Criptografía y Seguridad

Secreto Compartido

Esteganografía

Federico A. Elli – 50817

Juan Martín Buireo – 51061

Martín Purita – 51187

Introducción

El problema consiste en implementar un programa en lenguaje C que implemente el algoritmo de secreto compartido de Blakley 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:

- 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.

- Recuperar una imagen secreta de extensión ".bmp" a partir de k imágenes, también de extensión ".bmp"

Notación

A lo largo del documento se hará referencia a los siguientes parámetros, valores y abreviaturas.

- n: Cantidad de imágenes en las que la imagen original será distribuida.

- k: Mínima cantidad de imágenes necesarias para poder reconstruir la imagen original.

- BMP: BitMap que incluye un header de 54 Bytes.

- bpp: Bits por pixel

- imagen portadora: Imagen que lleva parte de la imagen a reconstruir.

Implementación

La implementación fue realizada en el lenguaje C. Los valores de k varían entre 2 y 3, y los valores de n entre 3 y 8 .

Descripción del algoritmo

El algoritmo consiste en tomar una imagen en escala de grises, y utilizando el método del paper, distribuirla en n imágenes también en escala de grises, distintas de la original. Las imágenes que contienen el secreto se llamarán imágenes sombra.

Luego utilizando cualesquiera k (< n) de las imágenes sombra se podrá recuperar la imagen original.

Análisis General

1. Discutir los siguientes aspectos relativos al documento de Ulutas y sus colegas:

a. Organización formal del documento.

La organización del documento nos parece adecuada ya que es clara y ordenada. Un detalle que observamos es que no se explica el significado de ciertas siglas, por ejemplo, qué significa PSNR (Peak Signal-to-Noise Ratio) utilizado para medir cuánto fueron modificadas las imágenes respecto de su estado original luego ser aplicado el algoritmo y de distribuirse la imagen secreta.

Como todo documento, comienza con una pequeña introducción de los distintos esquemas de secreto compartido y a su vez, brinda ejemplos de concretos de uso.

En la segunda sección se hace referencia al método de Blakley dado que el algoritmo se basa en este método.

La tercer sección explica detalladamente el algoritmo, separando los casos de distribución y recuperación de las imágenes.

La cuarta sección presenta resultados experimentales de utilizar distintos esquemas y se expone una comparación entre ellos.

La última sección (la de conclusiones), explica las ventajas de utilizar este método y las conclusiones del apartado anterior.

b. La descripción del paso 7 del algoritmo de reconstrucción.

El paso 7 del algoritmo de reconstrucción propone resolver el sistema lineal de congruencias calculando la matriz inversa del sistema y resolviendo $X = A^{-1} * B$. En nuestra implementación utilizamos la regla de cramer para resolver el sistema lineal (aplicando las congruencias correspondientes).

c. La notación utilizada, ¿es clara? ¿cambia a lo largo del documento?

La notación es clara, sin embargo escribe por ejemplo sobre 'stego images' sin explicar a que hace referencia (las imágenes sombra que contienen la imagen original escondida)

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?

En el esquema original si se deja una sombra con ceros, ya se está revelando parte del secreto ya que serían necesarias menos imágenes sombra para reconstruir la original. Por otro lado, la inclusión de los mismos puede dar lugar a que el sistema sea incompatible, dando lugar a infinitas soluciones.

En el método propuesto por Ulutas y sus colegas, las n ecuaciones pueden ser levemente modificadas para que dado cualquier subconjuntos de k ecuaciones, estas resulten linealmente independientes. Es decir que el sistema quedará compatible determinado para cualquier conjunto de k ecuaciones. Se habla de una leve modificación, ya que idealmente la modificación deberá modificar lo menos posible las sombras.

Es por lo expuesto en el párrafo anterior que Ulutas y sus colegas no se ven obligados a descartar las sombras que tengan ceros.

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).

La imagen recuperada es prácticamente igual a la original, excepto en algunos pixels blancos que son convertidos a negros. Esto se puede deber a un manejo incorrecto en los módulos durante la distribución.

4. Discutir los siguientes aspectos relativos al algoritmo implementado:

a. Facilidad de implementación

Las dificultades en la implementación son en parte relacionadas al lenguaje C. El resto de las dificultades son al momento de verificar la independencia lineal de las ecuaciones generadas. Hay lenguajes que permiten realizar esto de manera más simple (Octave por ejemplo).

b. Posibilidad de extender el algoritmo para que se usen imágenes en color.

El algoritmo se puede extender, realizando las validaciones de independencia lineal en cada canal de color rojo, verde y azul.

Ver: C.-C. Chang, C.-C. Lin, C.-H. Lin, and Y.-H. Chen 'A novel secret image sharing scheme in color images using small shadow images'

c. Ventajas respecto del algoritmo original de Shamir (mencionar por lo menos 2)

Al usar el algoritmo original, se generan imágenes que parecen ser ruido. Esto llama la atención y aumenta la probabilidad de que un atacante intente descifrar el secreto. Con el algoritmo de Ulutas se utiliza una imagen real sobre la cual se esconde el secreto, por lo tanto es más difícil detectar que existe una imagen secreta escondida.

El algoritmo de Shamir requiere imágenes de 2Nx2N para ocultar una imagen de NxN. Utilizando este nuevo método el tamaño puede ser reducido a NxN, mejorando los requerimientos de espacio y ancho de banda.

5. ¿Qué dificultades tuvieron en la lectura del documento y /o en la implementación?

Nos costó entender en un principio el funcionamiento del bit p. No comprendiamos su uso con lo cual fue necesario consultar a la cátedra entendiendo que dicho valor servía para conservar la integridad de una imagen.

En la implementación nos encontramos con problemas de programar en C (estando acostumbrados a programar en Java). Uno de los retos mayores fue el de lograr la independencia lineal y el de resolver un sistema de ecuaciones. La implementación de la regla de Cramer nos facilitó en gran medida estos problemas.

Intentamos hacer los algoritmos de forma general, pero encontramos que sirve para k = 2 y 3 pero ya no para k mayores.

6. ¿Qué extensiones o modificaciones harían a la implementación o al algoritmo?

Con la dificultad que esto agrega, se puede extender el algoritmo para usar Ks genéricos, el problema existe en la dificultad de verificar la independencia lineal de todas las ecuaciones. También hay dificultad en determinar cómo tomar los Ai's correspondientes, ya que a diferencia de k = 2 y k = 3, hay más de una manera.

Podrían combinarse estos métodos con métodos de compresión, procurando no destruir el secreto (preservando la independencia lineal)

Con el tiempo necesario, sería interesante realizar esto en Java utilizando alguna librería de manejo de imágenes (image4j, BMP I/O).

7. ¿En qué situaciones aplicarían este tipo de algoritmos?

Este tipo de algoritmos que mezclan esteganografía con secreto compartido son utilizados actualmente en algunas situaciones.

Las impresoras láser color de HP y Xerox agregan puntos de color a cada hoja. Al reconstruir el secreto se puede obtener el número de serie de la impresora, dirección IP, fechas y timestamps. Las mismas pueden ser utilizadas por fuerzas de seguridad para encontrar y detener delincuentes.

Grupos terroristas utilizan sitios masivos como Ebay, donde realizan publicaciones de algún producto y en su avatar esconden mensajes secretos con objetivos o planes maliciosos. La imagen se reconstruye tomando los avatares de k publicaciones.

En general siempre que se quiera esconder un secreto dentro de una imagen, y que sea difícil detectar (por el ojo humano), que la misma esconde parte de un secreto.

Resultados obtenidos

Se realizaron pruebas para k = 2 y k = 3 con el mismo n (3). Se distribuyeron imágenes utilizando las mismas de 300x300 y 600x600.

En el caso de 600x600, con k = 2, se generaron las siguientes imágenes shadow:

Shadow image 1 – 600x600 – k = 2

Shadow image 2 – 600x600 – k = 2

Shadow image 3 – 600x600 – k = 2

Con k = 3, se generaron las siguientes imágenes shadow:

Shadow image 1 – 600x600 – k = 3

Shadow image 2 – 600x600 – k = 3

Shadow image 3 – 600x600 – k = 3

El secreto distribuido en las imágenes anteriores fue el siguiente:

Imagen original

K = 2

K = 3

Comparando las imágenes originales con las mismas luego de ser distribuidas y recuperadas, se puede notar que son casi idénticas. Por otro lado, se puede notar claramente que en las imágenes shadow que fueron distribuidas con k = 3 el cambio se nota menos que en aquellas distribuidas con k = 2. Esto es lógico ya que en aquellos casos de k = 2 se están modificando más bits en cada posición para poder distribuir.

Conclusiones

Pudimos verificar que el método funciona, tanto en la recuperación como en la distribución.

En la recuperación hay algunos problemas con los píxeles blancos (los cuales se convierten en negros). Esto se debe a lo realizado previamente en la distribución.

Se presenta un error en la validación del bit p en la recuperación en aquellas imágenes que no fueron distribuidas por nuestro algoritmo. Igualmente, no cortamos la ejecución sino que solamente mostramos un warning aclarando que la imagen pudo haber sido modificada.

Anexo

Imágenes de la cátedra recuperadas: