Criptografía y Seguridad *(72.04)*

TRABAJO PRÁCTICO DE IMPLEMENTACIÓN: SECRETO COMPARTIDO EN IMÁGENES CON ESTEGANOGRAFÍA

1 Objetivos

- Introducirlos en el campo de la criptografía visual y sus aplicaciones, a través de la implementación de un algoritmo de Secreto Compartido en Imágenes.
- Introducirlos en el campo de la esteganografía y sus aplicaciones.
- Implementar y analizar un algoritmo descripto en un documento científico.

2 Consigna

Realizar un programa en **lenguaje C** que implemente el algoritmo de Secreto Compartido en Imágenes descripto en el documento "**Secret Image Sharing**" cuyos autores son Chih-Ching Thien y Ja-Chen Lin de la universidad de Nacional Chiao Tung, de Taiwan.

El programa permitirá:

1) Distribuir una imagen secreta de extensión ".bmp" en otras imágenes también de extensión ".bmp" que serán las sombras en un esquema (k, n) de secreto compartido.
2) Recuperar una imagen secreta de extensión ".bmp" a partir de k imágenes, también de extensión ".bmp"

3 Introducción

La ***criptografía visual*** es un concepto introducido en 1994 por Adi Shamir y Moni Naor. En su presentación en EUROCRYPT'94 ellos consideran un nuevo tipo de esquema criptográfico que puede decodificar imágenes secretas sin usar cálculos criptográficos clásicos. En esencia, el sistema que ellos idearon era una extensión del concepto de ***esquemas de secreto compartido***, pero aplicado a imágenes. Las imágenes que tenían la información secreta, distribuida de manera segura, se podían luego superponer para recuperar la imagen secreta.

El concepto de Esquema de Secreto Compartido, también fue, en parte, idea de Shamir. Adi Shamir y George Blakley conciben en 1979, aunque en forma separada, el concepto de Secreto Compartido como una manera de proteger claves.

Tanto Shamir como Blakley exponen que guardar la clave en un solo lugar es altamente riesgoso y guardar múltiples copias en diferentes lugares sólo aumenta la brecha de seguridad. Shamir, por ejemplo, concluye que el secreto (***D***) deberá dividirse en un número fijo de partes (D_1, D_2, ..., D_n) de forma tal que:

1. Conociendo un subconjunto de ***k*** cualesquiera de esas partes se pueda reconstruir ***D***.
2. Conociendo un subconjunto de ***k-1*** cualesquiera de esas partes el valor ***D*** quede ***indeterminado***.

El documento de Blakley describe una forma de lograr el objetivo de distribuir las sombras de la manera exigida, utilizando conceptos de ***geometría proyectiva***.

El documento que se pide implementar en este trabajo práctico propone un esquema para compartir una imagen secreta basado en el método de Shamir. Para lograr que la imagen que se oculta en las sombras sea prácticamente imperceptible, en el documento se menciona la posibilidad de hacer uso de métodos de ocultamiento, es decir, de ***esteganografía***.

La ***esteganografía*** (del griego στεγανος *steganos, encubierto u oculto* y γραπηος *graphos, escritura*) es la ciencia que se ocupa de la manera de **ocultar** un mensaje.

La existencia de un mensaje u objeto es ocultada dentro de otro, llamado **portador**. El objetivo es proteger información sensible, pero a diferencia de la criptografía que hace ininteligible dicha información, la esteganografía logra que la información pase completamente desapercibida al ocultar su existencia misma.

La criptografía y la esteganografía se complementan. Un mensaje cifrado mediante algoritmos criptográficos puede ser advertido por un intruso. Un mensaje cifrado que, además, ha sido ocultado mediante algún método de esteganografía, tiene un nivel de seguridad mucho mayor ya que los intrusos no

pueden detectar su existencia. Y si por algún motivo un intruso detectara la existencia del mensaje, encontraría la información cifrada.

4 Detalles del sistema

4.1 Generalidades

El programa debe recibir como parámetros obligatorios:[1]

- **-d** o bien **–r**
- **-secret** ***imagen***
- **-k** ***número***

Y los siguientes parámetros opcionales:

- **<-n** ***número*** **>**
- **<-dir** ***directorio*****>**

Significado de cada uno de los parámetros obligatorios:

- **-d**: indica que se va a distribuir una imagen secreta en otras imágenes.
- **–r**: indica que se va a recuperar una imagen secreta a partir de otras imágenes.
- **-secret** ***imagen***: El nombre ***imagen*** corresponde al nombre de un archivo de extensión .bmp. En el caso de que se haya elegido la opción (-d) éste archivo debe existir ya que es la imagen a ocultar. Si se eligió la opción (-r) éste archivo será el archivo de salida, con la imagen secreta revelada al finalizar el programa.
- **-k** ***número:*** El número corresponde a la cantidad mínima de sombras necesarias para recuperar el secreto en un esquema (k, n).

Significado de cada uno de los parámetros opcionales:

- **<-n** ***número*** **>:** El número corresponde a la cantidad total de sombras en las que se distribuirá el secreto en un esquema (k, n). Sólo puede usarse en el caso de que se haya elegido la opción (-d). Si no se usa, el programa elegirá como valor de n la cantidad total de imágenes del directorio.
- **<-dir** ***directorio*****>** El directorio donde se encuentran las imágenes en las que se distribuirá el secreto (en el caso de que se haya elegido la opción (-d)), o donde están las imágenes que contienen oculto el secreto (en el caso de que se haya elegido la opción (-r)). Si no se usa, el programa buscará las imágenes en el directorio actual.

Ejemplos:

- Ocultar la imagen “clave.bmp”, en un esquema (2, 4) buscando imágenes en el directorio “varias”

```
$visualSSS –d –secret clave.bmp –k 2 –n 4 –dir varias
```

- Ocultar la imagen “clave.bmp”, en un esquema que use k = 3 buscando imágenes en el directorio actual.

```
$visualSSS –d –secret clave.bmp –k 3
```

- Recuperar la imagen “secreta.bmp”, en un esquema (2, 4) buscando imágenes en el directorio “varias”

```
$visualSSS –r –secret secreta.bmp –k 2 –n 4 –dir varias
```

- Recuperar la imagen “secreta.bmp”, en un esquema que use k = 3 buscando imágenes en el directorio actual.

```
$visualSSS –r –secret secreta.bmp –k 3
```

4.2 Algoritmo de Distribución

En la distribución hay que tener en cuenta los siguientes aspectos:

[1] Respetar el orden de los parámetros.

4.2.1 Valor de k

El valor de k debe ser mayor o igual que 2 y menor o igual que n.

4.2.2 Valor de n

El valor de n será de, mínimo 2.

4.2.3 Imagen secreta

La imagen secreta debe ser de formato BMP, de 8 bits por píxel. (1 byte = 1 pixel)

El formato BMP es un formato de archivos **binario** de imagen bastante simple. Consta de dos partes:

i. encabezado → de 54 bytes

ii. Cuerpo → de tamaño variable.

El encabezado contiene información acerca del archivo: tamaño de archivo, ancho de imagen, alto de imagen, bits por píxel, si está comprimido, etc

IMPORTANTE: Leer bien el valor que indica en qué offset empieza la matriz de píxeles, ya que puede comenzar inmediatamente después de los 54 bytes del encabezado, o bien empezar más adelante.

En el cuerpo del archivo bmp, están los bits que definen la imagen propiamente dicha. La imagen se lee de abajo hacia arriba y de izquierda a derecha. Si la imagen es de 8 bits por píxel, es una imagen en tonos de grises: el píxel de valor 0x00 es de color negro y el píxel 0xFF es de color blanco.

Tener cuidado al elegir la imagen: revisarla con algún editor hexadecimal para asegurarse que no tenga información extra al final (metadata) y que se ajuste al formato que se pide.

Si la imagen tiene píxeles en el rango [251,255], se modificarán mediante truncado, tal como lo indica la sección 3.1 del documento.

4.2.4 Permutación de los píxeles de la imagen

Para generar la permutación de los píxeles de la imagen, se utilizará el algoritmo de Durstenfeld:

```
for i from n − 1 downto 1 do
      j ← random integer with 0 ≤ j ≤ i
      exchange a[j] and a[i]
```

Se utilizarán las funciones rand() y srand() de la librería estándar de C y las siguientes:

```
void randomize(int num)
{
srand((int) num);
}

double randnormalize(void)
{
return rand()/((double)RAND_MAX+1);
}

long int randint(long int max)
{
return (long int) (randnormalize()*(max + 1)); /*devuelve un número en [0,max]*/
}
```

4.2.5 Imágenes Portadoras y ocultamiento por esteganografía

Las imágenes portadoras deben ser de formato BMP, de 8 bits por píxel.

Esquema (8,n)

En el caso de que el valor de k sea igual a 8, las imágenes deberán tener igual tamaño (ancho y alto) que la imagen secreta. Si no tienen n imágenes que cumplan esa condición, se muestra mensaje de error y no se realiza nada.

El ocultamiento de la información se hará mediante el método de LSB replacement (Least Significant Bit – Reemplazo del bit menos significativo). Esto se hará en el orden en que se tengan los bytes a partir del primer píxel (tener en cuenta el offset) y considerando los bits de mayor a menor.

Así, suponiendo que el primer valor a ocultar fuera el 0xD1 (**1101 0001**)

Y suponiendo que el primer píxel comienza en el offset 1078:

	Valor actual	Ultimos 4 bits	Valor después	Ultimos 4 bits
Byte 1078	ED	1101	ED	110**1**
Byte 1079	A4	0100	A5	010**1**
Byte 1080	45	0101	44	010**0**
Byte 1081	36	0110	37	011**1**
Byte 1082	3A	1010	3A	101**0**
Byte 1083	3A	1010	3A	101**0**
Byte 1084	3A	1010	3A	101**0**
Byte 1085	39	1001	39	100**1**

La semilla de generación de números aleatorios será un número entero de 2 bytes y debe ocultarse en los bytes 6 y 7 del archivo bmp (sección de bytes reservados).

Así, si el número es 641 (0000 0010 1000 0001) se guardará [2]

	Valor actual	Valor después
Byte 6	00	81
Byte 7	00	02

El número de orden correspondiente a la sombra (es decir, si la sombra es la número 1, 2, 3, …k) deberá ocultarse en los bytes 8 y 9 del archivo bmp (sección de bytes reservados).

Así, si la sombra es la tercera (0000 0000 0000 0011) se guardará

	Valor actual	Valor después
Byte 8	00	03
Byte 9	00	00

Esquema (k,n) con k distinto de 8.

En el caso de que el valor de k sea distinto de 8, queda a criterio del grupo definir (justificadamente) el tamaño de las imágenes portadoras y el método de ocultamiento.

4.3 Algoritmo de Recuperación

4.3.1 Valor de k

El valor de k debe ser mayor o igual que 2.

4.3.2 Imagen Secreta

Esquema (8,n)

La imagen secreta se tendrá que generar del mismo tamaño que las imágenes portadoras. Para armar su encabezado, se puede tomar el encabezado de cualquiera de las imágenes portadoras.

Esquema (k,n) con k distinto de 8

Cada grupo deberá determinar cómo se regenera la imagen secreta (en correspondencia a lo establecido en el item 4.2.5)

4.3.3 Imágenes portadoras

Esquema (8,n)

Las imágenes portadoras debe ser de formato BMP, de 8 bits por píxel y todas del mismo tamaño (ancho y alto) entre sí. Si no se tienen 8 imágenes que cumplan esta condición, se muestra mensaje de error y no se realiza nada.

[2] Tener cuidado con codificacion Little/Big Endian. Tomar los bits como se muesta en el ejemplo.

Esquema (k,n) con k distinto de 8

Cada grupo deberá determinar cómo validar las imágenes portadoras (en correspondencia a lo establecido en el item 4.2.5)

4.3.4 Permutación

Se verifica de manera inversa a la distribución.

4.3.5 Recuperación del secreto.

Se recomienda usar el método de Gauss y no el de determinantes para resolver el sistema de ecuaciones. Tener en cuenta que se está trabajando en aritmética entera módulo 251.

5 Cuestiones a analizar.

Deberán analizarse las siguientes cuestiones:

1. Discutir los siguientes aspectos relativos al documento de Thien y Lin.
    a. Organización formal del documento.
    b. La descripción del algoritmo de distribución y la del algoritmo de recuperación.
    c. La notación utilizada, ¿es clara? ¿cambia a lo largo del documento?
2. ¿Cómo se puede evitar el truncamiento de píxeles mayores que 250 según el documento? ¿Podría haber otra alternativa?
3. Explicar el criterio utilizado para elegir imágenes portadoras en el caso de k distinto de 8. Indicar si hubo propuestas previas que fueron descartadas.
4. Explicar el criterio utilizado para ocultar las sombras en el caso de k distinto de 8. Indicar si hubo propuestas previas que fueron descartadas.
5. Discutir los siguientes aspectos relativos al algoritmo implementado:
    a. Facilidad de implementación
    b. Posibilidad de extender el algoritmo para que se usen imágenes en color.
6. ¿Qué dificultades tuvieron en la lectura del documento y /o en la implementación?
7. ¿Qué extensiones o modificaciones harían a la implementación o al algoritmo?
8. ¿En qué situaciones aplicarían este tipo de algoritmos?

6 Organización de los grupos

El trabajo será realizado en grupos, según la agrupación formada para realizar el trabajo práctico 1.

7 Entrega

La fecha de entrega es el día 22 de junio.

Cada grupo enviará por mail a la cátedra el archivo con el proyecto realizado en C, junto con la documentación correspondiente al uso del programa.

Además presentarán un informe **impreso** con la solución correspondiente a la recuperación del secreto a partir de los archivos que se le entregaran oportunamente al grupo y el detalle de lo analizado en el punto 5 (Cuestiones a analizar). Este informe se presenta durante la misma clase del 16 de junio.

8 Sobre los archivos a entregar por mail.

- El entregable debe ser un archivo comprimido cuyo nombre debe cumplir el formato: `grupoXX.(zip|tar.gz|rar)` donde XX es el numero de grupo.
- Debe respetar la estructura de carpetas:
    - **docs/** (Documentación e informe)
    - **src/** (Fuentes)
    - **README.txt** (en el root, **incluir comentarios pertinentes** para la ejecucion correcta de scripts y binarios asi como también dependencias de la aplicación)

        - Incluir makefile en el root. Debe generar sólo el binario a ejecutar. **No debe incluirse el binario en la entrega**.
- Excluir de la entrega:
    - Enunciado
    - Cualquier tipo de binario generado por el make.
    - Carpetas .svn y __MACOSX
    - Archivos de prueba entregados por la cátedra.
- Deben incluirse **únicamente los printf explicitados** en el enunciado. En caso de incluirse más printf que los especificados, deben ejecutarse únicamente especificando una opción de verbose.
- **Es condición necesaria de aprobación su correcto funcionamiento en entorno pampero de ITBA.**
- Debe respetarse la sintaxis de ejecución del enunciado. Respetar incluso las mayúsculas y minúsculas.
- Utilizar códigos de error correctos. Por ejemplo, utilizar EXIT_FAILURE y EXIT_SUCCESS de stdlib.h.
- El programa debe explicitar errores. Por ejemplo, si hubo un error en un parámetro de entrada, se debe informar al usuario su error e informar la sintaxis correcta.

9 Criterios de Aprobación

Para aprobar el trabajo, se tendrán en cuenta:

- Entrega en la fecha indicada.
- Que el programa pueda efectuar la distribución del secreto y la recuperación del mismo.
- Que el contenido del informe sea correcto y completo, esto es, que estén contestadas todas las cuestiones del punto 5.
- Que el archivo ejecutable y el código en C se ajusten a los requerimientos y a lo establecido en el apartado 8.

La nota se conformará en un 60% por el programa y en un 40% por el informe. Son obligatorios el informe y el programa.

Si el trabajo, presentado en la fecha 22 de junio, resultara luego desaprobado, se podrá recuperar una sola vez. El trabajo recuperado sólo podrá tener una nota máxima de 4 (cuatro)

Para la entrega, así como para cualquier inconveniente, los mails de contacto son:

- Ana Arias : ariasroigana@gmail.com
- Rodrigo Ramele: rramele@itba.edu.ar

10 Material de lectura:

- **Capítulo 15 de Computer Security - Art and Science, Matt Bishop, Addison-Wesley, 2004**
- **Capítulo 10 y 12 de Handbook of Applied Cryptography, Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone, CRC Press, 1997**
- **"Secret image sharing", de** Chih-Ching Thien y Ja-Chen Lin.
- **"Secreto Compartido", de** Ana María Arias Roig.

Sobre Criptografía Visual

- Página de Criptografía visual de Doug Stinson: http://cacr.uwaterloo.ca/~dstinson/visual.html
- **"Visual Cryptography"**, Moni Naor y Adi Shamir.

http://www.wisdom.weizmann.ac.il/~naor/PUZZLES/visual_pap.ps.gz

Sobre Formato BMP

- http://www.fileformat.info/format/bmp/corion.htm