# 72.04 Criptografía y Seguridad

Trabajo Práctico Especial

Secreto Compartido en Imágenes con Esteganografía


Alumnos

Alonso, Juan Manuel - 56080

Cifuentes, Ignacio Imanol - 54311

Cortés Rodríguez, Kevin - 54775

Docentes

Ing. Abad, Pablo

Lic. Roig, Ana

Dr. Ing. Ramele, Rodrigo

Índice

# Introducción

El siguiente informe tiene como objetivo detallar la implementación de un algoritmo de distribución de una imagen secreta en otras imágenes propuesto en los papers "*Verifiable Image Secret Sharing Using Matrix Projection*" de Azzahra y Sugeng, y "*An image secret sharing method*" de Li Bai y Biswas.

El objetivo del mismo es, en una primer parte, distribuir una imagen secreta en otras, con extensión .bmp. Luego, el sistema debería ser capaz de reconstruir de las otras k imágenes modificadas la imagen secreta.

# Descripción del algoritmo

Para que el usuario que quiera ocultar una imagen en otras, deberá ejecutar el código con algunos parámetros indispensables para hacer la distribución del secreto, y consecuentemente en otro momento, el recupero de la misma.

Uno de los parámetros necesarios es la de indicar un set de dos números **k** y **n**. El valor n es el número total de sombras* a las que se distribuirá el secreto. Mientras que k representa a la mínima cantidad de sombras necesarias para recuperar el secreto.

## Distribución

El usuario debe ingresar como parámetro que se quiere distribuir una imagen secreta en un esquema de **n** imágenes.
Para ello el algoritmo realiza los siguientes pasos:

1. Construcción de una matriz A random de n columnas x k filas.

2. Calcular la matriz Sd haciendo la proyección de A módulo 251* de la matriz A obtenida en el punto 1. Defínase la función proyección como:

$$proj(A) = A\,(A^t x\, A)^{-1} A^t \pmod{251}$$

3. Construir **n** vectores de k filas x 1 columna linealmente independientes. Dichos valores los guardaremos en una matriz V de k x n.

4. Calculamos los vectores shares v que serán obtenidas de haber hecho el producto vectorial de la matriz random A (punto 1) con cada uno de los n vectores obtenidos en el punto 3.

5. Calcular la matriz R Reminder, haciendo la resta de la matriz S (de n filas x n columnas, obtenidas directamente de la imagen secreta pasada por parámetro por parte del usuario) con la matriz Sd calculada en el punto 2. Dicha operación también será en módulo 251.

6. Mediante el esquema de secreto compartido propuesto por Thien y Lin, calcular las matrices G.

$$G_j = [g_1^{(j)} \;\; g_2^{(j)} \;\; \dots \;\; g_{ceil(n/k)}^{(j)} \;\; ]$$

$$g_t^{(j)}(i) \;=\; (\sum_{h=1}^{k} I(i, k*(t-1)+h) \;*\; c_j^{\,h-1}) \; (mod\; 251)$$

7. Obtener la imagen de marca de agua, pasada por parámetro por el usuario y calcular la matriz Reminder Rw que es la resta de la matriz W (marca de agua) con la matriz Sd (punto 2).

8. Obtener las matrices Sh (shadows - sombras), obtenidas mediante la concatenación de los vectores obtenidos en el punto 3 y las matrices G, del punto 6.

9. Publicar la matriz Rw, como si fuese una imagen de 8 bits por pixel (es decir, una imagen blanco y negro).

## Distribución de las matrices Sh en las imágenes

La distribución de las matrices Sh es distinta si tomamos el esquema de (k=2, n=4) ó (k=4, n=8).

Para ambos casos la idea es la de tomar cada uno de los valores de la matriz y persistir un bit (o dos bits, dependiendo el esquema) en cada uno de los bytes de la imagen original, pero no cualquier bit, sino el más insignificante, de forma que la diferencia sea imperceptible al ojo humano.
Por ejemplo, si tenemos que persistir el número X, debemos obtener el binario del mismo y persistir un bit en la última posición de cada byte de la imagen.
De ésta forma, persistir un byte en la imágen es equivalente a modificar 8 bytes de la imagen original.

Si hay una imagen de 280 x 440, tenemos en total 123.200 píxeles que persistir. Como cada píxel, modificará a 8 bytes para el caso n = 8, necesitaríamos a los 8 participantes para distribuir la imagen.

En el caso de que usemos el esquema de (k=2, n=4), en lugar de modificar un bit, modificaremos 2 bits (Least Significant Bit LSB2).

## Recuperación

El usuario debe escribir un comando especial (-r), haciendo referencia a que quiere recuperar una imagen secreta a partir de k participantes. Para ello, debe solicitar las k imágenes, la matriz Rw que supone ser pública y especificar el esquema (2,4) o (4,8).

Inicialmente el programa debe tomar todas las imágenes del directorio pasado por parámetro y obtener las matrices Sh almacenadas en ellas. Para eso, se procede a hacer el proceso de recuperación de bits por cada uno de los bytes de la imágen. Es sumamente importante conocer el esquema ya que deberá contemplar el caso de tomar 1 ó 2 bits.

Una vez que obtenemos las k matrices Sh, debemos obtener los vectores vi y matrices Gj. De todos los vectores vi, obtendremos una matriz B = [v1 v2 ... vj]. Luego podemos recuperar la matriz Sd producto de haber hecho la proyección de B (la misma con la que se hace referencia en el punto 2 de distribución).

Por otra parte, tenemos k matrices Gj de n filas x 2 columnas, siendo 2 el resultado de la operación $ceil(n/k)$. Con cada una de ellas ahora deberíamos poder reconstruir la matriz Reminder R aplicando operaciones de cálculo matricial (Gauss-Jordan) sobre la matriz de I desconocida:

$$\begin{pmatrix} c_1^0 & c_1^1 & \cdots & c_1^k \\ c_2^0 & c_2^1 & & c_2^k \\ c_3^0 & c_3^1 & \cdots & c_3^k \\ \cdots & \cdots & & \\ c_j^0 & c_j^1 & & c_j^k \end{pmatrix} x \begin{pmatrix} I(x,2(y-1)+1) \\ I(x,2(y-1)+2) \\ \cdots \\ I(x,2(y-1)+k) \end{pmatrix} = \begin{pmatrix} G_1(x,y) \\ G_2(x,y) \\ \cdots \\ G_j(x,y) \end{pmatrix}$$

La matriz de Cj ya es conocida, y el vector de $G_j(x,y)$ lo obtenemos de las imágenes de recuperación.

## Notación

La notación del proyecto es snake case y las funciones están divididos por funcionalidad, así sea de distribución o recupero.

**distribution.c**
Contiene a todas las funciones necesarias para comenzar la ejecución de la distribución de la imagen secreta según el esquema de Thien y Lin. Por otro lado, contiene a todas las funciones que generan las matrices descritas en el ítem de distribución previamente explicado en el informe.
Una vez que se generaron todas las matrices necesarias, se llama la

función principal “distribute”.

**recovery.c**
Contiene toda la lógica para hacer el proceso de recuperación de todas las matrices descritas en , en principal la Reminder R. Por otro lado, contiene el método de Gauss-Jordan, necesario para el recupero de la matriz previamente mencionada.

**global.h**
Únicamente para manejo de algunas variables importantes definidas por parámetro inicialmente

**io.c**
Utilizado inicialmente cuando el usuario especifica datos de los parámetros, directorios o nombres de archivos. Lee de los directorios los archivos .bmp, y en caso de que no existan, puede generarlo.

**matrix.c**
Operaciones de matrices en módulo 251. No contiene lógica de distribución ni recuperación.

**modular.c**
Calcula el módulo de un número en 251. Retorna un valor entre 0 y 250 inclusive. También calcula la matriz de inversos multiplicativos.

**random.c**
Genera números aleatorios criptográficamente seguros. Por otro lado, también puede devolver n números aleatorios entre 0 y 250 sin repetidos.

**bitmap.h**
Genera los tipos de estructura para procesar archivos BMP, considerando especialmente los casos de 8 bpp, donde se adiciona una serie de 1078 bytes de valores fijos, *RGBQUAD*.

# Preguntas conceptuales de Campus

## ¿Por qué la propuesta de Azzahra y Sugeng supone una mejora a la propuesta de Li Bai?

El esquema de secreto compartido es una técnica para compartir datos secretos en n piezas basadas en un esquema de (k, n) -especificados inicialmente del informe-. Sin embargo, el problema con el esquema de intercambio secreto de Shamir es que no proporcionan ninguna forma de verificar que la distribución fue correcta y que las acciones fueron válidas. Este problema también ocurre en el intercambio de imágenes de Thien y Lin o en el modelo de construcción utilizando la proyección de matrices de Li Bai. Por otro lado, este protocolo permite a cada participante validar su pieza recibida y confirmar la autenticidad del secreto. Por lo tanto, este documento discutió un esquema propuesto basado en el intercambio secreto verificable, en el que la proyección de matriz se utiliza para crear imágenes compartidas y una matriz pública a partir de la imagen de imágenes de marca de agua. La imagen de la marca de agua se usó para verificarla cuando la imagen ya se haya reconstruida.

## ¿Qué dificultades se encuentran al elegir pares (k, n) distintos de los establecidos en este TP?

La dificultad radica principalmente en cómo guardamos los bits byte a byte en las imágenes portadoras que se pasan por parámetro. Por ejemplo, en el informe se detalla que en el caso de que estemos hablando del esquema (k=4, n=8) debiéramos usar el esquema de LSB1, es decir, solo se modifica 1 bit por pixel. En el otro caso del esquema propuesto, debemos modificar 2 bits.
En caso de querer utilizar otro esquema, con otro set de imágenes, nuestro algoritmo debe contemplar que en la imagen portadora entre la cantidad de información que debemos guardar. En caso de que el n y k sean altos, la matriz Sh consecuentemente será mayor y vamos a requerir de más píxeles para almacenar el secreto compartido. Si tuviésemos ese caso pero con imágenes muy pequeñas, es decir de pocos pixeles, la imagen original se distorsiona al punto de visualizarse. En caso de que ésto suceda, el concepto y/o idea original del algoritmo se desvanecerá.

## ¿Por qué es importante controlar el rango de A y el resultado de At.A?

El rango de la matriz A debe ser k, pero es importante que cumpla el requisito que

$$n > 2 * (k - 1) - 1$$

por el hecho que k representa la cantidad mínima de sombras necesarias para recuperar el secreto y n la cantidad total en la que el sistema distribuirá el secreto. En otras palabras, el valor de k nos dice el mínimo de participantes para recuperar el secreto. A menor número de k, el secreto no se puede recuperar. Es importante controlar el rango de A, ya que las operaciones matriciales que hay luego basadas en ésta fallan, la Sd tambien falla y así sucesivamente. Si no se controla, puede quedar no invertible alguna matriz y con sólo el rango de k, no queda asegurado.

## ¿Por qué es válida la forma de generar los Xi?

Es correcta porque estamos generando n vectores de k elementos considerando que deben ser vectores linealmente independientes entre ellos. De esta, forma garantizamos la existencia de n vectores, una por cada imágen a extender el secreto.
El esquema elige aleatoriamente dos vectores linealmente independientes tales que su producto es el secreto para ser compartido entre los participantes. Si el secreto es s luego, los vectores linealmente independientes $v_i$ se generan aleatoriamente de modo que $V = [v_1\ v_2\ v_3\ ...\ v_n]$. Esto agrega seguridad al esquema, ya que el secreto no se comparte directamente, sino que se comparte en forma colectiva.

Suponiendo el caso de que sea linealmente dependiente, cuando hagamos operaciones matriciales en la distribución y/o recupero, puede ocurrir de que se cancele una fila y se opere con matrices de dimensiones menores, y por consiguiente, la distribución o recuperación falle.

## La imagen RW que se obtiene es una imagen “con ruido”. ¿Sería necesario ocultarla mediante esteganografía? ¿Cómo podría hacerse?

En sí, no sería del todo necesario ya que con la imagen Rw no tenemos la información secreta.

Podría ser una opción ocultarla mediante otros métodos esteganográficos (no precisamente como el que detallamos en este TP ya que ello implica generar otra

imagen Rw para ocultar distribuidamente el antiguo Rw y sería lo mismo). Ésto conlleva a que usar otros métodos de ocultación de imágenes no garantizan confiabilidad en que la imagen a restaurar sea 100% exacta ya que, precisamente, el método que estamos implementando lo asegura.

Otra opción es la de encriptar la imagen Rw con algún método visto en teoría (usando clave simétrica ya que no sería del todo conveniente usar claves públicas/privadas ya que existen **n** participantes y todos ellos deberían tener la misma clave privada). Al final de cuentas, encriptar una imagen con "ruido" me dará otra imagen similar, o sea, una imagen que no hace sentido.

### ¿Por qué siempre hay que indicar n, aún al recuperar?

Es indispensable ya que para la recuperación se hacen operaciones y productos con matrices de tamaño n filas x k columnas. En caso de que no contemos con éste valor, no va a ser posible continuar con el recupero.

### ¿En qué otro lugar puede guardarse el número de sombra?

Se puede guardar en reserved byte 2 del header de los archivos bmp.

## Discutir los siguientes aspectos relativos al algoritmo implementado:

- Facilidad de implementación
- Posibilidad de extender el algoritmo o modificarlo.

El algoritmo implementado se hizo procurando la modularización del código. Para esto se busco el mayor desacople posible entre las operaciones entre matrices y el proceso a llevar a cabo.

Los procesos quedaron limitados a los archivos distribution.c y recovery.c. Los mismos tienen las secuencias de pasos para realizar su respectiva tarea haciendo uso de funciones implementadas en otros archivos. Esto nos permite tener un único lugar donde enunciar los pasos de cada proceso y poder modificarlos con facilidad (sea para agregar, quitar o modificar algún paso).

Por otro lado, se destaca el archivo matrix.c. Este contiene todas las operaciones de matrices que usan ambos procesos. Las mismas están pensadas para ser lo mas atómicas posibles y poder ser reutilizadas por cualquier paso de cualquiera de los dos procesos. De esta manera si alguno necesita una operación con alguna modificación, podrá hacer uso de la función y luego llevar a cabo la modificación, o en su defecto generar su propio método en matrix.c sin afectar otros pasos.

Al desarrollar de esta manera la implementación se hizo más fácil ya que se limitó a los archivos de procesos la implementación de los conocimientos de

criptografía removiendo cualquier distracción por implementaciones de cosas no tan propias a la materia como operaciones de matrices o manejo de archivos.

A nivel de extensión, no sería difícil hacerlo ya que, como hicimos mención antes, el código está dividido por funcionalidad. El único caso que debería contemplarse que, por cuestiones de practicidad no se hizo, fue la de validar y chequear las operaciones matriciales tomando esquemas distintos a los que en la práctica se menciona; (2,4) ó (4,8). Por otro lado, para la distribución de los bits en los píxeles de las imágenes portadoras, deberá contemplarse la cuestión de LSB, es decir, cuántos bytes se deben modificar. En nuestro caso, como el enunciado toma dos esquemas, ya nos referencia si debemos usar LSB1 o LSB2.

Por otro lado, hemos el software que hemos desarrollado, basados en los papers, presenta un inconveniente al momento de recuperar algunas imágenes; particularmente las distribuidas por nosotros mismos. Por momentos, se dibujan “lineas” aleatorias en las imagenes de recuperación. Encontrar el error era clave ya que, la imagen no se visualizaba correctamente en su totalidad. La solución fue cambiar el tipo de retorno de la función determinante de una matriz, pasando de int a int64_t.

Otro inconveniente que hemos encontrado, pero que no pudimos encontrar la solución efectiva, es la de que los blancos extremos (los que están en el rango de 251-255), al hacer las operaciones entre matrices para el recupero, tenemos un “overflow” que provoca que el byte se vaya de rango y se convierta en negro.

## ¿En qué situaciones aplicarían este tipo de algoritmos?

El problema con el esquema de secreto compartido que usa la proyección de matriz publicada por Li Bai es que no brindan una manera para que los participantes validen su imagen recibida, donde existe la posibilidad de una perturbación en la transmisión o un problema que surja al tratar con comerciantes o accionistas deshonestos.
Una extensión al esquema de intercambio secreto desarrollado por Wang et al. en la que se compartirá una imagen de marca de agua n × n con imágenes compartidas y, al determinar la precisión de la imagen de marca de agua, se podrá verificar la imagen secreta reconstruida.
El trabajo realizado modifica la imagen secreta utilizando la matriz proyectada por Li Bai utilizando la imagen de marca de agua en el esquema de Wang et al. para resolver el problema mencionado anteriormente.

## Ejemplos de prueba (cátedra)

El día Domingo recibimos casos para probar, tomando los esquemas (k=2,n=4) y (k=4,n=8). En el caso del esquema (4,2),  mostramos el siguiente ejemplo de recuperación

**Watermark recuperado**

**Secreto recuperado**