ESTEGANOGRAFÍA

CRIPTOGRAFÍA Y SEGURIDAD (72.44)
GRUPO LOS CRIPTÓGRAFOS

Alumnos:

LAUTARO PINILLA	57504
BRIAN AIL	49254
MICAELA BANFI	57293

Instituto Tecnológico de Buenos Aires
ITBA
25-06-2020

Contenidos

1 Introducción

En el siguiente informe se presentará el trabajo práctico sobre esteganografía para la materia criptografía y seguridad.

1.1 Objetivos

Realizar un programa que cumpla los siguientes requisitos

- Oculte un archivo cualquiera en un .bmp, mediante un método de esteganografiado elegido, con o sin algún método extra de encriptación.
- Descubrir archivos ocultos en un .bmp que haya sido previamente esteganografiado

También estegoanalizar un archivo .bmp, determinar que tipo de archivo tiene un su interior, con que algoritmos fue estenografiado y extraerlo adecuadamente

1.2 Esteganografía

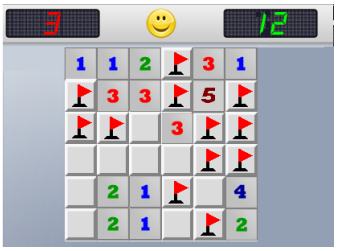
La esteganografía es la ciencia que se ocupa de la manera de ocultar un objeto, tanto un archivo como cualquier tipo de información, en algún otro objeto. Para ello se utilizan dos elementos: el objeto que queremos ocultar y el objeto en donde lo ocultaremos, al cual llamaremos portador. También se requiere de un algoritmo para realizar la esteganografia, es decir, la manera en la cual el objeto a ocultar se esconderá en el portador, y si fuera necesario, algún tipo de clave para su desencripción. La esteganografía logra que la información pase completamente desapercibida al ocultar su existencia misma. En general se utilizan archivos multimedia como portadores, aunque existen otras posibilidades.

2 Procedimiento realizado para descubrir objetos ocultos

Se trabajara fundamentalmente con 4 imágenes .bmp provistas por la cátedra, las cuales 3 tienen información oculta con alguno de los algoritmos de esteganografiado y una con algoritmo desconocido. El trabajo se realizo en lenguaje C, ya que consideramos que el manejo de bits iba a ser mas fácil, y se usaron librerías externas de openssl. Se fueron probando los diferentes algoritmos en las 4 imágenes, hasta obtener algo coherente.

2.1 Imagen roma.bmp

Esta imagen esta esteganografiada con LSB1 y tiene oculta otra imagen la cual llamaremos extract roma.png



Notamos que habían unos colores raros cerca del numero 12, entonces al mirar el código hexadecimal reconocimos la firma digital de un archivo PKI, la cual daba un indicio a que había algo zipeado dentro del buscaminas. Al deszipearlo se obtuvo el texto sol5.txt, con el siguiente contenido

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

Se completo el buscaminas, para verificar si había mas minas, y se obtuvo lo siguiente

Se armo una tabla para poder averiguar el modo y el algoritmo de la siguiente manera

Bytes	Valor Decimal	Letra
01000100	68	D
01100101	101	e
01110011	115	s
01000011	67	c
01101110	102	f
01100010	98	b

Ejecutando `strings back.bmp` encontramos "la password es camuflado". Entonces obtuvimos que hay un .wmv oculto con encripción des cfb y contraseña camuflado en alguno de los otros archivos.

2.2 Imagen Budapest.bmp

Esta imagen esta esteganografiada con LSBI y oculta un .pdf con la siguiente información

al .png cambiarle la extension por .zip y descomprimir

Lo cual no nos fue de mucha ayuda ya que lo habíamos descubierto con roma.bmp

2.3 Imagen Back.bmp

Como se explico en la sección de roma.bmp, haciendo `strings back.bmp` o abriendo el archivo con un editor hexadecimal se pudo obtener la contraseña

```
0044AA40   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
0044AA50   00 00 00 00  00 00 6C 61  20 70 61 73  73 77 6F 72  ......la passwor
0044AA60   64 20 65 73  20 63 61 6D  75 66 6C 61  64 6F        d es camuflado
---  back.bmp       --0x44AA6E/0x44AA6E----------------------------------------
```

2.4 Imagen Hugo4.bmp

Con la información obtenida con las imágenes anteriores, al desencriptar hugo4.bmp con lsb4, modo cfb, algoritmo des, contraseña "camuflado" encontramos un vídeo magia, el que habla sobre tener un archivo mas grande de lo que debería, por ende, puede tener información oculta.

3 Cuestiones a analizar

3.1 Discutir los siguientes aspectos relativos al documento

a. Organización formal del documento.

Hay ciertos detalles de la organización que se pueden discutir:

1. La escritura del documento en dos columnas hace un poco confusa la lectura, aunque esto puede ser mas bien una opinión personal. En especial al estar mezclada con gráficos.
2. Volviendo a los gráficos en particular, son altamente confusos ya cuando sacan flechas hacia un numero, que sigue en otra pagina, se hace muy complicado de seguir.

3. La organización en si no parece mal, introduce el tema, explica el algoritmo y finalmente pone las conclusiones,

b. La descripción del algoritmo.

La descripción del algoritmo es clara. En particular parecía ser suficiente con las secciones 2.1 Embedding stage y 2.2 Extracting stage. La descripción del algoritmo RC4 es un poco mas complicada y no se entiende bien. La parte de implementación parece un poco innecesaria, en especial porque en gran parte son fotos de una aplicación, que no se pueden ni leer. Como mencionamos en el punto a. los gráficos son poco claros y difícil de seguir.

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

No hemos notado error o contradicción alguna. La única parte que no se si es un error o mala interpretación, pero el gráfico de la Fig. 1 da a entender que en el proceso de Decode, se puede extraer la imagen portadora original. Lo cual a nuestro entender esto no podría ser. (Usa una linea punteada para describir esto, y no se entiende bien que significaría). El gráfico de encripción RC4 en particular es un poco confuso. Donde dice "Find J" en la fig.3, y no explica que significa esta "J" ni de donde salió.

3.2 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.

Aspectos	LSB1	LSB4	LSBI
Fácil de implementar	Sencillo	Sencillo	Más complicado
Espacio requerido	Requiere mucho espacio ya que utiliza 8 bytes por cada byte de la imágen	Al poder insertar 4, se reduce considerablemente la cantidad de bytes requeridos	Como inserta de a 1 bit, requiere el mismo espacio mínimo que LSBI
Velocidad	Velocidad normal	Notablemente más rápido que lsb1	El más lento de todos ya que por cada byte a insertar se debe calcular el salto
Detección a la vista humana	Dado a que inserta 1 bit cada vez, no se altera tanto la imágen por lo que es imperceptible a la vista humana	Dado a que inserta 4 bits cada vez, se puede llegar a detectar a simple vista	Dado a que inserta 1 bit cada vez y encima de a saltos, el más imperceptible, sobre todo si seguimos las especificaciones del *paper* que indica que se obtienen mejores resultados con imágenes a blanco y negro.
Detección automática	Debido a que es el más común y popular, es fácil de detectar por herramientas automáticas	Si bien no estan popular como LSB1, también existen herramientas automáticas para detectarlo	Debido a los saltos que realiza, no es detectable por los programas mencionados anteriormente a menos que sea un programa específico para detectar esto, aún así, el requerimiento del *paper* indica que la información siempre debe estar encriptada con RC4 por lo que aun así la herramienta tendría que poder desencriptar el resultado también

3.3 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.

Explicado en el punto 3 del informe.

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

Explicado en el punto 3 del informe. Todos los mensajes tenían información extra oculta.

3.5 Uno de los archivos ocultos era una porción de un vídeo, donde se ve ejemplificado una manera de ocultar información ¿cuál fue el portador?

En el vídeo comienzan hablando acerca de un mail, así que se entiende que este es el archivo que no tiene el tamaño que debería, por ende, esta ocultado información.

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

Fueron encontrados dos métodos que no eran los mencionados. Por un lado se oculto texto plano en un archivo binario (en nuestro caso, en el archivo back.bmp se oculto el texto "la password es camuflado"). Este método no es para nada seguro, ya que le permite al atacante recuperar la información muy fácilmente, utilizando un editor de texto hexadecimal. En nuestro caso fue muy simple encontrar la información ya que estaba legible al ojo humano. Podría complicarse un poco si el texto estaba ocultado en ciertas secuencias, y no todo junto. Otro método fue agregar información extra al final de un archivo (el .zip en lo que extraímos de roma.bmp). Es un poco mas seguro que el anterior pero sigue siendo no muy eficaz, ya que es fácil notar que el tamaño del header no coincide con el tamaño del archivo.

3.7 Para la implementación del algoritmo del documento de Juneja y Sandhu, se tomó como clave RC4 los primeros píxeles de la imagen portadora. ¿de qué otra manera podría considerarse o generarse o guardarse la clave RC4?

Para el correcto funcionamiento de RC4 se debe partir de una clave secreta compartida por emisor y receptor. El algoritmo admite claves de hasta 256 bytes de longitud,

típicamente se utilizan claves entre 40 y 128 bits. Podría utilizarse cualquier combinación de bytes, siempre y cuando sea secreta y tanto el emisor como el receptor sepan cual es. La clave podría ubicarse en cualquier lugar a convenir entre las dos partes, como por ejemplo, al inicio o al final de lo que se busca ocultar y no tiene porque ser de tamaño 6 bytes.

3.8 Según el libro de Katz, hay una forma más segura de usar RC4. ¿se podría implementar en este algoritmo LSBI?

Katz dice que RC4 puede considerarse seguro si los 1024 bits iniciales de la salida se descartan. Si en nuestro caso descartamos los 1024 bits iniciales, lo que se va a esteganografiar no seria el documento completo que se quiere ocultar, por ende no serviria.

3.9 ¿por qué la propuesta del documento de Juneja y Sandhu es realmente una mejora respecto de LSB común?

LSB común no encripta el texto, por lo que cualquier atacante que sospeche de que haya un texto estenografiado con LSB, puede recuperar el texto original, encriptar con RC4 le agrega una capa extra de seguridad, en caso de que un atacante detecte que una imagen tiene un texto escondido, de todas formas necesita poder desencriptarlo. En cuanto a la diferencia con ir secuencialmente haciendo el estenografiado byte a byte, sin usar el "Hops" que define el documento. Si bien no esta explicado en el documento, hace que el texto escondido quede mejor distribuido en la imagen portadora, esto podría hacer que sea mas difícil de detectar que una imagen tenga texto escondido adentro.

3.10 En el documento, Juneja y Sandhu indican que la inserción de los bits en la imagen es aleatoria. ¿es realmente así? ¿de qué otra manera podría hacerse los “saltos” de inserción de bits?

No es aleatoria, de hecho un simple programa podría agarrar cualquier imagen y probar extraer un texto usando los distintos 'Hops' que define el documento, el texto escondido siempre queda con los mismos patrones, definidos por el Hops.

Si se usa alguna variación del Hops, o sea, ir teniendo un offset de X bytes y volver a empezar desde el principio cuando se llega al final, por fuera de los múltiplos de 2 definidos en el documento, se podría detectar de la misma forma que nombramos en el punto anterior. Lo que se podría hacer quizás es una especie de lista encadenada donde se usen los N bytes menos significativos de la imagen portadora para ver el offset del próximo byte, pero puede que esto traiga problemas de coaliciones o algo así. De todas formas, el objetivo de los saltos es la distribución del texto escondido en la imagen para que sea difícil de detectar por un ojo humano o por algún algoritmo de ML, pero cuando un atacante ya sabe

que hay algo estenografiado, no importa mucho como se randomice este Hops, el estenografiado ya perdió su efecto.

3.11 ¿Qué dificultades encontraron en la implementación del algoritmo?

Una de las mayores dificultades ocurrieron al implementar lsbi ya que al principio teníamos algunas dudas sobre la forma en la que se recorría la imágen, si los saltos eran dados de a bytes o pixels y cómo recorrer luego el mismo recorrido para extraer la imágen.

También tenemos que sumar la complejidad para testear manualmente casos reales, lo que llevó a dificultar el debugging, por lo que cada error a la hora de estenografiar con cualquier método lsb, lleve un gran tiempo para arreglar. Para complementar esto, nos enfocamos en hacer un testeo unitario extenso sobre las funciones auxiliares de estenografiado para asegurar que su funcionamiento sea el correcto. Además, al implementar los estenografiados con threads, se agregó un nivel extra de complejidad a la hora de debuggear, lo que fue otro de los motivos de nuestro extenso testeo unitario.

Otra de las complicaciones que tuvimos fue a la hora de agarrar la extensión de los archivos a extraer, sobre todo si el archivo estaba previamente encriptado, para esto, debatimos en el grupo si no era más fácil asignar al archivo final la extensión en base a la firma digital del mismo (ya que nos manejábamos con un conjunto de tipos de archivos acotado).

3.12 ¿Qué mejoras o futuras extensiones harías al programa stegobmp?

1. Como extensión primaria, creemos que el programa final sería más eficiente si utilizara threads no solo a la hora de estenografiar sino a la hora de extraer.
2. Buscar formas de optimizar el algoritmo que realiza lsbi
3. Nos quedo a mitad de hacer, pero dada una password, probar automáticamente con todos los algoritmos y modos para desencriptar, sin necesitar saberlos de antemano o probar con todos hasta encontrar algo lógico
4. En la practica, la librería de encripcion de C tira unos errores que crashean al programa, y no llegamos a implementarle una vuelta a esto para poder ciclar con todos los métodos.