Instituto Tecnologico de Buenos Aires

Secreto Compartido en Imágenes con Esteganografía

Criptografia y Seguridad

Martin Victory	**56086**
Segundo Fariña	**56176**
Florencia Cavallin	**56015**

1. Discutir los siguientes aspectos relativos al documento.

a. Organización formal del documento.

El sistema generado de encriptación y ocultamiento de imágenes, se lo puede considerar como un conjunto de distintas partes. Inicialmente se debe poder leer y escribir imágenes en formato bmp. Luego se debe poder realizar estenografía (la cual se entra en mas en detalle mas adelante) sobre esas imágenes, para poder ocultar la imagen efectivamente. Y por último, y más importante se encuentra el algoritmo de encripción y desencripción.

Toda esta información se encuentra comentada en el documento provisto por la cátedra. Pero para poder realizar correctamente la ultima parte, se debe profundizar totalmente los temas, analizando y comprendiendo lo mencionado en los distintos papers.

b. La descripción del algoritmo de distribución y la del algoritmo de recuperación.

La descripción de ambos algoritmos, descripta en el paper de Azzahra y Sugeng no es perfecta. Principalmente esto se debe a que no se explican en detalle los pasos a realizar, sino que se mencionan en forma de lista, asumiendo que el lector puede comprenderlos. Por suerte, se incluye un ejemplo del funcionamiento con todas las cuentas realizadas, lo cual ayuda fuertemente a la comprensión del funcionamiento del algoritmo. Aun así, es recomendable leer el paper original de Li Bai, ya que en este se encuentra en más detalle, como es que ciertas partes del algoritmo funcionan.

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

A pesar de no estar explicada, en general, la notación se entiende. Aun así, en ciertos puntos seria conveniente mayor información. Por ejemplo, en el algoritmo de encriptación no se especifica que representa el valor j, ni tampoco que Gj es una matriz. En ese mismo punto, se pueden generar confusiones, ya que se utilizan indices comenzando de 1 en lugar de comenzar en 0. De igual modo, en la desencripción, no queda para nada claro cómo es que se debe generar la matriz R. Por suerte, el ejemplo da pasos mas detallados de su resolución, si bien no son perfectos, ya que no son explicados.

Puntualmente, nosotros tuvimos dificultades en este punto, ya que al tratar de implementar el esquema $(4, 8)$, no queda claro cómo se debía resolver el sistema de ecuaciones. Tras varias pruebas en papel, logramos comprender el funcionamiento del algoritmo en su totalidad.

Con respecto a la variación de notación, existe un cambio en el nombre de las variables al momento de recuperar la matriz R. Se remplaza i, j por las variables x, y. También existe un pequeño error en la formula presentada en el algoritmo de Schamir, pero se logra comprender el sentido de todas maneras.

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

Si bien ambas propuestas están basadas en el mismo concepto, la propuesta de Li Bai presenta un potencial problema. Es un algoritmo muy seguro, pero posee un único punto de falla en la matriz *R*. Si esta matriz se corrompe, o sé altera maliciosamente, no existe manera alguna de corroborar que la información luego recuperada sea exactamente la misma que la encriptada originalmente. Es por esto que Azzahra y Sugeng mejoran el algoritmo agregando el concepto de marca de agua. Al momento de la encripción no solo se encripta un mensaje, sino también la marca de agua, que sirve únicamente para poder verificar en la desencripción, que el secreto es el correcto.

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

El algoritmo propuesto por Azzahra y Sugeng no presenta restricciones a los valores de *(k, n)* que se puedan usar. Pero la limitación a los pares *(2, 4)* y *(4, 8)* se da debido a la accion que se realiza con la salida del algoritmo. Una vez procesada la encriptacion de los valores, estos se ocultan en imágenes denominadas *shares*. Para poder realizar esto, se almacena cada uno de los valores obtenidos en los bits menos significativos de los pixeles de las *shares*. En el esquema *(2,4)* los dos últimos bits de cada pixel de las imágenes *shares* son remplazados. Mientras que en el esquema *(4, 8)* el ultimo bit de cada pixel es remplazado. Al variar los valors de *k, n* se generarían complicaciones con la esteganografía, la cantidad de pixels no coincidiría con los necesarios, o también se podría dar que se deban almacenar mayor cantidad de bits por pixel, provocando que se observen cambios notorios en las imágenes *shares*.

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

Una de las partes centrales del algoritmo, se basa en calcular la proyección de la matriz *A*. La proyección de una matriz es un conjunto de operaciones incluyendo multiplicaciones, transposiciones, e inversas. Para poder completar correctamente el algoritmo, la proyección debe ser realizada satisfactoriamente. Es por esto que previo a comenzar el algoritmo, nos debemos asegurar que A posea esas propiedades, ya que son las que indican si la matriz es invertible, y por consecuencia, si se le puede realizar una proyección.

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

El método para generar los distintos *Xi* linealmente independientes se basa en tomar un valor aleatorio nunca antes elegido. Ese sera con el que se construya *Xi*, al aplicar operaciones de

potencia, para obtener cada uno de los elementos. Inicialmente el valor elegido *r* se lo eleva a la 0, lo que genera que el primer elemento del vector sea un *1*. El segundo elemento, se lo genera elevando *r* a la 1, obteniendo el mismo valor. En este punto tenemos un vector con los valores *[1, r]*. Esto ya es linealmente independiente siempre que *r* no se repita, ya que todos tiene un elemento 1 y se distinguen en el segundo, por lo que no existe ningún factor que al multiplicarlo por un vector *Xi* me genere otro vector que haya generado previamente, o pueda generar en el futuro. También es importante que *r* sea modulo *251*, ya que sino se repetirían valores.

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

En principio, no es necesario ocultar la matrix *RW*, ya que por sí sola no puede informar nada sobre la matriz secreta *S*. Aun así, si lo que se desea es ocultar una imagen para transferirla a otra persona de manera secreta, y se pasa *RW* como una imagen simple, se pueden generar dudas sobre lo que realmente se esta transfiriendo. Cualquier persona normal que mire las imágenes transferidas no notaria nada al ver los *shares* pero sí se sorprendería al ver *RW*. Para evitar esto, se puede decidir ocultar *RW* también con esteganorafia, como se realiza con los *shares*.

Ocultar *RW* no es igual que ocultar los *shares*, ya que la cantidad de valores difiere entre estas dos matrices. Para poder ocultar *RW* correctamente se debería tomar una imagen base más grande que las demás, o se debería distribuir su información en un conjunto de imágenes base que al juntarse contengan toda la información de *RW*.

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

Al recuperar no es estrictamente necesario saber el valor de *n* a nivel algoritmo. Pero dada la implementación provista, se utiliza el valor de *n* como tamaño de las matrices *S* y *R* a recuperar. A su vez, para la sección de esteganografia, es necesario saber el valor de *n*, ya que de esta manera podemos saber cuántos bits están ocultos en cada uno de los pixels de las imágenes *shares*.

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

Actualmente, el numero de sombra se guarda en uno de los bytes reservados que existen en el header de las imágenes bmp. Este es un lugar ideal, ya que esta totalmente oculto, y es una porción de la imagen que no tiene un propósito especifico. De no poder usar esa ubicación, se podría almacenar como parte del nombre, aunque no es recomendable, ya que esta totalmente sujeto a que un usuario lo modifique y se corrompa la información. O también se podría ocultar

en bytes extra de la imagen (que no sean el reservado) y que se ubiquen después del header, incrementando el valor de offset que indica el comienzo de los pixels de la imagen.

9. Discutir los siguientes aspectos relativos al algoritmo implementado:

a. Facilidad de implementación

Si bien la implementación no es excesivamente difícil, presenta cierta complejidad, ya que se debe implementar de cero todo el manejo de matrices. También se incrementa la dificultad, ya que el paper no esta completamente explicado, habiendo varios puntos en los que se debe interpretar el funcionamiento ocasionando fallas.

Para un correcto funcionamiento, es de vital importancia realizar un código bien modularizado y poder testear los distintos componentes por separado.

El orden de dificultad de las partes esta dado principalmente por el manejo de matrices y sus operaciones. Seguido por la implementación del algoritmo a nivel de matrices. Luego por el manejo de archivos bmp, y por ultimo la implementación de esteganografia.

b. Posibilidad de extender el algoritmo o modificarlo.

Existen distintas extensiones que se le podrían realizar al algoritmo, principalmente se podría ocultar los valores de *RW* en un conjunto de imágenes para evitar sospechas. También se podrían aplicar mejoras a nivel algoritmo, para optimizar la eficiencia del mismo.

Debido a que el algoritmo se debe realizar en modulo 251, las imágenes desencriptadas tienen pequeños puntos negros que se deben a ese rango de valores no comprendidos por el algoritmo. Una mejora posible seria encontrar un método para recuperar estos valores y evitar alterar la imagen original.

Otra extension posible seria el no estar limitado a encriptar imágenes únicamente en escalas de grises. Se podría manejar imágenes a color, aplicando él algoritmo a cada una de las bandas RGB por separado.

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

Este algoritmo es de gran ayuda cuando se quiere ocultar imágenes sensibles al publico general, pero se quiere poder acceder a ellas cuando sea requerido. De esta forma, se podría tener imágenes publicas distribuidas, y cuando es necesario se las junta y se obtiene la imagen secreta. Por ejemplo, el algoritmo se podría utilizar en una aplicación móvil para ocultar imágenes en la galería de fotos, y únicamente al ingresar una contraseña se unen las imágenes para obtener la imagen secreta.