****

**Desafío I**

**Informe de procesamiento digital de imágenes - Desafío I**

Santiago Galeano García

Alejandro Ayala Usuga

Aníbal Guerra y Augusto Salazar

Profesores del departamento de ingeniería electrónica y telecomunicaciones

Universidad de Antioquia  
Departamento de Ingeniería Electrónica y de Telecomunicaciones

Medellín

2025-1

**Introducción**

El procesamiento digital de imágenes es ciertamente necesario para muchas de las aplicaciones del mundo actual, así mismo, este desafío tiene como objetivo afianzar los conceptos necesarios para desarrollar habilidades que podrían ser imprescindibles para nuestro mundo profesional. Este desafío consiste a groso modo desencriptar una imagen distorsionada teniendo en cuenta el proceso de encriptamiento desarrollado por los profesores, este trata básicamente de modificar una imagen original () con un contenido de información coherente ( imagen con información valiosa ) a partir de transformaciones ( operaciones) como, XOR, desplazamientos, rotaciones a nivel de bits mediadas por una imagen generada de manera aleatoria () que distorsiona la imagen original, luego se elige de manera aleatoria por cada transformación un pixel al cual se le va a aplicar un enmascaramiento por medio de sumas de canales RGB que contienen información por cada transformación efectuada.

**Análisis del problema**

Teniendo en cuenta lo anterior, el análisis del problema se desarrolla en consecuencia de los datos o insumos proporcionados por el reto, así mismo, pensamos en gestionar un método que revirtiera los resultados de las transformaciones desconocidas en medida de lo posible con la información proporcionada por el enmascaramiento (M):

* Resultado del enmascaramiento después de cada transformación.
* Se conoce el valor de desplazamiento (s) en cada etapa.
* Se obtiene la máscara M y la salida final.

Asumiendo los insumos proporcionados pensamos en un método de reconstrucción progresiva de los datos con verificación paso a paso de la siguiente manera, el método desarrollado consiste en revertir cada transformación en el orden inverso al que fue aplicada inicialmente utilizando la información de los archivos .txt proporcionados después de cada enmascaramiento así:

**Primera etapa**  
  
**Lectura de insumos :** Imagen encriptada , archivos .txt y máscara (M).

**Desenmascaramiento**

Sabiendo que la ecuación del enmascaramiento está dada por que nos indica el valor del enmascaramiento en función de una posición de la imagen desplazada y distorsionada para un pixel aleatorio en que a su vez está asociado a la máscara en una posición **k**, podríamos hacer la operación inversa o desenmascaramiento tal que así:

Obteniendo los valores previos de la imagen distorsionada, es decir los valores obtenidos justo después de cada transformación. para esto declaramos una opción de función tentativa para desenmascarar la imagen usando la máscara **M** y el desplazamiento tal que así, usando punteros para la utilización de memoria dinámica:

**unsigned char\* desenmascarar (unsigned char\* enmascarado,unsigned char\* mascara, int desplazamiento, int ancho, int alto )**

**Operaciones inversas bit a bit**

Para esta etapa se tiene que encontrar el patrón desconocido de las operaciones de distorsión de imagen que se le hacen a la imagen original y revertirlas, considerando las limitaciones de rotaciones máximas de 8 bits y condiciones intrínsecas a las operaciones XOR y desplazamientos.

Asimismo ya habiendo descifrado el patrón hacer operación inversa es decir desplazamiento hacia la derecha si antes se desplazó hacia la izquierda, rotación de izquierda a derecha de N (máximo 8 bits) bits si antes se tuvo rotación de derecha a izquierda y así sucesivamente, sin embargo para las operaciones de desplazamiento realmente no tendrán relevancia en este caso como las operaciones de rotaciones y XOR , debido a la pérdida de información que hay en el procesamiento de esta operación por lo que las funciones prueba principales serán de rotación y de XOR de la siguiente manera:

**void rotacion\_inversa (unsigned char\* imagen, int ancho, int alto, int bits)**

**void revertir\_XOR(unsigned char\* imagen, unsigned char\* IM, int ancho, int alto)**

**Verificación**

Finalizando el tratamiento de la imagen distorsionada se requiere verificación para saber si realmente coincide con la imagen desencriptada, para esto es simplemente comprar el resultado del paso con los datos en el archivo .txt y si los valores RGB coinciden con la suma esperada la verificación por paso puede ser exitosa, así de manera iterativa hasta obtener la imagen original. La funciones propuestas o tentativas estarían dadas por:

**bool verificar\_txt(unsigned char\* imagen, std::string rutaTxt, unsigned char\* mascara);**

**void reconstruirImagen(std::vector<std::string> archivosEnmascaramiento, unsigned char\* IM, unsigned char\* mascara);**

En resumen, la estructura de nuestro programa sigue una secuencia lógica diseñada para alcanzar el objetivo planteado en el desafío. El proceso comienza con la carga de la imagen distorsionada junto a su respectiva máscara. A continuación, se leen los archivos **.txt** que contienen la información necesaria para continuar con la reconstrucción. Luego, se procede a aplicar el proceso de desenmascaramiento, seguido por la inversión de las transformaciones realizadas previamente. Finalmente, se exporta la imagen ya reconstruida, completando así el flujo completo del sistema.