Criptografía y seguridad

Trabajo Práctico Especial de implementación

Secreto Compartido en Imágenes con Esteganografía

Grupo 4:

- María de los Angeles Arlanti marlanti@itba.edu.ar 53373
- Mauricio Minestrelli mminestr@itba.edu.ar 52015
- Santiago Ocamica socamica@itba.edu.ar 53346
- Agop Matías Hurmuz ahurmuz@itba.edu.ar 53248

Profesores:

- Abad, Pablo
- Ramele, Rodrigo
- Arias Roig, Ana
- Cristian Ontivero

Índice

Introducción

En el siguiente informe se presenta un análisis sobre la implementación del algoritmo de Secreto Compartido en Imágenes descripto en los papers “An Efficient Secret Image Sharing Scheme” (Luang-Shyr Wu y Tsung-Ming) y “Secret image Sharing” (Thien y Lin), basados en el método de Secreto Compartido desarrollado por Adi Shamir y George Blakley.
El programa fue realizado en Java y cuenta con las siguientes funcionalidades:

1. Distribuye una imagen secreta de extensión “.bmp” en otras imágenes también de extensión “.bmp” que constituyen las sombras en un esquema (k, n) de secreto compartido.
2. Recupera una imagen secreta de extensión “.bmp” a partir de k imágenes, también de extensión “.bmp”

Recuperación del Secreto

Tomando como valor de `r` la totalidad de las sombras entregadas por la cátedra, se recuperó la imagen secreta escondida. La misma es la siguiente:

Figura 1: Imagen secreto recuperada con el programa.

Comentarios sobre documento Thien y Lin[1]

El documento describe de forma ordenada y concisa el método para compartir una imagen secreta repartida en n sombras y que cualquier cantidad `r <= n` de dichas sombras pueden ser utilizadas para recuperar la imagen secreto completa.
El documento se encuentra organizado de tal forma que se exponen los algoritmos de distribución y recuperación descritos en pasos y con aclaraciones a los contenidos más formales (matemáticamente hablando) lo que facilitó tu comprensión. Por otro lado se

[1] ftp://ftp.im.tku.edu.tw/Prof_Hou/%BCv%B9%B3%B3B%B2z%BBP%B8%EA%B0T%C1%F4%C2%C3/secret%20image%20sharing.pdf

expone una variante para no truncar los valores de pixels de la imagen secreta y así recuperar una imagen sin pérdidas.
Luego describe los resultados obtenidos y realiza un análisis acerca de los aspectos de seguridad del método. Se puede concluir que el documento ofrece:

- Un método tal que la imagen puede ser compartida por varias imágenes sombra.
- El tamaño de las imágenes es `1/r` de la imágen secreto. El tamaño pequeño de las imágenes sombra facilitan el almacenamiento, transmisión y ocultamiento.
- Dos versiones para suplir el problema que algunos píxeles de la imagen secreta puede llegar a tener valores mayores a 250.

Recuperación de Imagen

La imagen es exactamente igual a la oculta excepto por ciertos píxeles que corresponden a los casos en que una evaluación del polinomio daría congruente a 256 módulo 257. En esos casos difiere por lo general en 1 bit de 1 píxel, lo que es imperceptible a simple vista.
Usando algún software de comparación de imágenes se podrían encontrar las diferencias. Como se observa en la siguiente figura recuperada por nuestro algoritmo:

Figura 2: Los píxeles magenta indican diferencias con la imagen original. Las mismas son imperceptibles a simple vista.

Imágenes Portadoras

Siguiendo el paper de Thien y Lin, el tamaño de cada sombra debe ser más chico que el de la imagen secreto. Esta propiedad da el beneficio de una mejor capacidad de almacenamiento, transmisión, u ocultamiento de las sombras. El tamaño de las sombras debe ser entonces `1/r` según el método, siendo `r` la cantidad mínima de sombras necesarias para recuperar el secreto.
Por otro lado, el método de ocultamiento requiere 8 bytes por cada byte que se quiere ocultar, entonces, para un caso que `r` es distinto de 8, el tamaño de las imágenes portadoras debería ser `8/r` el tamaño de la imagen a ocultar. Por ejemplo con `r=2`, las imágenes deben ser 4 veces más grandes que la original respecto de la cantidad de pixels.

Algoritmo Implementado

Dificultades Encontradas

La utilización de los papers "An Efficient Secret Image Sharing Scheme" y "Secret image Sharing" facilitó mucho la compresión de los pasos a realizar y la implementación del algoritmo.
La mayor dificultad radicó en la aplicación de la técnica de esteganografía, sobretodo en la modificación de los bits.
Tuvimos problemas para ocultar correctamente la información en los archivos .bmp ya que al principio tomábamos mal el offset. Luego como en Java se dificulta el acceso a los bits, hay que acudir a realizar shifts o utilización de funciones lógicas. Todos estos inconvenientes nos llevaron a pensar a bajo nivel y a tener un mayor entendimiento del manejo del formato de los archivos y su aprovechamiento.
Complementamos la información obtenida de los papers y el enunciado con un artículo sobre la estructura del header de los archivos BMP[2].

Extensiones Posibles

El algoritmo está preparado para el manejo de imágenes en escala de grises y formato bmp. El formato 8 bit por pixel utilizado soporta 256 colores y almacena 1 pixel por byte, cada byte es un index a una tabla que representa dicha cantidad de colores. Es una posibilidad extender la investigación para realizar un aprovechamiento de dichas tablas y poder soportar imágenes con las mismas características que las que actualmente recibimos, pero en color.

[2] https://en.wikipedia.org/wiki/BMP_file_format#Bitmap_file_header

Por otro lado imágenes con mayor definición (como por ejemplo 24 bits por pixel) soportan 16,777,216 colores distintos y albergan un pixel cada 3 bytes. Ya que soportar imágenes con mayor definición supone mayor tamaño de los archivos, hay que tener en consideración modificaciones en la performance del algoritmo y manejo de memoria. Habría que tener en cuenta cargas parciales de las imágenes en memoria para realizar las distintas modificaciones.

Por último, sería trivial extender nuestra implementación a cualquier formato “lossless”, ya que los mismos conservan intacta la información de los píxeles. Por el contrario, soportar otro tipo de formatos de imágenes tales como jpg, gif, png, o cualquier formato que tenga compresión con pérdida de información representaría un desafío mayor.

Posibles Aplicaciones

Una implementación de un algoritmo de secreto compartido aplicado a imágenes encuentra aplicación en cualquier escenario donde las imágenes a compartir digitalmente son de extrema confidencialidad. En el área comercial, es posible querer proteger capturas de archivos, contratos. En el área de la medicina es posible querer ocultar imágenes de estudios de ciertos pacientes, donde el diagnóstico es reservado. En la milicia, imágenes de mapas, estrategias, etc.
Otro tipo de técnica construida para incrementar la seguridad del secreto es el Digital Watermarking. Dicha técnica es un tipo de marca embebida en una señal tolerante a los ruidos tal como una imagen. Es típicamente utilizada para identificar la autoridad o el copyright de dicha señal.

Conclusiones

Se ha logrado implementar el algoritmo de Adi Shamir y George Blackley a través del concepto de Imagen Secreta Compartida de Chih-Ching Thien, JaChen Lin, Kuang-Shyr Wu y Tsung-Ming Lo.
Siguiendo el método de Kuang-Shyr Wu y Tsung-Ming logramos que una imagen secreta en escala de grises de 8 bits pueda ser reconstruida desde cualquier cantidad r de sombras casi sin ninguna pérdida. Se redujo el tamaño de las sombras de las imágenes para un mejor almacenamiento y transporte. En caso de que $(r-1)$ o menos sombras sean robadas, el método protege el secreto, haciendo la probabilidad de acierto muy pequeña.