas lento (debido a las inversiones a realizar)

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***

    1. Se abrieron en un Hexedit los 4 archivos, para analizar qué tenían adentro. Se sabía que había un archivo que no estaba esteganografiado, con lo cual, al analizar los 4 archivos, vimos que el “back.bmp” al final de todo tenía escrito *“la password es camuflado”,* con lo cual, a partir de eso dedujimos que este archivo era el que no estaba esteganografiado, y dicha password era la correspondiente a usar para realizar la extracción del archivo con encripción.

2. A partir de esto último, tomamos el archivo “budapest.bmp”, y empezamos a probar los métodos de extracción de esteganografía sin encripción, a ver cual era el correspondiente a dicho archivo. Luego de una serie de pruebas, al realizarse la extracción con LSBI, obtuvimos un pdf, que adentro tenia la siguiente pista : “*al .png cambiarle la extensión por .zip y descomprimir”.* Con lo cual, a partir de esto obtuvimos que este era el archivo esteganografiado con LSBI, y otra pista que nos serviria para la extracción del archivo encriptado.

3. Para el archivo “roma.bmp” nos quedaban 2 posibilidades de métodos de esteganografía (LSB1 y LSB4). Al probar con ambas, obtuvimos un png con la imagen de un buscaminas. Al tener dicha extensión, y teniendo en cuenta la pista obtenida en el archivo anterior, cambiamos la extensión de dicha imagen a un .zip, y al descomprimirlo, obtuvimos un archivo de texto “*sol7.txt”,* con indicaciones de como interpretar la imagen del buscaminas. Siguiendo las mismas, obtuvimos que el método y modo de encriptación a utilizar para el archivo encriptado eran Des y Cfb respectivamente

4. Habiendo hecho lo anterior, sabemos entonces que el archivo “hugo4.bmp” es el encriptado, y hay que realizar la extracción con método de esteganografiado LSB4, método de encriptación Des, modo de encripcion Cfb y password camuflado. Al ejecutarse el programa se obtiene un video de extension .wmv, con una parte de una serie donde se hace referencia a ocultar archivos dentro de un portador.

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

El mensaje que tenia otro archivo oculto era el que devolvía el archivo “roma.bmp”, que era el png que al cambiarle la extension a .zip, se obtiene adentro del mismo el txt con las indicaciones de como interpretar la imagen del buscaminas. Para analizar como se oculto el mismo, abrimos el png que se obtuvo de respuesta en el hexedit, y vimos que sobre el final del mismo se veía un “*sol7.txt”,* que es el archivo que esta adentro del zip. Luego, al cambiarse la extensión y ver las propiedades del zip, observamos que el peso es mucho mayor que el del txt que es el único elemento del zip, y por ende tambien es mucho mas grande que un nuevo zip armado a partir de dicho archivo. Por ende, lo que interpretamos es que al cambiarse la extensión, el archivo de respuesta no es el zip con el txt dentro, sino que sigue siendo la imagen (por ende, posee el mismo tamaño que la misma), que al final de la misma tiene el zip correspondiente adentro, por ende al cambiar la extension a .zip y al encontrar el zip correspondiente, funciona como tal.

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?

Por lo que vimos en el video, a la chica le llamaba la atencion que el archivo pese mas de lo que deberia, y menciona que puede ser un “error de compresion”. A su vez, el chico despues le dice que “el agente pudo haber ocultado datos en el archivo”. Con lo cual, a partir de eso, y de lo explicado en la pregunta anterior, entendemos que se oculta una carpeta comprimida (por ejemplo un .zip) en un portador BMP.

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

Este método consistia en poner la password directamente sobre la imagen, pisando los bytes de la misma. Este no es un método eficaz, debido a que con un editor de archivos hexadecimal como el hexedit puede verse directamente la información, y a su vez mientras mas grande sea la información a ocultar, al pisarse sobre los bytes de la imagen esta quedara más distorsionada.

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

Es una mejora respecto de LSB común, debido a que mediante el registro de cambios de cada patrón y la inversión de bits menos significativos, se logra que la imagen cambie lo menos posible, lo que hace que sea más complejo encontrar la información escondida en la misma.

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

Se podrían utilizar los 4 bits menos significativos del primer byte de la imagen para guardar dicho registro, por ejemplo, si el primer byte de la imagen es 1010**1001,** indicaría que los patrones 00 y 11 se invirtieron, mientras que los restantes no. Otra alternativa podría ser dejar los primeros 4 bytes en 0 o en 1, dependiendo si el patrón correspondiente debe o no ser invertido. Para el ejemplo anterior, estos deberían ser 00000001 00000000 00000000 00000001.

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

Las ventajas de dicho esquema en comparación al primero es que hace que la imagen se mantenga más parecida a la original, dado que por cada patrón, se invierte el bit menos significativo únicamente en caso de que la cantidad que haya cambiado de 0 a 1 sea mayor a la cantidad que se mantuvo en 1, y de la misma forma para los que cambiaron de 1 a 0 frente a los que se mantuvieron el 1, haciendo que la imagen original cambie mucho menos.

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

Una desventaja de este esquema consiste en que ahora se tienen 8 patrones en vez de 4 (ya que por cada uno de los del primer esquema, se analiza el bit menos significativo original), con lo cual, ahora para registrar los patrones invertidos serian necesarios 8 bytes en vez de 4, lo que implica que se requiere mas espacio de la imagen portadora. Por otro lado, otra gran desventaja es que para interpretar si un byte tiene o no inversión, es necesario tener la imagen original, dado que si por ejemplo se tiene el byte 01100**010** (patron 010), y para el ocultamiento los patrones 010 se dejaban iguales y los patrones 011 se invierten, teniendo solo la imagen modificada no hay forma de saber si corresponde a un byte con patron 010 que se mantuvo constante, o a uno con patron 011 que se invirtió, con lo cual se requiere la imagen original para saber a cual de los 2 casos corresponde. Creemos que este problema surge de que este esquema, a diferencia del anterior, utiliza como indicador un bit que esta modificando, lo cual a la hora de realizar la extracción de archivos a partir del mismo puede dar lugar a confusión.

11. ¿Qué dificultades encontraron en la implementación del algoritmo del paper?

En primer lugar, el algoritmo demostró ser más complicado que los anteriores ya que demandaba dos pasadas y requería de lógica más sofisticada que los anteriores. Si bien la implementación en sí no era demasiado difícil, el principal inconveniente que sufrimos fue la dificultad de depuración que tenía el programa.

En un principio nos equivocamos intentando un uso indebido de la lógica del LSB1 a la hora de hacer el embedding. Escribimos en la primera pasada (adaptando el algoritmo LSB1) y eso llevaba a errores a la hora de la inversión en la segunda pasada. También tuvimos inconvenientes haciendo uso de máscaras para identificar los patrones, o recorriendo los archivos para saltear el espacio designado a los bytes de inversión, escribiendo el payload en lugares indebidos.

Estos problemas no son demasiado graves, pero demostraron ser difíciles de debuggear. La manipulación fina de punteros que demandaba el algoritmo necesitaba de un proceso de diseño meticuloso, y en nuestro apuro terminamos leyendo el hexa de la salida del programa en busca de patrones que explicaran el error. Esto demostró ser tedioso y un considerable sumidero de tiempo.

Por otro lado, la extracción utilizando LSBI no demostró ser demasiado problemática. Quizás fue porque realizamos su implementación luego de haber desistido temporalmente con la depuración de la funcionalidad de embedding, por lo que teníamos un mejor conocimiento del algoritmo y no teníamos que lidiar con una implementación rota. Nuevamente arrancar de cero a veces demuestra ser mejor que intentar aplicar parches sobre código roto.

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

Futuras Mejoras:

Hay algunas cuestiones que en el caso de seguir desarrollando el **stegobmp**, estaria bueno agregar, entre ellas estan la capacidad de soportar bmps comprimidos y una funcionalidad que te intente detectar por su cuenta como se cree que esta esteganografiado el archivo y con que método de encripción es que se encriptó (en el caso de estar encriptado).