Universidad de Antioquia
Facultad de Ingeniería – Ingeniería Electrónica
Informática II
Informe Desafío 1
Brayan Camilo Silva Porras
Hamilton Alexander Suarez Pérez
2025-2

INFORME

La empresa Informa2 requiere un sistema capaz de recuperar mensajes originales a partir de archivos que han sido sometidos a procesos de compresión y encriptación, sin conocer de antemano los parámetros exactos utilizados. Dada la naturaleza del problema, donde existen 2 métodos de compresión posibles (RLE o LZ78), 7 valores de rotación (1-7 bits), y 256 claves XOR posibles (0x00-0xFF), el espacio de búsqueda total comprende 3,584 combinaciones que deben evaluarse de forma exhaustiva. Adicionalmente, se dispone únicamente de un fragmento del mensaje original (pista) como referencia para validar la correcta recuperación del texto.

Para resolver este desafío, se diseñó e implementó en lenguaje C++ un programa modular que permite leer los archivos encriptados, aplicar desencriptación mediante rotación de bits y operaciones XOR, y posteriormente descomprimir la información utilizando los algoritmos RLE y LZ78. El presente informe describe el proceso de análisis, diseño, implementación y pruebas de dicho sistema, así como los resultados obtenidos en la recuperación de los textos originales.

ANÁLISIS

Algoritmo RLE (Run-Length Encoding)

El algoritmo RLE es un método de compresión sin pérdida que se basa en la representación de secuencias repetitivas de símbolos mediante un contador y el símbolo en cuestión. En lugar de almacenar cada carácter repetido, se guarda únicamente la cantidad de repeticiones seguida del símbolo.

LZ78

Este algoritmo crea un diccionario dinámico a medida que procesa el texto, guardando pares formados por un índice que apunta a una secuencia previa y el nuevo símbolo que la extiende. De esta manera logra identificar y reutilizar patrones repetidos, aunque no estén uno al lado del otro, obteniendo mejor compresión en textos variados, aunque requiere más memoria y procesamiento que RLE.

Rotación de bits

La rotación de bits es una operación a nivel binario que desplaza todos los bits de un dato hacia la izquierda o la derecha, haciendo que los que "salen" por un extremo vuelvan a entrar por el otro. A diferencia de un corrimiento normal, donde los bits que se desplazan se reemplazan por ceros, en la rotación no se pierde información. Esta técnica se utiliza como parte del proceso de encriptación porque altera la representación binaria de los datos de manera reversible: aplicando la rotación inversa se puede recuperar el valor original.

XOR

La operación XOR (o "OR exclusivo") es una función lógica que compara dos bits y devuelve 1 solo si son diferentes, y 0 si son iguales. Aplicada a nivel de bytes, permite encriptar información combinando cada dato con una clave: el resultado parece distinto al original, pero se puede recuperar aplicando nuevamente la misma operación con la misma clave. Esto se debe a que XOR es reversible, lo que la hace una herramienta sencilla y eficaz para ocultar datos en procesos de encriptación básicos.

Al analizar los archivos desencriptados se identificó que la información comprimida se encontraba organizada en bloques de **tres bytes**. En el caso de la compresión RLE, se observó que el primer byte correspondía a información irrelevante ("basura"), mientras que los dos siguientes contenían los datos necesarios: la cantidad de repeticiones y el símbolo a expandir. Mientras que para la compresión LZ78 los 3 datos eran relevantes, dando un índice alto, un índice bajo y el símbolo.

Esta estructura permitió reconstruir correctamente los textos originales.

En el problema se desconoce cuál de los dos métodos de compresión fue utilizado (RLE o LZ78), el valor exacto de la rotación de bits aplicada (un entero entre 1 y 7) y la clave empleada en la operación XOR (un valor entre 0x00 y 0xFF). Al combinar estas posibilidades, el espacio de búsqueda total asciende a 3,584 configuraciones distintas (2×7×256), las cuales deben evaluarse de manera exhaustiva hasta encontrar la que permita recuperar correctamente el mensaje original.

Para abordar el problema se cuenta con un archivo que contiene el mensaje comprimido y encriptado, un fragmento del texto original que sirve como pista para validar la correcta recuperación, y la especificación de los algoritmos empleados en el proceso: compresión mediante RLE o LZ78, combinada con operaciones de rotación de bits y encriptación XOR.

Estrategia

Ya que el espacio de búsqueda es finito y no muy grande es manejable y ya que no existe una forma directa de calcular los parámetros el método de búsqueda exhaustiva es viable

Para cada combinación de parámetros se inicia aplicando una clave para el XOR luego una rotación de bit y se descomprime usando RLE para luego buscar la pista dentro del texto dado, si no es positivo se usa LZ78, si tampoco es positivo se prueba el siguiente número en rotación de bit hasta usar los 7 posibles y luego rotando la clave del XOR hasta usar también las 256 posibles. Cuando finalmente se encuentra se guarda el texto encontrado en un nuevo archivo y se muestra en pantalla la clave, el número de rotación y el método de compresión utilizado.

Los retos que enfrentamos fue con la información que se recibía al final y el manejo de información basura un poco de dificultad con ciclos que se iban al infinito y nunca hallaban nada y la organización de los módulos, el uso de los punteros dificultaba un poco la lectura del código para interpretar lo que se hacía en un inicio y el uso de software nuevo para nosotros que al inicio dificultaba el trabajo en equipo de manera remota pero se pudo evolucionar todo poco a poco probando desde los métodos de lectura hasta lograr el método de compresión lz78 cada uno organizado en sus propios módulos para mantener un orden en el código