

TRABAJO PRÁCTICO DE IMPLEMENTACIÓN: SECRETO COMPARTIDO 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.
- Introducirlos en el campo de la esteganografía y sus aplicaciones.
- Implementar y analizar un algoritmo descrito en un documento científico.

2 Consigna

Realizar un programa en **lenguaje C** que implemente el algoritmo de Imagen Secreta Compartida descrito en el documento “**Sistema de Imagen Secreta Compartida con Optimización de la Carga Útil**” cuyos autores son Angelina Espejel Trujillo, Iván Castillo Camacho, Mariko Nakano Miyatake y Héctor Pérez Meana del Instituto Politécnico Nacional de México.

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, \dots, 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 γραφηος *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 o camuflaje**. 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**
- **imagenSecreta**
- **número (k)**
- **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.
- **imagenSecreta**: 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) este archivo será el archivo de salida, con la imagen secreta revelada al finalizar el programa.
- **Número k**: El número corresponde a la cantidad mínima de sombras necesarias para recuperar el secreto en un esquema (k, n).
- **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 8 bits por pixel, de igual tamaño que la imagen secreta. Además, deberá verificarse que existan por lo menos k imágenes en el directorio.

Ejemplos:

- Distribuir la imagen “Albert.bmp” según esquema (4,8) guardando las sombras en imágenes del directorio “color280x440”:

```
./ss d Albert.bmp 4 color280x440/
```
- Recuperar la imagen “secreto.bmp”, en un esquema (4,8) buscando imágenes en el directorio “color280x440/”

```
./ss r secreto.bmp 4 color280x440/
```

4.2 Algoritmo de Distribución

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

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.

¹ Respetar el orden y sintaxis de los parámetros.

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 bloques de k bytes, el total de pixeles debe ser divisible por k.

4.2.2 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 valores de k entre 4 y 6.

Es decir, $k = 4, 5, 6$.

4.2.3 Imágenes Portadoras y ocultamiento por esteganografía

Para esta implementación, se usarán imágenes portadoras de formato BMP y de 8 bits por píxel, de igual tamaño que la imagen secreta. Para facilitar toda la operatoria, asumiremos que son de igual alto y ancho que la imagen secreta.

En cada una de las imágenes portadoras se ocultarán las sombras obtenidas.

Cada imagen portadora se divide en bloques de 2x2: XWVU.

Observando las ilustraciones del documento, vemos que los bytes que se tomarían para cada primer bloque son los de la parte superior izquierda de la imagen.

Imágenes Camuflaje



Sin embargo, en el archivo BMP esos bytes no son los que aparecen a partir del offset.

Suponiendo un archivo BMP de 6x8:

42	43	44	45	46	47
36	37	38	39	40	41
30	31	32	33	34	35
24	25	26	27	28	29
18	19	20	21	22	23
12	13	14	15	16	17
6	7	8	9	10	11
0	1	2	3	4	5

Enseguida después del offset, el byte (supongamos de valor 42) es el que se ve en la parte superior izquierda de la imagen. Si se tocan los bytes 0 a 5, se toca lo que se ve en la parte de debajo de la imagen.

Por lo tanto, el primer bloque

X	W
V	U

Tendrá los valores:

42	43
36	37

Que luego se puede trabajar como arreglo: [42,43,36,37]

Entonces, hay que tener cuidado de que si se levanta el bloque de pixeles tal como está en el archivo, se levantará:

[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47]

Pero se trabajará con los bloques así:

42	43	44	45	46	47
36	37	38	39	40	41
30	31	32	33	34	35
24	25	26	27	28	29
18	19	20	21	22	23
12	13	14	15	16	17
6	7	8	9	10	11
0	1	2	3	4	5

[[42,43,36,37], [44,45,38,39], [46,47,40,41], [30,31,24,25], [32,33,26,27], [34,35,28,29], [18,19,12,13], [20,21,14,15], [22,23,16,17], [6,7,0,1], [8,9,2,3], [10,11,4,5]]

Una vez modificados los bloques como indica el algoritmo propuesto, deberán volver a guardarse en el BMP en el orden correcto.

IMPORTANTE:

De la imagen secreta sólo se guardan los pixeles (no el encabezado).

Dichos pixeles se leen y guardan en orden [0, 1, 2, ...] (es decir como se presentan los bytes después del offset).

Es decir, si $k = 6$, y el offset empieza en 1078:

Secreto, bloque 0	
Pixel 0	Byte 1078
Pixel 1	Byte 1079
Pixel 2	Byte 1080
Pixel 3	Byte 1081
Pixel 4	Byte 1082
Pixel 5	Byte 1083
Pixel 6	Byte 1084

Con esos valores se obtiene $F(x)$.

Luego, de la portadora se usa:

Portadora i, Bloque 0	
Pixel 42	Byte 1120
Pixel 43	Byte 1121
Pixel 36	Byte 1114

El byte 1120 será X.

Se calcula $F(X) = y$, que se guarda en los otros bytes como indica el documento.

IMPORTANTE:

Si dos portadoras en el bloque j tienen el mismo valor de X, debe modificarse, ya que si se generan dos sombras iguales no podrá luego resolverse el sistema.

En caso de que lo anterior suceda, habrá que cambiar el X, sumándole 1, con la precaución de que al llegar al 255, se guarde 0 u otro valor conveniente.

No hacer esta operación con $GF(2^8)$, por lo siguiente:

Si X está repetido, se hace $X' = X+1$ (en $GF(2^8)$)

Si X' vuelve a estar repetido, y se vuelve a hacer $X'' = X'+1$ (en $GF(2^8)$) este X'' termina siendo igual al primer valor X, porque la suma en $GF(2^8)$ es un xor.

4.2.4 Campo de Galois

Todas las operaciones con $F(x)$ deben realizarse en un $GF(2^8)$ (Campo de Galois sobre 2^8). Esto permite que se puedan obtener resultados en el rango $[0,255]$ sin inconvenientes. En un documento aparte está la explicación sobre Campos de Galois, en qué consisten y cómo se realizan e implementan las operaciones.

IMPORTANTE:

Se usará como polinomio primitivo generador $m(x) = x^8+x^6+x^5+x^1+1$ (que equivale al número 355)

4.3 Algoritmo de Recuperación

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

4.3.1 Imágenes portadoras

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 k imágenes que cumplan esta condición, se muestra mensaje de error y no se realiza nada.

Luego al resolver, la imagen secreta conservará los datos de encabezado de cualquiera de las imágenes portadoras, siendo entonces de igual formato BMP, de 8 bits por píxel y de igual ancho y alto.

4.3.2 Recuperación del secreto

Se obtiene recuperando cada bloque XWVU para cada una de las k sombras y con los k pares (x, $F(x)$) se obtiene, mediante interpolación de Polinomios, el $F(x)$ que es el secreto y que se guardará en secreto.bmp.

Tener en cuenta lo indicado en 4.2. respecto del orden en que se leen los bytes de las portadoras.

IMPORTANTE:

Todas las operaciones se efectuarán en $GF(2^8)$ usando el polinomio generador $m(x) = x^8+x^6+x^5+x^1+1$ (que equivale al número 355)

La división en $GF(2^8)$ equivale a dividir por el inverso multiplicativo.

Conviene tener una tabla generada con los inversos para proceder con mayor rapidez.

5 Cuestiones a analizar.

Deberán analizarse las siguientes cuestiones:

1. Discutir los siguientes aspectos relativos al documento.
 - a. Organización formal del documento (¿es adecuada? ¿es confusa?)
 - b. La descripción del algoritmo de distribución y la del algoritmo de recuperación. (¿es clara? ¿es confusa? ¿es detallada? ¿es completa?)

- c. La notación utilizada, ¿es clara? ¿cambia a lo largo del documento? ¿hay algún error?
2. El título del documento hace referencia a que optimiza la carga útil ¿a qué se refiere? ¿Qué relación existe entre k y el tamaño de la portadora?
3. ¿Qué ventajas y qué desventajas ofrece trabajar en $GF(2^8)$ respecto de trabajar con congruencias módulo?
4. ¿se puede trabajar con otro polinomio generador? ¿podría guardarse como “clave”?
5. Según el documento se pueden guardar secretos de todo tipo (imágenes, pdf, ejecutables). ¿por qué? (relacionarlo con la pregunta 3)
6. ¿cómo podría adaptarse la implementación realizada para poder guardar un archivo de imagen completo? (es decir guardando encabezado y pixeles)
7. Analizar cómo resultaría el algoritmo si se usaran imágenes en color (24bits por píxel)
8. ¿se podrían tomar los bloques de otra manera, en lugar de como matrices 2×2 ? Explicar.
9. Discutir los siguientes aspectos relativos al algoritmo implementado:
 - a. Facilidad de implementación
 - b. Posibilidad de extender el algoritmo o modificarlo.
10. ¿En qué situaciones aplicarían este tipo de algoritmos?

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 en campos de Galois.
- Interpolación polinómica.
- esquema propuesto con bloques pequeños (Por ej.: $k = 4$, $n = 10$, secreto de 60 bytes)

Una vez que se aseguran de 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 o en Java), junto con la documentación correspondiente al uso del programa.

Además presentarán el mismo día un informe 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).

No se aceptará:

- Que no haya README con indicaciones de compilación / ejecución.
- Que no funcione en Pampero.
- Que no respeten el orden de los argumentos del programa.
- Que no funcione para los archivos que se les dé al grupo (es la manera de asegurar que respetaron el algoritmo propuesto)

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 ejecución correcta de scripts y binarios así 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.
- | |
|--|
| <ul style="list-style-type: none"> ● 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:

- Espejel, A., Castillo, I., Nakano, M. y Pérez H. "Sistema de Imagen Secreta Compartida con Optimización de la Carga Útil", Información Tecnológica Vol 25(4), La Serena, 2014. **Disponible en** <https://scielo.conicyt.cl/pdf/infotec/v25n4/art21.pdf> (Visitado Enero 2021)
- 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