Preinforme de Proyecto:

Desencriptado de Imágenes BMP usando Qt y C++

1. Marco Teórico

Transformaciones a nivel de bit\*: Son operaciones realizadas directamente sobre los bits individuales que componen los datos digitales, como los valores RGB de los píxeles en una imagen. Las más comunes incluyen:

XOR (Operación exclusiva OR): Operación binaria que devuelve 1 si los bits comparados son diferentes. Es reversible: aplicar XOR dos veces con la misma clave devuelve el valor original.

Rotación de bits: Desplazamiento circular de los bits hacia la izquierda o derecha. Es diferente al desplazamiento simple porque los bits que salen de un extremo regresan por el otro.

Desplazamiento (Shift): Desplazamiento simple de los bits hacia la izquierda o derecha. A diferencia de la rotación, los bits que se desplazan fuera del límite se pierden.

Enmascaramiento\*: Proceso que consiste en modificar una imagen utilizando una máscara (otra imagen de este o menor tamaño). En este proyecto, se realiza sumando los valores RGB de la imagen transformada y los de la máscara a partir de una posición de desplazamiento definida por una semilla.

Archivos BMP y representación RGB888\*: Un archivo BMP almacena imágenes sin comprimir. En formato RGB888, cada píxel se representa con 3 bytes: un byte para cada canal de color (Rojo, Verde y Azul). Es importante procesar las imágenes sin estructuras ni STL, utilizando punteros y arreglos para cumplir las restricciones del proyecto.

Desencriptación basada en simulación inversa\*: Al no conocer el orden exacto de las transformaciones, el desencriptado debe simular distintas combinaciones y compararlas con los datos del enmascaramiento almacenados en los archivos .txt. Al encontrar coincidencias exactas, se puede deducir la transformación aplicada y deshacerla progresivamente.

2. Análisis del Problema

El objetivo principal del proyecto es reconstruir una imagen original (IO) a partir de una imagen encriptada (ID) que ha sido sometida a una serie de transformaciones a nivel de bits (XOR, rotaciones, desplazamientos), aplicadas en orden desconocido. Además, luego de cada transformación, se aplicó un enmascaramiento que alteró parte de la imagen con una máscara (M), cuyo resultado se almacenó en archivos de texto (M1.txt, M2.txt, etc.).

Cada archivo .txt contiene:

- Una línea inicial con la semilla (offset s) usada para enmascarar.

- Una lista de tripletas RGB que representan la suma entre los valores de la imagen transformada (ID) en la posición desplazada y la máscara M.

El reto consiste en identificar cuáles operaciones se aplicaron, en qué orden, y revertirlas para recuperar la imagen original, utilizando solo los archivos .txt, la imagen modificada y la máscara, sin conocimiento previo del orden de transformaciones.

3. Consideraciones dadas por el profesor

- Cada archivo .txt corresponde a una sola transformación aplicada antes del enmascaramiento.

- El programa debe analizar los archivos .txt en orden secuencial, desde el primero hasta el último, reconstruyendo paso a paso el proceso de encriptación.

- La máscara está completamente contenida dentro de los límites de la imagen y se superpone\* sobre ella.

4. Solución Propuesta

Se propuso una solución modular compuesta por las siguientes fases:

Fase 1: Operaciones a Nivel de Bit

Implementación de funciones para aplicar XOR, rotación (izquierda y derecha hasta 8 bits), y desplazamientos (shifts) en cada canal de los píxeles de una imagen BMP, usando punteros y arreglos dinámicos sin STL ni estructuras.

Fase 2: Experimentación con Transformaciones

Aplicación de las operaciones bit a bit sobre pares de imágenes para observar el impacto visual y estadístico, con el fin de comprender cuál es la transformación más destructiva o reversible según el contexto.

Fase 3: Verificador de Enmascaramiento

Desarrollo de una función para verificar si una imagen candidata es la fuente del archivo .txt correspondiente, utilizando la relación: S(k) = ID (k + s) + M(k).

Fase 4: Deducción del Orden de Transformaciones

Prueba de combinaciones posibles de transformaciones, comparando los resultados simulados del enmascaramiento con los datos reales de los archivos .txt. Esto permite reconstruir la cadena de transformaciones y aplicar las operaciones inversas hasta recuperar IO.

Fase 5: Identificación de Transformaciones por Archivo de Texto

Cada archivo txt representa exactamente una transformación a nivel de bit aplicada sobre la imagen antes del enmascaramiento. El programa debe analizar los datos del archivo, aplicar una transformación tentativa sobre la imagen modificada, simular el enmascaramiento y verificar si los resultados coinciden.

Si la simulación es exitosa, se concluye que esa fue la transformación utilizada. Se registra el tipo y parámetros (por ejemplo, "rotación 3 bits a la derecha") y se reconstruye la imagen intermedia. Esto permite deducir paso a paso la cadena completa de transformaciones.

Estrategia secuencial

Se aplicará un enfoque de comparación secuencial: el programa comenzará con el primer archivo .txt y comparará los resultados del enmascaramiento simulado con los datos reales. Si coinciden, se confirma la transformación correspondiente. Luego se actualiza la imagen base y se repite el proceso con el siguiente archivo, hasta llegar al último. Esta estrategia permite reconstruir el orden exacto en que se aplicaron las transformaciones.

5. Esquema de Desarrollo

- Carga de imágenes BMP (RGB888)

- Lectura de archivos .txt con semilla y valores RGB

- Implementación de operaciones XOR, rotación y shift

- Verificador de enmascaramiento

- Sistema de prueba de combinaciones (detección del orden)

- Reconstrucción total de la imagen original

6. Preguntas Clave para Guiar el Desarrollo

- ¿Cómo afecta el orden de operaciones a la posibilidad de desencriptar?

- ¿Puede una combinación de transformaciones dar el mismo resultado que otra?

- ¿Es posible reconstruir sin conocer la máscara?

- ¿Cómo se detecta que se ha aplicado la transformación correcta en un archivo txt?

- ¿Qué pasa si una imagen transformada parece coincidir con más de un archivo txt?

7. Conclusiones Parciales

Hasta este punto, se ha desarrollado e implementado satisfactoriamente la Fase 1 del proyecto. Se han definido claramente los módulos necesarios para continuar y se han identificado preguntas críticas que orientan la estrategia de desencriptación.

El siguiente paso es construir el verificador de enmascaramiento (Fase 3), que permitirá validar hipótesis sobre la secuencia de transformaciones y avanzar hacia la reconstrucción completa de la imagen original.