Criptografía y Seguridad *(72.04)*

TRABAJO PRÁCTICO 2: 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 "**Improvements in Geometry-Based Secret Image Sharing Approach with Steganography**" cuyos autores son Mustafá Ulutas, Vasif V. Nabiyev y Guzin Ulutas de la universidad de Karadeniz, Turquía.

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***.

Hay una dualidad entre el esquema de Shamir y el de Blakley para cuando k es igual a 2.

En ese caso:

- El esquema de Shamir usa dos **puntos** como sombras, y la **recta** que incide en ellos es el secreto.
- El esquema de Blakley usa dos **rectas** como sombras, y el **punto** donde inciden es el secreto.

El documento que se pide implementar en este trabajo práctico propone un esquema para compartir una imagen secreta utilizando el método de Blakley. Para lograr que la imagen que se oculta en las sombras sea prácticaente imperceptible el algoritmo propuesto hace uso de la ***esteganografía***.

La ***esteganografía*** (del griego στεγανοζ *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.

En el algoritmo propuesto por Ulutas y sus colegas, la imagen secreta se distribuye según un esquema umbral (k, n) basado en Blakley utilizando esteganografía para que al ver las imágenes sombra no sea perceptible la existencia de información secreta oculta en las mismas.

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
```

[1] Respetar el orden de los parámetros.

- 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:

4.2.1 Valor de k

Sólo se tendrá en cuenta un valor de k igual a 2, 3 ó 4.

4.2.2 Valor de n

El valor de n será de, mínimo 3 y máximo 8.

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.

4.2.4 Imagenes portadoras

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

4.2.5 Ocultamiento por esteganografia

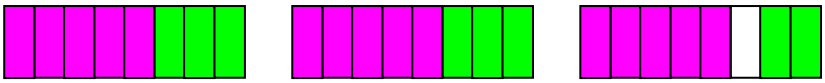
Como indica el documento, tanto las imágenes portadoras como la imagen secreta se parten en bloques de tamaño k.

Según ese valor k, se determina cuántos bits en cada píxel serán destinados a ocultar el valor B que resulta del cálculo de la sombra a partir del bloque secreto.

Así, si k es igual a 3, el conjunto B es $\{b_0 = 3, b_1 = 3, b_2 = 2\}$[2]

Si los píxeles del primer bloque están en el offset 1078:

[2] El documento aplica una función “ceil” para redondear el número b_i.



Byte 1078 Byte 1079 Byte 1080

A1 A2 A3

Bits para ocultar B

Al ocultar el valor B, se pondrán entonces los bits más significativos en la porción que corresponda en el primer byte, los siguientes en el segundo, y así. En este caso, que eran tres bytes, quedan los bits menos significativos de B en el tercer píxel.

4.2.6 Autenticación

El bit en blanco en el apartado anterior es el que corresponde al bit de autenticación.

Dicho bit se obtiene efectuando las fórmulas 3.6 y 3.7 del documento.

Para facilitar la tarea, se considerará que el hash es de la concatenación de bits en el siguiente orden:

Los bytes del bloque se toman en el mismo sentido a como se tomaron para ocultar B.

Los bits correspondientes al byte que debe contener el bit de hash (que en el proceso de ocultamiento era el primero) se toman así: primero los bits de a3, luego los bits del valor de B que se ocultó en ese píxel y el último bit en cero.

Ejemplo:

Bits del byte 1078 Bits del byte 1079 Bits del byte 1080

Luego se efectúa el hash (con MD5) de los kx8 bits concatenados (k bytes), obteniéndose 128 bits a partir de los que se calculará el bit de autenticación.

4.2.7 Revisión de las sombras obtenidas

Cada grupo de k píxeles de cada sombra tiene, como indica el documento, los coeficientes de una ecuación lineal. En la etapa de recuperación, al tomar un grupo de k píxeles de cada una de las k sombras elegidas, se podrá construir un sistema de k ecuaciones con k incógnitas.

Dicho sistema de ecuaciones siempre tiene solución (porque las ecuaciones fueron generadas de manera que la solución sea el grupo de píxeles de la imagen secreta). Sin embargo, dicho sistema puede ser indeterminado (tener infinitas soluciones)

Analizándolo desde el punto de vista geométrico, se entiende mejor.

Suponiendo que el esquema de reparto sea (3, n). Cada grupo de 3 pixeles contiene los coeficientes de un plano. Suponiendo que tomamos el primer bloque de 3 pixeles en las 3 imágenes sombras que se poseen y generamos un sistema de ecuaciones:

Primer bloque de S1: $a_{11}x + a_{12}y + a_{13}z = b_1$

Primer bloque de S2: $a_{21}x + a_{22}y + a_{23}z = b_2$

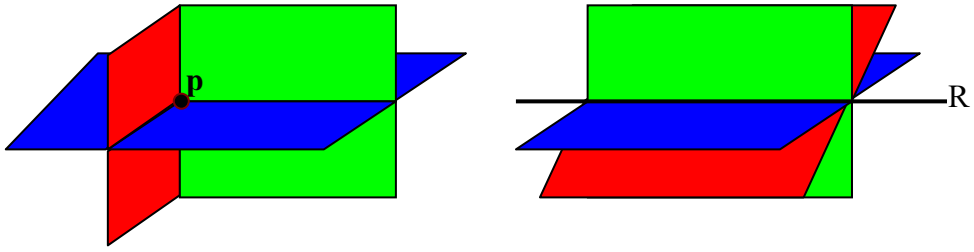
Primer bloque de S3: $a_{31}x + a_{32}y + a_{33}z = b_3$

Queda un sistema de 3x3:

$$\begin{cases} a_{11}x + a_{12}y + a_{13}z = b_1 \\ a_{21}x + a_{22}y + a_{23}z = b_2 \\ a_{31}x + a_{32}y + a_{33}z = b_3 \end{cases}$$

Si pensamos que son planos, por construcción son planos **no** paralelos.

Sin embargo, puede ocurrir:



$S1 \cap S2 \cap S3 = \{p\}$ $S1 \cap S2 \cap S3 = R$

Obviamente, interesa la solución que corresponde a un único punto, ya que, volviendo a nuestro problema, dicho único punto tiene los datos de las tres coordenadas del píxel secreto. La solución que corresponde a una recta, contiene al píxel secreto, pero no es posible determinarlo.

Cuando se hace el algoritmo de distribución, se obtienen n ecuaciones. Hay que revisar entonces que cualquier grupo de k de ellas sea linealmente independiente y tenga solución única.

Entonces, si hubiera alguna ecuación que, en combinación con alguna otra genera ese problema hay que cambiarla sutilmente. Este cambio debe ser hecho de manera que la imagen que contenga la sombra se altere lo menos posible.

Algunas recomendaciones para hacer esta verificación:

- trabajar con las ecuaciones implícitas (es decir, sin el valor del término independiente)
- usar el método de reducción de gauss, teniendo en cuenta que las operaciones son todas en aritmética entera módulo 251.
- Cuando se modifica algún píxel, se modifica primero la parte correspondiente a “Ai” y luego hay que recalcular b y el bit de autenticación.
- Al modificar el píxel, tener cuidado con los overflows (si un píxel era blanco (255) y le sumo 1, por overflow quedará un 0, es decir, negro). Tener cuidado porque no siempre el tope es 255, el tope puede ser menor, porque del píxel no se están usando todos los bis, sino una parte.
- Al modificar puede ser conveniente elegir aleatoriamente cuál píxel modificar y cuánto.

4.3 Algoritmo de Recuperación

4.3.1 Valor de k

Sólo se tendrá en cuenta un valor de k igual a 2, 3 ó 4.

4.3.2 Imagen Secreta

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.

4.3.3 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.

4.3.4 Autenticación

Se verifica de la misma manera que en la distribución.

4.3.5 Recuperación del secreto.

Si al intentar recuperar el secreto ocurriera que un sistema de ecuaciones tiene infinitas soluciones, deberá mostrarse que no es posible encontrar la solución. (Esto no debería ocurrir si la verificación al ocultar se hizo correctamente)

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 Ulutas y sus colegas:
    a. Organización formal del documento.
    b. La descripción del paso 7 del algoritmo de reconstrucción.
    c. La notación utilizada, ¿es clara? ¿cambia a lo largo del documento?
2. En el método original de secreto compartido de Blakley se descartan las sombras que tengan ceros. ¿por qué? ¿Por qué crees que Ulutas y sus colegas no tuvieron en cuenta eso?
3. Una vez recuperada la imagen secreta, ¿es esta imagen exactamente igual a la imagen ocultada?¿Por qué? (Tener en cuenta sólo la matriz de píxeles, no el encabezado).
4. 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.
    c. Ventajas respecto del algoritmo original de Shamir (mencionar por lo menos 2)
5. ¿Qué dificultades tuvieron en la lectura del documento y /o en la implementación?
6. ¿Qué extensiones o modificaciones harían a la implementación o al algoritmo?
7. ¿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 16 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 16 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**
- **"Improvements in Geometry-Based Secret Image Sharing Approach with Steganography", de** Mustafá Ulutas, Vasif V. Nabiyev, y Guzin Ulutas.
- **"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