

Análisis y diseño de la solución del problema.

El reto que se propone en el Desafío 1 es de desarrollar un programa en el cual debemos de reconstruir la imagen BMP dándole transformaciones bit a bit de la imagen hasta encontrar el resultado de la imagen original esto se hace para recuperar tipo BMP original (IO) podemos percatarnos que no tenemos conocimiento del orden después de cada transformación, se debe aplicar un proceso de enmascaramiento usando una máscara M y un desplazamiento aleatorio s, y el resultado es almacenado en archivos .txt.

Según mi investigación del problema lo que se requiere es realizar ingeniería inversa aplicando operaciones inversas en el orden correcto para recuperar IO usando la imagen final ID, La imagen IM usada como insumo para las operaciones XOR, la máscara M, Los archivos de rastreo con información de enmascaramiento y esto sin reconocer el orden real de las operaciones

Algunos requisitos son usando C++ con Qt pero no se puede utilizar STL ni estructuras.

Las consideraciones técnicas son de las imágenes BMP de 24 bits cada 3 bytes (RGB) y las operaciones posibles son XOR (bit a bit entre imágenes), rotación de bits a derecha e izquierda por pixel y desplazamientos y el enmascaramiento es:

$$S(k) = ID(k + s) + M(k)$$

Donde S es un desplazamiento aleatorio, y la suma es modulo 256

Mi estrategia de Resolución es:

1. Cargar imagen ID (resultado final)
2. Cargar IM (imagen utilizada para XOR)
3. Cargar M(máscara)
4. Leer archivos de rastreo .txt
5. Implementación de funciones de transformación
6. Verificar el enmascaramiento
7. Deducción del orden de transformación
8. Reconstrucción