DECODIFICACIÓN Y DESCOMPRESIÓN DE MENSAJES CIFRADOS

Desafío 1

Daniela Escobar Velandia
Ingeniería Electrónica
Universidad de Antioquia
Medellín, Colombia
daniela.escobarv@udea.edu.co

I. DESARROLLO

Este informe presenta un análisiS del problema planteado, así como las consideraciones clave para el diseño e implementación de una solución. El objetivo principal es descifrar un mensaje que ha sido comprimido y luego encriptado, utilizando un fragmento del mensaje original como pista. Para ello, se deberá identificar el método de compresión (RLE o LZ78) y los parámetros de encriptación (rotación de bits 'n' y clave 'K'). Luego, se aplicarán los procesos inversos para reconstruir el mensaje original.

A. Análisis del problema y consideraciones para la alternativa de solución propuesta.

El desafío es un ejercicio de ingeniería inversa que combina dos procesos: compresión (RLE o LZ78) y encriptación (rotación de bits + XOR). El enfoque de la solución debe ser deductivo, ya que se parte de un mensaje final y un fragmento conocido del original para determinar los parámetros desconocidos y revertir las transformaciones.

El proceso de solución se puede dividir en tres etapas principales:

1) **Descifrado de Parámetros:** Se debe iterar a través de las combinaciones posibles de los parámetros de encriptación ('n' y K') y los métodos de compresión (RLE y LZ78) para encontrar la combinación correcta. La rotación de bits 'n' puede variar de 1 a 7, mientras que la clave 'K' puede ser cualquier valor de un solo byte (0-255).

2) **Desencriptación:** Una vez que se encuentran los parámetros correctos ('n' y 'K'), se aplicarán las operaciones inversas a cada byte del mensaje comprimido y encriptado. Esto significa primero aplicar una operación XOR con la clave 'K' y luego una rotación de bits a la derecha en 'n' posiciones para recuperar el mensaje comprimido original.

3) **Descompresión:** Con el mensaje ya desencriptado, se aplicará el algoritmo de descompresión correspondiente (RLE o LZ78) para reconstruir el texto original completo.

B. Esquema de tareas para el desarrollo:

Se propone un esquema de tareas que aborda el problema de manera modular y estructurada:

1) **Módulo de Descifrado de Parámetros:**
   - Función *desencriptar_byte(byte, n, K):* Aplica las operaciones inversas (XOR y rotación a la derecha) a un solo byte.
   - Función *probar_metodo_RLE(mensaje_encriptado, fragmento_original):*
     - Itera a través de los posibles valores de 'n' y 'K'.
     - Para cada combinación, desencripta el mensaje completo.
     - Intenta descomprimir el mensaje desencriptado con el algoritmo RLE.
     - Compara si el inicio del mensaje descomprimido coincide con el *fragmento_original*.
     - Si hay una coincidencia, retorna los parámetros (RLE, n, K).
   - Función *probar_metodo_LZ78(mensaje_encriptado, fragmento_original):*
     - Similar a la función RLE, pero utiliza el algoritmo de descompresión LZ78.
     - Retorna los parámetros (LZ78, n, K) si encuentra una coincidencia.

2) **Módulo de Descompresión RLE:**
   - Función *descomprimir_RLE(mensaje_comprimido):*
     - Recorre el mensaje comprimido, identificando la longitud y el símbolo.
     - Reconstruye el mensaje original repitiendo cada símbolo la cantidad de veces indicada.

3) **Módulo de Descompresión LZ78:**

- Función *descomprimir_LZ78(mensaje_comprimido):*
  - Inicializa un diccionario vacío.
  - Lee la secuencia de pares (índice, carácter) del mensaje comprimido.
  - Para cada par, reconstruye la cadena (prefijo + caracter) y la añade al diccionario.
  - El diccionario se debe manejar de forma dinámica.
  - Imprime el mensaje reconstruido.

4) **Módulo Principal:**

- Llama a *probar_metodo_RLE* y si no encuentra una solución, llama a *probar_metodo_LZ78*.
- Una vez que se encuentran los parámetros, utiliza las funciones de desencriptación y descompresión correspondientes para obtener el mensaje final.
- Imprime los resultados: el método de compresión, los parámetros de encriptación y el mensaje original reconstruido.