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 "**Verifiable Image Secret Sharing Using Matrix Projection**" cuyas autoras son Nurfathiya Faradiena Azzahra y Kiki Ariyanti Sugeng de Universitas Indonesia, de Indonesia. La propuesta de las autoras es una mejora del esquema propuesto en "**An Image Secret Sharing Method**" de Li Bai, Saroj Biswas, de la Universidad Temple de Filadelfia.

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 γραπηοζ *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:[1]

- **-d** o bien **-r**
- **-s *imagenSecreta***
- **-m *imagenMarca***
- **-k *número***
- **-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.
- **-s *imagen***: El nombre ***imagen*** corresponde al nombre de un archivo de extensión .bmp. En el caso de que se haya elegido la opción (-d) este archivo debe existir ya que es la imagen a ocultar y debe ser una imagen en blanco y negro (8 bits por pixel) Si se eligió la opción (-r) éste archivo será el archivo de salida, con la imagen secreta revelada al finalizar el programa.
- **-m *imagen***: El nombre imagen corresponde al nombre de un archivo con extensión .bmp. En el caso de que se haya elegido la opción (-d) este archivo es una imagen en blanco y negro que servirá como "marca de agua" para verificar todo el proceso. Debe ser de igual tamaño que la imagen secreta. En el caso de que se haya elegido la opción (-r) este archivo es una imagen en blanco y negro que contiene la transformación de la imagen de "marca de agua".
- **-k *número:*** El número corresponde a la cantidad mínima de sombras necesarias para recuperar el secreto en un esquema (k, n).
- **-n *número***: El número corresponde a la cantidad total de sombras en las que se distribuirá el secreto en un esquema (k, n).
- **-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)). Debe contener imágenes de extensión .bmp, de 24 bits por pixel.

Ejemplos:

- Distribuir la imagen "Albert.bmp" con watermark "Paris.bmp" según esquema (4,8) guardando las sombras en imágenes del directorio "color280x440":

```
./ss –d –s Albert.bmp –m Paris.bmp –k 4 – n 8 –dir color280x440/
```

- Recuperar la imagen "secreto.bmp", con watermark "RW.bmp" (en el directorio color280x440/RW) en un esquema (4,8) buscando imágenes en el directorio "color280x440/

```
./ss –r –s secreto.bmp –m color280x440/RW/RW.bmp –k 4 –n 8 –dir color280x440/
```

4.2 Algoritmo de Distribución

El algoritmo propuesto en "Verifiable…" es una ampliación del propuesto en "An Image Secret…", así que sugerimos leer los dos documentos para entender todo mejor.

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

[1] Respetar el orden y sintaxis de los parámetros.

4.2.1 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. 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.

Como la imagen se va a subdividir en matrices disjuntas de tamaño nxn, el total de pixeles debe ser divisible por nxn.

4.2.2 Matrices S

Se obtienen directamente de la imagen. Para ello, se toma una secuencia de nxn pixeles (como se dijo anteriormente, al tomar imágenes de 8 bits, un pixel es un byte, asi que se toma una secuencia de nxn bytes). Y con ello se arma la matriz $n \times n$.

Ejemplo:

Si n = 4
y se leen los bytes [120, 121, 121, 120, 119, 118, 116, 115, 110, 120, 119, 118, 110, 100, 99, 98] entonces se arma la matriz:

$$S = \begin{bmatrix} 120 & 121 & 121 & 120 \\ 119 & 118 & 116 & 115 \\ 110 & 120 & 119 & 118 \\ 110 & 100 & 99 & 98 \end{bmatrix}$$

4.2.3 Matrices A

Se obtienen generando aleatoriamente secuencias de $n \times k$ valores en el rango [0, 251].

Esa matriz A debe cumplir tres condiciones:

- ser de rango k.
- $\left(A^t A\right)$ debe ser inversible
- $proj(A)$ y $S - proj(A)$ no deben tener valores mayores que 251.

Si no cumple alguna de las condiciones anteriores, debe volver a generarse.

4.2.4 Matrices Sdoble y R

Se obtienen de operaciones matriciales con A, teniendo en cuenta que todas las operaciones son en Z_{251}

4.2.5 Matrices X

Las matrices X son n vectores $k \times 1$ que deben ser linealmente independientes.

Para asegurar que son linealmente independientes, se generarán con el siguiente método:

a. Elegir un número $a \in Z_{251}$

b. Generar la secuencia $a^0, a^1, ...a^{k-1}$ y guardarla en la matriz X.

Cada matriz X debe generarse a partir de un valor $a \in Z_{251}$ distinto, para asegurar la independencia lineal.

4.2.6 Matrices V

Se obtienen de operaciones matriciales con A y los vectores X teniendo en cuenta que todas las operaciones son en Z_{251}

4.2.7 Matrices G

Se obtienen de operaciones matriciales con los valores de R y los valores de los vectores V teniendo en cuenta que todas las operaciones son en Z_{251}

4.2.8 Matrices W

Se obtienen directamente de la imagen marca. Para ello, se toma una secuencia de nxn pixeles y con ello se arma la matriz $n \times n$.

4.2.9 Matrices Rw

Se obtienen de operaciones matriciales con los valores de W y Sdoble teniendo en cuenta que todas las operaciones son en Z_{251}.

4.2.10 Matrices Sh

Se obtienen anexando los valores de las matrices V y las G.

4.2.11 Valores de k y de n

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

Pero para poder ocultarlo fácilmente por esteganografía, usaremos sólo estas dos variantes:

- $k = 2 \rightarrow n = 4$
- $k = 4 \rightarrow n = 8$

4.2.12 Imágenes Portadoras y ocultamiento por esteganografía

Las imágenes portadoras deben ser de formato BMP, de 24 bits por píxel.

En cada una de las imágenes portadoras se ocultarán las sombras obtenidas (matrices Sh) correspondientes a cada participante.

Esquema (4, 8)

En el caso de que el valor de k sea igual a 4 y n es 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
Pixel 0	Byte 1078	ED	1101	ED	110**1**
	Byte 1079	A4	0100	A5	010**1**
	Byte 1080	45	0101	44	010**0**

Pixel 1	Byte 1081	36	0110	37	0111
	Byte 1082	3A	1010	3A	1010
	Byte 1083	3A	1010	3A	1010
Pixel 2	Byte 1084	3A	1010	3A	1010
	Byte 1085	39	1001	39	1001

El número de orden correspondiente a la sombra (es decir, si la sombra es la número 1, 2, 3, …k) deberá ocultarse en el primer byte reservado (byte 6) 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 6	00	03

Esquema (2,4)

En el caso de que el valor de k sea igual a 2 y n es 4, 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 **LSB2** replacement (Least Significant Bit – Reemplazo de **los dos bits** 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
Pixel 0	Byte 1078	ED	1101	EF	1111
	Byte 1079	A4	0100	A5	0101
	Byte 1080	45	0101	44	0100
Pixel 1	Byte 1081	36	0110	35	0101

4.2.13 Imagen de Marca de Agua

Los bytes obtenidos en Rw, como transformación de W a partir de Sdoble, se guardarán directamente en una imagen blanco y negro (de 8 bits por pixel) del mismo tamaño que la original. Es decir, quedará una nueva imagen visible pero con ruido.

4.3 Algoritmo de Recuperación

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

4.3.1 Imágenes portadoras

Esquema (4,8)

Las imágenes portadoras debe ser de formato BMP, de 24 bits por píxel y todas del mismo tamaño (ancho y alto) entre sí. Si no se tienen 4 imágenes que cumplan esta condición, se muestra mensaje de error y no se realiza nada.

Luego al resolver, tener en cuenta que se tiene que volver a armar con los bytes obtenidos una imagen en formato BMP pero de 8 bits por píxel.

Esquema (2,4)

Las imágenes portadoras debe ser de formato BMP, de 24 bits por píxel y todas del mismo tamaño (ancho y alto) entre sí. Si no se tienen 2 imágenes que cumplan esta condición, se muestra mensaje de error y no se realiza nada.

Luego al resolver, tener en cuenta que se tiene que volver a armar con los bytes obtenidos una imagen en formato BMP pero de 8 bits por píxel

4.3.2 Recuperación de R

Cada matriz R se obtiene a partir de los G extraídos de cada matriz Sh recuperada de las sombras.

En este paso hay que resolver sistemas de ecuaciones, se recomienda usar el método de Gauss y tener en cuenta que todas las operaciones son en Z_{251}.

4.3.3 Recuperación de Sdoble

Se obtiene a través de operaciones matriciales a partir de los vectores V extraídos de cada matriz Sh recuperada de las sombras.

Tener en cuenta que todas las operaciones son en Z_{251}.

4.3.4 Recuperación de S

Se obtiene a través de operaciones matriciales a partir de Sdoble y R obtenidos previamente.

Tener en cuenta que todas las operaciones son en Z_{251}.

Una vez recuperados todas las matrices S, se arma la imagen BMP teniendo en cuenta que es en blanco y negro, es decir 8 bits por pixel.

4.3.5 Recuperación de W

Se obtiene a través de operaciones matriciales a partir de Sdoble y Rw, habiendo obtenido previamente Rw de los bytes de la imagen de Marca de Agua.

Una vez recuperadas todas las matrices W, se arma la imagen BMP teniendo en cuenta que es en blanco y negro, es decir 8 bits por pixel.

5 Cuestiones a analizar.

Deberán analizar una serie de cuestiones que serán publicadas en mayo.

6 Organización de los grupos

El trabajo será realizado en grupos de, máximo, 3 integrantes.

7 Sugerencias

Se sugiere encararlo en forma modularizada, probando por separado las cuestiones:

- manejo de archivos bmp
- operaciones matriciales
- esquema propuesto a nivel matriz
- esteganografiado

Una vez que se aseguran que por separado funciona, integrarlo.

8 Entrega

La fecha de entrega es el día 24 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 entregarán 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 24 de junio.

9 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.

10 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 para los archivos entregados por la cátedra.**
- 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 9.

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 24 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, el mail de contacto es:

- Ana Arias : mroig@itba.edu.ar

11 Material de lectura:

- N F Azzahra, K A Sugeng, “Verifiable Image Secret Sharing Using Matrix Projection”, Journal of Physics: Conference Series, Volume 1108, conference 1 Indonesia, (2018) **Disponible en https://iopscience.iop.org/article/10.1088/1742-6596/1108/1/012082/pdf (visitado Febrero 2019)**
- Li Bai, S. Biswas, A. Ortiz, and D. Dalessandro, “An Image Secret Sharing Method”, Proceeding of 9th International Conference on Information Fusion, Italy, (2006), pp. 1-6. **Disponible en https://apps.dtic.mil/dtic/tr/fulltext/u2/a521717.pdf (visitado Febrero 2019)**
- Thien CC , Lin J C “Secret image sharing”, Computers & Graphics vol 26 n 5, (2002) pp765-770. **Disponible en https://mafiadoc.com/secret-image-sharing_59a26dfc1723dd0a40e07b3e.html (visitado Febrero 2019)**
- 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