Criptografía y Seguridad

Trabajo Práctico de Implementación

Secreto Compartido con Esteganografía

Brula, Matias Alejandro - **58639**
Tallar, Julián - **59356**
Vuoso, Julián Matías - **59347**

Grupo **02**

2021Q1

Índice

1. Introducción

Entre los principales objetivos de este trabajo están la introducción al campo de la criptografía visual y sus posibles aplicaciones en la vida cotidiana; introducción al campo de la esteganografía y nuevamente, sus aplicaciones; y por último el proceso creativo y lógico que conlleva implementar y analizar dicho algoritmo.

El concepto de criptografía visual es uno relativamente nuevo, introducido por por Adi Shamir y Moni Naor en la presentación Eurocrypt del 1994. Distribuir de manera segura imágenes con información secreta que, al momento de superponerse, recuperan la imagen secreta.

Las aplicaciones más comunes de estos algoritmos varían, pero las más comunes y frecuentemente utilizadas son[1]:

- Encriptar una imagen de contenido sensible y/o privado
- Esconder información dentro de un archivo PDF
- Delitos cibernéticos (escondiendo datos a plena vista)
- Prevenir el acceso no autorizado

La particularidad de este sistema de encriptación no es el fin en sí mismo, sino cómo lo logra. Como todo sistema de encriptación este busca ocultar, esconder, o bien no permitir el acceso a información o sistemas de terceros.

A diferencia del resto, la criptografía visual permite apreciar la potencialidad de la misma, escondiendo información no visible al ojo humano a simple vista. Como dijo mi hermano menor, “parece magia”.

[1] https://ieeexplore.ieee.org/document/7784984

2. Modelo Teórico

2.1. Secreto Compartido

En criptografía, un secreto compartido es un pedazo de información, sólo conocido por las partes involucradas en una comunicación segura. Dicha información puede ser una contraseña, una frase secreta, un número primo grande, un vector de bits random (OTP) o bien como para este caso, imágenes portadoras.

Habiendo varios tipos de secretos compartidos, el tratado en cuestión es de tipo secreto umbral-(t, n). Este es un esquema eficiente creado por Shamir y Blakely donde:

- Se codifica el secreto en n segmentos, compartidas entre n participantes
- Solo se puede decodificar si hay al menos k participantes aportando su segmento
- De no haber al menos k, no se puede revelar la información

2.2. Esteganografía

Por definición la esteganografía es la ciencia que busca la manera de ocultar un mensaje. Estos métodos de ocultamiento son utilizados para lograr que la imagen secreta en cuestión (que se busca esconder), se oculte en las sombras siendo prácticamente imperceptible. Aquel que contiene el mensaje secreto (ya sea imagen, texto, etc), se lo denomina portador o bien camuflaje.

A diferencia de la criptografía regular, la cual logra la inteligibilidad de la información dada, la esteganografía logra que la información pase completamente desapercibida al ocultar su existencia misma. A pesar de esto, son complementarias, y se puede usar una cifrar el mensaje y la otra para esconder el mensaje cifrado, obteniendo así un nivel de seguridad superior (dado a la dificultad de detectar su existencia).

2.3. Campo de Galois

Un campo de Galois es un cuerpo definido sobre un conjunto finito de elementos. Todos los cuerpos finitos tienen un número de elementos q = pn, para algún número primo p y algún entero positivo n. Para cada cardinalidad q así definida hay una y solo una manera posible de definir un cuerpo finito, por lo que todos los cuerpos finitos de igual orden son isomorfos entre sí.

Los campos de Galois serán entonces utilizados para codificar y decodificar los datos secretos sin pérdidas.

3. Implementación

3.1. Pseudocódigo

A continuación se mostrará el pseudocódigo del algoritmo implementado. Este busca ilustrar sin centrarse en los procedimientos, los distintos módulos desarrollados y su interconexión final.

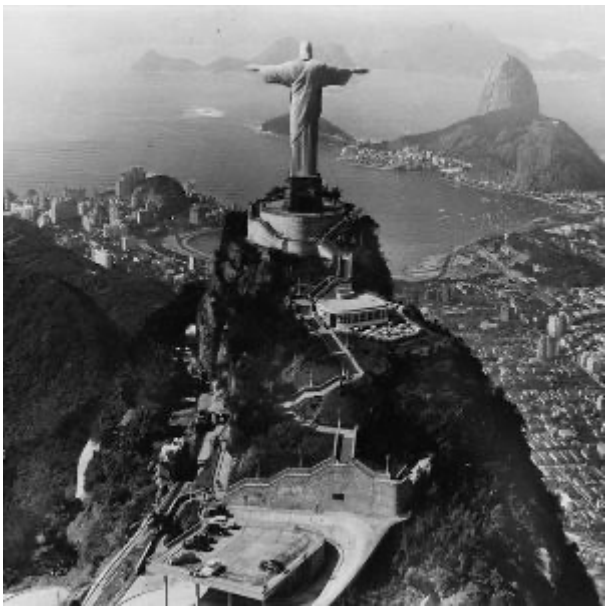
```
START
    // data parsing
    galois_init()
    params <- parse_params(argc, argv)
    image_data <- read_image_extras(params.paths, params.k)
    courrier_images <-
          read_from_file(params.paths, params.courrier_count, params.k, image_data)

    // distribution
    IF (params.action == distribute)
          secret_image <- read_from_file(params.path, params.k, image_data)
          distribute(secret_image, courrier_images, params.courrier_count)
          write_images(courrier_images, params.courrier_count, params.k, image_data)

    // retrieving
    ELSE IF (params.action == retrieve)
          secret_image <- recover(courrier_images, params.k, params.block_count)
          write_image(secret_image, image_data)
    END

    // deallocating used resources
    free_resources()
END
```

3.2. Resolución Particular

Dado un conjunto de imágenes portadoras con un secreto específico y desconocido, se buscará utilizar el algoritmo desarrollado para así obtener el secreto. Este es un caso concreto de uso del algoritmo desarrollado, utilizado para probar su correcto funcionamiento.

Las imágenes portadoras utilizadas son[2]:

A partir de ellas, mediante la utilización del algoritmo desarrollado se obtuvo la siguiente imagen secreta:



[2] No se encuentran en sus resoluciones reales, están a modo ilustrativo.

4. Análisis

4.1. Documento

Sobre el documento se notaron ciertos aspectos que se mencionan a continuación. En un principio la organización del mismo cumple con los estándares mundiales para la presentación de papers, investigaciones. Este se distribuye de manera ordenada en las siguientes partes:

1. Resumen
2. Introducción
3. Esquema propuesto
4. Resultados
5. Conclusiones
6. Agradecimientos
7. Referencias

Las primeras dos secciones son fundamentales para poner al lector al tanto de la problemática a resolver y la historia detrás de los intentos (si es que los hubo). Luego se pasa al desarrollo propio del esquema, el cual se encuentra subdividido en las partes propias del esquema propuesto.

Los resultados y comparaciones del esquema propuesto son necesarios para bajar a la realidad la efectividad del mismo bajo distintas condiciones. Es de esta manera que se puede realmente apreciar las ventajas (y posibles desventajas) del esquema propuesto. Finalmente las conclusiones son esenciales al momento de cerrar y recapitular sobre lo realizado y obtenido a lo largo del trabajo de investigación.

Las referencias finales -utilizadas a lo largo del paper- son requeridas ya que los fundamentos teóricos en los que uno se basa, deben estar correctamente referenciados.

La disposición y ordenamiento del contenido ayuda a esclarecer la comprensión del contenido y se adecúa a la elaboración del trabajo.

Específicamente para el esquema propuesto, en la descripción de los algoritmos, queda muy claro el procedimiento de distribución mayormente a partir de los esquemas de las figuras (1), (2) y (3), donde puede observarse cómo funciona el mismo. Por otro lado, el algoritmo de recuperación no está tan bien logrado como el anterior. Si bien la idea general se logra entender, la explicación de la interpolación de los pares obtenidos no es clara y requirió de una explicación complementaria para comprenderla.

A lo largo de todo el documento se comparte la misma notación de fórmulas, es consistente, lo que favorece su comprensión. Cabe destacar que en la mayoría de las fórmulas, ya sea por un problema de formateo o de formato, los ‘-’ (signo menos) no se encuentran, pudiendo en algunos caso provocar confusión en el lector. Sin embargo, a pesar de ese inconveniente, en líneas generales la nomenclatura conlleva claridad.

Como nota aparte se pudo notar palabras pegadas (sin espacios en el medio), no se trata particularmente de fórmulas, pero sucede transversalmente a lo largo de todo el documento.

El título del documento, "*Sistema de Imagen Secreta Compartida con Optimización de la Carga Útil*", refiere a la intención y capacidad de aumentar esa carga de los datos de los secretos gracias al uso de GF(2^8), que contribuyen a una recuperación sin pérdida de los datos. De ese secreto, existen k píxeles que son ocultados en bloques de 4 píxeles de las imágenes camufladas, lo que da una relación k:4. Si k es mayor que 4, la máxima carga útil de los datos secretos puede ser tan grande como el tamaño de esa imagen camuflaje.

4.2. Generadores y Usos

Como generador a lo largo del esquema fue utilizado y propuesto el GF(2^8). Este tiene la peculiaridad de contribuir a la recuperación de los datos secretos sin pérdida alguna, no resuelto así con los generadores en congruencias de módulo. Esto se debe a que, como el módulo debe ser primo, el número primo más grande en el rango [0, 255] es 251, por lo que no se puede evitar perder parte de la información.

La principal ventaja de no tener dicha pérdida, es la posibilidad de manejar *cualquier* tipo de datos digitales como documentos, imágenes, audio, archivos ejecutables y archivos. Esto se debe principalmente a que las imágenes, a diferencia del resto de los datos digitales, pueden perder un poco de información sin comprometer su utilidad (representación visual). No así, con el resto de los datos digitales.

Por otro lado, una desventaja de GF es su complejidad de implementación respecto a la función módulo, que es de complejidad mínima (de hecho tiene su instrucción respectiva en código de máquina en assembler).

Con respecto al polinomio generador, se puede trabajar con cualquiera siempre y cuando sea uno irreducible de grado n, tenga un número impar de términos y por lo menos una constante (el 1). En el caso de GF(2^8), existen 16 polinomios primitivos posibles.

4.3. Implementación

La implementación realizada se encuentra "hardcodeada" para el tratamiento de imágenes similares. Se utilizan sólo los píxeles del secreto y se genera su encabezado a partir del de las imágenes portadoras.

De desearse guardar un archivo de imagen completo, es decir sus píxeles junto con un encabezado, no habría inconvenientes. Lo que se debería adaptar en la implementación es la generalización de su tratamiento. Ese tratamiento es el utilizado para archivos digitales, no

estrictamente para imágenes, y de esta manera la imagen completa (datos y header) se podría tratar como un documento más.

En el caso de trabajar con imágenes en color, creemos que pueden utilizarse dos enfoques posibles. Por un lado, se podría ocultar cada pixel de 24 bits en un pixel de 24 bits de la portadora siguiendo un enfoque similar al de las imágenes blanco y negro. En este caso, en lugar de modificar 9 bits (8 del pixel y 1 de paridad) por bloque de 2x2 de la portadora, se modificarían 25 o 27 bits (24 del pixel y 1 o 3 de paridad) por bloque de 2x2 de la portadora, teniendo en cuenta que aquí las portadoras también tendrían 24 bits por pixel.

Por otro lado, si sólo la imagen secreta tiene color, se la podría tomar como un array de bytes o un documento, ocultando cada byte en forma individual más allá de la composición del pixel. De ser así, el algoritmo podría ser el mismo.

Con respecto al tamaño de las matrices, creemos que se deben tomar bloques cuadrados para respetar la morfología propia de los píxeles. El tamaño de la matriz de 2x2 es el más eficiente en la proporción del número de píxeles entre la imagen secreta y la imagen camuflaje, siendo este de k:4. Podría tomarse una matriz de 3x3, pasando más desapercibida aún la modificación de las sombras (solo cambiando 1 o 2 bits por pixel), pero la relación pasaría a ser de k:9. En ese caso, podrían ocultarse mayor cantidad de bytes dentro de ese mismo bloque para buscar mantener la relación, pero seguiría siendo peor a la del tamaño original.

El algoritmo en sí no presentó grandes dificultades basándonos en la explicación y enumeración de pasos dispuestos en el paper, especialmente con la parte de codificación. Por su parte respecto a la decodificación se presentaron más complejidades que llevaron a requerir un análisis algo más profundo, en específico para el uso de Lagrange. Pero finalmente terminó sacándose adelante gracias a la complementación con los fundamentos teóricos dados en clase por la cátedra.

Por su lado en cuanto a los campos de Galois, fue de gran ayuda el encontrar una librería en C que lo implemente y resuelva. Luego sólo restó interpretar su utilización, cambiar su polinomio primitivo y adaptarla a nuestro código.

Justamente como fue mencionado previamente, se podría extender el algoritmo implementado para que realice un tratamiento sobre documentos digitales de forma genérica, es decir, sin discriminar su formato o extensión.

Sin dudas que se podrían hacer mejoras de velocidad en algunas partes, y perfeccionamiento en el manejo de memoria en disco, logrando optimizarlo al máximo. Con eso se buscaría permitir el manejo constante de archivos de cualquier tipo y dimensión, garantizando a su vez que no existan pérdidas de tiempo ni espacio.

De hecho, incluso sería opción viable encargarse de implementar nuevamente las funciones de GF(2^8), de forma de tener control de todo lo que compone al algoritmo y tener acceso a una completa optimización del mismo. Sin embargo, claramente, esta opción ya escapa al alcance de este trabajo.

4.4. Posibles Aplicaciones

Este tipo de algoritmos despiertan múltiples potenciales aplicaciones entre las que se nos ocurren algunas más específicas y otras más bien genéricas. Entre ellas surgen las siguientes, abarcando distintos motivos.

- ➔ **Informativo.** Distribución de información y fotos sensibles a diversos medios de comunicación forzando y obligando a que trabajen en conjunto. Esto con el objetivo de lograr una imparcialidad al momento de investigar, mostrar y contar un hecho.
- ➔ **Lúdico.** Búsqueda del tesoro virtual en donde se necesitan múltiples imágenes para destrabar la llave que lleva a la siguiente etapa. Incluso hasta se puede hacer que esa última sea a su vez portadora para la siguiente etapa.
- ➔ **Almacenamiento privado.** Ocultamiento de información y documentos en la librería de fotos pudiendo almacenarlas en cualquier dispositivo electrónico, o bien en la nube (Google, Apple, Microsoft).

Estos ejemplos son algunos de los que fueron surgiendo a lo largo del trabajo. Claramente las posibles aplicaciones del esquema no se encuentran limitadas a estas opciones. Dicho esquema tiene una infinidad de usos tanto legales como ilegales, y usos que son propios tanto de la esteganografía (como fue mencionado al comienzo del trabajo) como de los secretos compartidos.

5. Conclusiones

En un principio nos incumbe resaltar que de tenerse k cantidad de portadoras, la entropía del secreto dados k-1 (o menos) portadoras se mantiene en h(s), es decir, no se revela información alguna. Mientras que a partir de k portadoras la entropía se vuelve nula: tenemos certezas de cuál es la información oculta.

Cabe destacar también, que a diferencia del esquema original de Shamir donde el secreto es únicamente el coeficiente independiente del polinomio, en el esquema analizado el secreto es el polinomio entero. Es por esto que en el esquema de Shamir, no es posible evaluar el polinomio en 0 ya que se obtendría el secreto propiamente dicho. Por otro lado, en el esquema analizado no aplica esta restricción ya que se obtendría solamente uno de los valores para realizar la interpolación.

Finalmente, bajo nuestro punto de vista, elegimos pensar a las imágenes portadoras como un medio para un fin y no un fin en sí mismo. Se trata de llaves digitales fáciles de esconder en plena vista para quienes no son experimentados en el tema. Sumado a esto es importante destacar que por sí solas no *generan* información.