

Análisis

Análisis del Problema

En este desafío se nos plantea el reto de reconstruir una imagen original a partir de una versión final que ha pasado por varias transformaciones a nivel de bits. Las transformaciones incluyen operaciones como:

- XOR con una imagen generada aleatoriamente (IM).
- Rotaciones de bits.
- Desplazamientos de bits.

Después de cada transformación (excepto la última), se aplicó una técnica de enmascaramiento, usando una máscara de color (M) y una posición aleatoria (semilla). El resultado de ese enmascaramiento fue almacenado en archivos `.txt`, que sirven como única pista para deducir las operaciones aplicadas y su orden.

El reto consiste en:

- Identificar qué operaciones se aplicaron.
- Determinar el orden correcto.
- Verificar que el resultado de cada paso coincide con los archivos de enmascaramiento.
- Reconstruir así la imagen original.

Enfoque y Alternativa de Solución

Nuestra solución consiste en aplicar lógica y programación en C++ para simular las posibles operaciones aplicadas y verificar si los resultados intermedios coinciden con los datos en los archivos `.txt`.

Para esto, planteamos los siguientes pasos:

1. Leer imágenes y datos:

- Cargar la imagen transformada (I_D), la imagen I_M, y la máscara M.
- Leer los archivos `.txt` que contienen el resultado del enmascaramiento (suma de imagen transformada y máscara).

2. Implementar operaciones a nivel de bits:

- XOR entre dos imágenes.
- Rotación de bits (hasta 8 bits).
- Desplazamiento de bits (hasta 8 bits).

3. Simular transformaciones posibles:

- Generar combinaciones posibles de transformaciones (según los archivos de enmascaramiento disponibles).

- Aplicarlas en distintos órdenes sobre I_D o I_M.

4. Verificar enmascaramientos:

- Comparar el resultado intermedio de cada transformación con los archivos `.txt`, usando la máscara y el desplazamiento indicado.

5. Reconstruir la imagen original:

- Una vez encontrado el orden correcto de transformaciones, aplicar el proceso inverso para obtener la imagen inicial.

Esquema de Trabajo

1. Cargar imágenes y archivos de enmascaramiento.

2. Implementar funciones para:

- XOR

- Rotación de bits

- Desplazamiento de bits

3. Implementar verificación del enmascaramiento.

4. Probar combinaciones posibles de transformaciones.

5. Identificar el orden correcto.

6. Reconstruir la imagen original aplicando las operaciones inversas.

Algoritmos a Implementar

- aplicarXOR(img1, img2)

Aplica una operación XOR bit a bit entre dos imágenes.

- rotarBitsDerecha(pixel, n)

Rota los bits hacia la derecha.

- rotarBitsIzquierda(pixel, n)

Rota los bits hacia la izquierda.

- desplazarBitsDerecha(pixel, n)

Desplaza los bits hacia la derecha.

- desplazarBitsIzquierda(pixel, n)

Desplaza los bits hacia la izquierda.

- verificarEnmascaramiento(I_D_transformada, M, archivo_txt)

Verifica si la imagen transformada cumple con los criterios.

- reconstruirImagenFinal(I_D, I_M, M, archivos_txt)

Reconstruye la imagen final.