Criptografía y Seguridad

Trabajo Práctico Especial

Secreto Compartido en Imágenes con Esteganografía

Grupo 9

24 de Junio de 2019

Grupo 09

Arco, Martina - 57091
Godfrid, Juan - 56609
Radnic, Pablo Ignacio - 57013

Introducción

Resumen

Este documento consta de tres partes, las resoluciones de los ejemplos provistos por la cátedra, la sección “Cuestiones a analizar” donde se detallan las respuestas a las preguntas provistas y la sección “Dificultades Encontradas” donde se detallan otros problemas relacionados con la resolución de la problemática provista.

Metodología de resolución

La resolución del problema fue realizada en C. el equipo decidió hacerlo mediante la modalidad TDD (Test Driven Development), lo cual facilitó la a los miembros del equipo a verificar errores y asegurar que los cambios sobre el trabajo traigan calidad y no afecten negativamente a componentes no relacionados.

El trabajo se dividió en 5 módulos:

- **Matrices.** Operaciones matriciales y matemáticas.
- **Azzahra**. Resolución del algoritmo propuesto en el paper de referencia.
- **Image Manipulation**. Creación, lectura y escritura de imágenes.
- **Steganography**. Escritura y lectura de información oculta.
- **CryptoService**. Servicio de Encriptación y Desencriptación utilizando el algoritmo detallado.

Detalles de Implementación

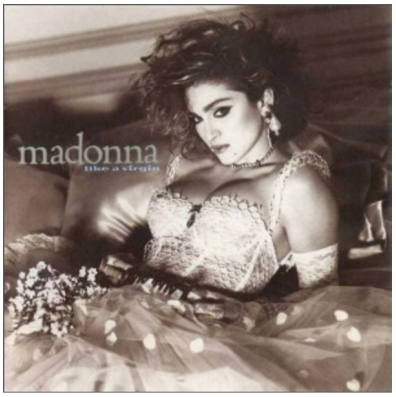
El proceso de compilación y ejecución del programa está detallado en el documento *readme.txt*, pero vale la pena destacar que más allá de haber implementado todas las opciones detalladas en la **sección 4.1: Generalidades,** se implementó una opción **-t** con la cual se ejecutan los *tests* unitarios mencionados anteriormente.

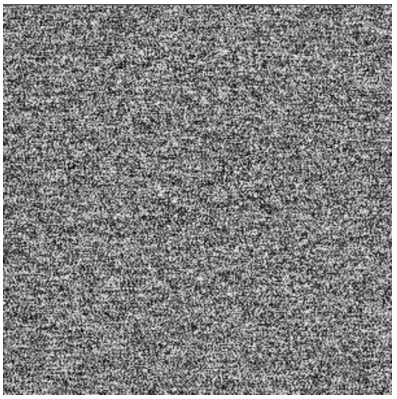
El proyecto contiene una discrepancia con el modo de ejecutar la opción que el enunciado especifica como “-dir”, ya que consideramos que para seguir el estándar POSIX las opciones *long* (de más de un carácter) deberían ejecutarse con doble raya, por lo tanto nosotros implementamos dicha opción con el argumento “--dir”.

Solución de Ejemplos de la Cátedra

Esquema 2-4:

Utilizando las siguientes sombras (grupo 9 esquema 2-4) y la RW provista por la cátedra:



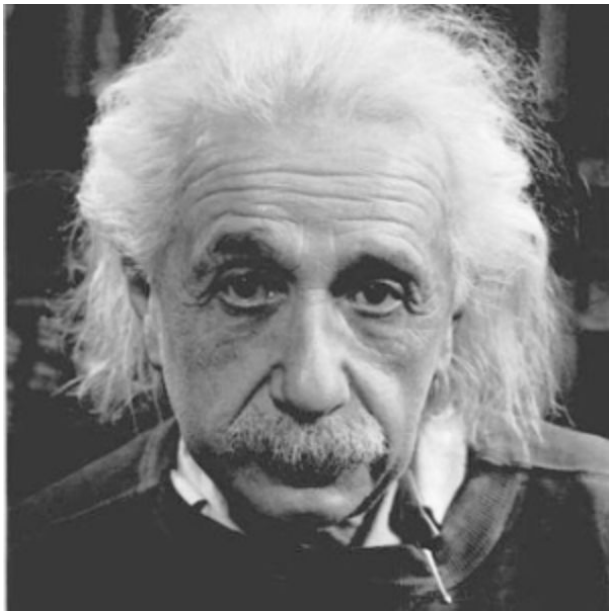


izquierda y centro sombras, derecha RW

Ejecutando el código:

./ss -r -s ../secreto_out.bmp -m ../2-4/rw/RW.bmp -k 2 -n 4 --dir ../2-4/

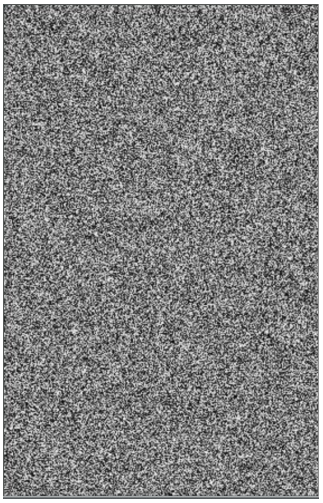
Se obtienen las siguientes imágenes:




izquiera '*secreto_out.bmp*' derecha '*watermark_out.bmp*'

Esquema 4-8

Utilizando las siguientes imágenes (grupo 9 esquema 4-8) provistas por la cátedra:



arriba sombras abajo RW.bmp

Ejecutando el código:

./ss -r -s ../secreto_out.bmp -m ../4-8/rw/RW.bmp -k 2 -n 4 --dir ../4-8/

Se obtienen las siguientes imágenes:

izquiera '*secreto_out.bmp*' derecha '*watermark_out.bmp*'

Cuestiones a Analizar

1. ***Discutir los siguientes aspectos relativos al documento.***
    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? ¿hay algún error?*

    El algoritmo de distribución utiliza la técnica descrita por Azzahra y Sugeng en el paper de referencia.Se desarrolló el algoritmo para dos tipos de esquemas k-n: 2-4 y 4-8.
    Partiendo de la imagen original se construyen muchas matrices nxn. Para cada una de ellas se efectúa el algoritmo de encriptación.
    Una vez que se encuentran construidas las matrices Sh no se las distribuye directamente sino que se utiliza un mecanismo de esteganografía para ocultar las mismas en otras imágenes provistas.

    Para efectuar la recuperación se recibe un subconjunto de tamaño k de las imágenes distribuídas y se extraen las matrices ocultas de ellas, con el método de esteganografía correspondiente.
    Una vez terminado esto se reconstruye la imagen original utilizando el algoritmo de desencriptación de *Azzahra*.

    El algoritmo de **encriptación de *Azzahra*** se puede definir en los siguiente pasos:

    i Construir Matriz A aleatoria de nxk
    ii Calcular Matriz SS la Proyección de A
    iii Construir n vectores linealmente independientes Xi.
    iv Elegir n escalares distintos Ci.
    v Construir n vectores Vi = A x Xi
    vi Utilizando operaciones sobre Vi y S construir n matrices Gi
    vii Construir matrices Shi concatenando Vi y Gi.

    El algoritmo de **desencriptación de Azzahra** puede definirse en los siguientes pasos:

    i Recuperar Matriz R resolviendo ecuaciones lineales sobre Shi
    ii Construir B concatenando Valores de Shi
    iii Recuperar SS igual a la proyección de B.
    iv Recuperar S, utilizando SS y R
    v Recuperar W, utiliizando Rw y SS

2. ***¿Por qué la propuesta de Azzahra y Sugeng supone una mejora a la propuesta de Li Bai?***
   El problema que tenía el algoritmo de Li Bai es, por un lado, que R representa un SPOF. Además, no provee una forma para que los participantes puedan validar las imágenes share. Consecuentemente, puede haber un problema en la transmisión o con la gente que las distribuye.
   El propuesto por Azzahara y Sugeng propone utilizar una imagen como marca de agua y, determinando su exactitud, entonces la imagen secreta obtenida puede ser verificada.
3. ***¿Qué dificultades se encuentran al elegir pares (k, n) distintos de los establecidos en este TP?***
   Las dificultades que se pueden encontrar son que, al hacer la esteganografía, no se puedan guardar los bits necesarios en las imágenes. Además, el tamaño de la imagen debe ser divisible por nxn para poder tener matrices cuadradas S.
   También, con estos valores se cumple la condición que $n>2(k-1)-1$ necesaria para crear las x.
4. ***¿Por qué es importante controlar el rango de A y el resultado de At.A?***
   El rango de la matriz determina cuántas filas o columnas linealmente independientes tiene la matriz. Si estas restricciones no se cumplen, luego al desencriptar no se puede calcular la inversa en la proyección.
5. ***¿Por qué es válida la forma de generar los Xi?***
   Para ser una forma válida se debe generar un conjunto de vectores linealmente independientes.
   El enunciado define los Xi como secuencias exponenciales de base i. Por ser secuencias exponenciales de distinta base son linealmente independientes.

6. ***La imagen RW que se obtiene es una imagen "con ruido".¿Sería necesario ocultarla mediante esteganografía?¿Cómo podría hacerse?***
   No es necesario ya que R es de dominio público. Si se quiere hacer, se puede utilizar una imágen más grande (24 bpp) y realizar lsb, tal como se hace en las shares.
7. ***¿Por qué siempre hay que indicar n, aún al recuperar?***
   Porque se necesita saber el tamaño de las matrices a recuperar para hacer esteganografía. Además, se puede utilizar para saber que el participante sabe cómo utilizar el programa. Es decir, que es el indicado para tener las shares.
8. ***¿En qué otro lugar puede guardarse el número de sombra?***

Offset hex	Offset dec	Size	Purpose
00	0	2 bytes	The header field used to identify the BMP and DIB file is `0x42 0x4D` in hexadecimal, same as `BM` in ASCII. The following entries are possible: **BM** Windows 3.1x, 95, NT, ... etc. **BA** OS/2 struct bitmap array **CI** OS/2 struct color icon **CP** OS/2 const color pointer **IC** OS/2 struct icon **PT** OS/2 pointer
02	2	4 bytes	The size of the BMP file in bytes
06	6	2 bytes	Reserved; actual value depends on the application that creates the image
08	8	2 bytes	Reserved; actual value depends on the application that creates the image
0A	10	4 bytes	The offset, i.e. starting address, of the byte where the bitmap image data (pixel array) can be found.

   El número de sombra, además de en el byte 6 se podría guardar en los bytes 7 8 y 9.
   Una alternativa que pensó el equipo es la de desarrollar un protocolo de datos propio con su propio encabezado que se almacenaría directamente en el cuerpo de la imágen. Este encabezado podría guardar información como el número de sombra y la cantidad de bytes almacenados permitiendo que el ocultamiento se pueda llevar a cabo en cualquier imágen con el tamaño suficiente para almacenar los datos y quitaría la restricción de que las sombras y la imagen original debieran tener el mismo tamaño.

9. ***Discutir los siguientes aspectos relativos al algoritmo implementado:***
    a. *Facilidad de implementación*
    b. *Posibilidad de extender el algoritmo o modificarlo.*

    La facilidad de implementación es objetable. Si bien el algoritmo es claro y basándose en el paper provisto de *azzahra* se puede realizar una primera implementación de manera relativamente rápida, garantizar que el algoritmo sea robusto es notablemente más difícil. Implementar algoritmos que manejan números de órdenes grandes, tomando en consideración el orden de precisión de los tipos enteros en C, resultó particularmente difícil.

    En cuanto a la extensión, el algoritmo carece de firmas digitales o MACs, añadiendo alguna de estas dos funcionalidades a la hora de la encriptación y revisando la misma a la hora de la desencriptación se conseguirá un algoritmo más criptográficamente seguro.

10. ***¿En qué situaciones aplicarían este tipo de algoritmos?***

    Este algoritmo es efectivo para sistemas que tienen información distribuida, con conectividad permanente y con altos requerimientos de seguridad.
    La caída de un nodo no alcanza para evitar que los otros nodos de la red continúen consumiendo la información, y se garantiza que ningún adversario pueda interceptar la información encriptada leyendo la información guardada en el nodo caído.
    Algunos ejemplos de escenarios reales que pueden ser empoderados por la utilización de este tipo de algoritmos son:

    Imágenes satelitales.
    imágenes médicas.

Dificultades encontradas

Problema de los Píxeles altos

Durante la ejecución del programa se observó que las imágenes recuperadas difieren de las originales en algunas áreas. Estas zonas en la imagen original eran particularmente blancas, y en la imagen recuperada se las observaba de colores oscuros, el equipo se refirió a esto como el ‘quemado’ de la imagen (observar ‘*secreto_out.bmp’* en la página anterior) .

Luego de un breve análisis se descubrió el causante de estas zonas ‘quemadas’. El algoritmo necesita la elección de un valor *p* que debe ser primo. El valor de *p* elegido fue 251, por ser este el elegido en el paper de azzahra de referencia, esto trajo el problema que para zonas de la imagen donde los valores de los pixeles eran mayores al *p* elegido el algoritmo ‘reducía’ los números al aplicarle el módulo *p*.

Para solucionar este problema el equipo decidió que se normalizarían los valores de los píxeles antes de hacer la encriptación. Aquellos píxeles cuyo valor fuera superior o igual a *p* se reducirían a *p-1* para evitar cambios dramáticos en la recuperación.

Si bien esta solución hace que la imagen pierda detalle consideramos que sus efectos no son inmediatamente visibles y no disminuyen la efectividad del algoritmo.

El grupo también consideró la posibilidad de utilizar *p* con valor de 257 (otro número primo) lo cual evitaría pérdidas de información, sin embargo por carecer de los ejemplos de referencia y las disminuidas mejoras que otorgaría el cambio se optó por permanecer con la solución anterior.

¿Cómo aumentar el ‘Ruido’?

Una de las áreas en las que el grupo se encontró con las mayores dificultades fue en la generación de la matriz A. Esta matriz fue identificada por el grupo como la principal fuente de ‘Ruido’ del algoritmo, cuanto más aleatoria era A, más ruidosas se veían las imágenes encriptadas.

Sin embargo ocurría un problema al hacer la matriz A completamente aleatoria, el resultado de ello es que la matriz SS no es recuperable debido a que en el cálculo de la proyección se encuentra una matriz no inversible.

Debido a esto el equipo debió hacer un trabajo exhaustivo para conseguir matrices A que garanticen su correcta desencripción pero que mantengan niveles altos de ruido.

Se optó por utilizar el algoritmo propuesto en el enunciado para la generación de X’s linealmente independientes para la generación de las columnas de A.

Con este método se garantiza la correcta recuperación de la imágen.

Problema de los determinantes grandes

Otro de los mayores problemas con los que se topó el equipo, particularmente al utilizar el esquema 4-8, fue el de los determinantes grandes.

Para determinados cálculos los valores de las matrices excedían la precisión dada por los tipos enteros (‘*int*’) en *C*, este efecto era particularmente notable en el cálculo de determinantes de matrices 8x8.

Una vez identificado el problema el equipo determinó dos alternativas, la primera, cambiar los tipos de las variables por enteros de mayor grado de precisión (*‘int_64t’*). O aplicar cuidadosamente la función módulo en los pasos intermedios de las funciones de alta complejidad.

Se optó por utilizar la segunda la cual ofrecía un menor grado de efecto en componentes no relacionados.

Referencias

1. Enunciado.
   '*TRABAJO PRÁCTICO DE IMPLEMENTACIÓN:SECRETO COMPARTIDO EN IMÁGENES CON ESTEGANOGRAFÍA*'
2. Paper de Li Bai
   'An Image Secret Sharing Method'
3. Paper de Azzahra y Sugeng
   '*Verifiable Image Secret Sharing Using Matrix Projection*'