Práctica 1. Criptología. Cifrado simétrico (1 punto)

Grupos

Esta práctica se realiza en grupo (mismos integrantes que para el trabajo tutelado).

Todos los ejercicios deben ser realizados por todos los integrantes del grupo, aunque en la memoria, bastará con que se incluyan las evidencias de uno de ellos.

NOTA: En la nomenclatura de los archivos, GN hace referencia a su grupo de prácticas (p.ej: 1.1.1)

1) Instalen Cryptool 1 y realicen pruebas de cifrado y descifrado con los algoritmos vistos en clase de teoría.
    a. Busquen y documenten las opciones de Cryptool para el criptoanálisis de algoritmos de sustitución monoalfabeto
2) Seleccionen tres fragmentos de texto con las siguientes características:
    - Idioma español
    - Con sentido (idealmente, el fragmento de un libro, mensaje, etc.)
    - Entre 1000 y 1500 caracteres
    - El texto sólo puede contener caracteres del alfabeto mostrado en la Tabla 1. Si en el texto original hay otros caracteres, deben ser sustituidos o eliminados. Por ejemplo, una 'Ñ', puede ser reemplazada por 'N', 'NH' o 'NN'. Deben eliminarse las tildes, diéresis, espacios y signos de puntuación.

    a) Guarden estos fragmentos como `GN_fragmento1.txt`, `GN_fragmento2.txt` y `GN_fragmento3.txt`.

Tabla 1. Alfabeto válido

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

3) Ideen e implementen en Python un algoritmo de cifrado **simétrico** por **sustitución monoalfabeto**.
    a) Expliquen el mecanismo de funcionamiento del algoritmo.
        - El algoritmo únicamente debe aceptar como entrada caracteres del alfabeto mostrado en la Tabla 1. Esos son también los únicos caracteres válidos para la salida.
    b) Guarden el código implementado en un archivo denominado `monoalfabeto.py`. (o `monoalfabeto.zip`, en caso de que se utilice más de un archivo).
        - El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `monoalfabeto.py`
    c) Indiquen la sintaxis de uso del comando `monoalfabeto.py`.

4) Cifren, utilizando el código implementado en el ejercicio 3), el texto de los archivos `GN_fragmento1.txt` y `GN_fragmento2.txt`.
    a) Indiquen en la memoria las instrucciones para descifrar los textos (comando y parámetros necesarios)
    b) Guarden los textos cifrados como `GN_fragmento1.mono` y `GN_fragmento2.mono` respectivamente y súbanlos a la carpeta `P1_Retos` en Teams.

5) Implementen en Python el algoritmo de Vigenère.
   a) Expliquen el mecanismo de funcionamiento del algoritmo.
      - El algoritmo únicamente debe aceptar como entrada caracteres del alfabeto mostrado en la Tabla 1. Esos son también los únicos caracteres válidos para la salida.
   b) Guarden el código implementado en un archivo `vigenere.py` (o `vigenere.zip`, en caso de que se utilice más de un archivo).
      - El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `vigenere.py`
   c) Indiquen la sintaxis de uso del comando `vigenere.py`.

6) Cifren, utilizando el código implementado en el ejercicio 5) el texto del archivo `GN_fragmento3.txt`.
   a) Indiquen en la memoria las instrucciones para descifrar los textos (comando y parámetros necesarios)
      - La contraseña puede tener, como máximo, 7 caracteres
   b) Guarden el texto cifrado como `GN_fragmento3.vig` y súbanlo a la carpeta `P1_Retos` en Teams.

7) Seleccionen dos fragmentos de tipo `*.mono` de otro grupo.
   a) En la memoria se debe indicar el nombre del fragmento seleccionado. No es necesario incluir el fragmento seleccionado.
   b) Traten de obtener el texto en claro correspondiente, aplicando análisis de frecuencias. Indiquen los pasos que se han seguido.
   c) Muestren el texto descifrado.
   d) Indiquen la clave de cifrado
   e) Expliquen, si es posible, el funcionamiento del algoritmo de cifrado analizado.

8) Seleccionen un fragmento de tipo `*.vig` de otro grupo.
   a) En la memoria se debe indicar el nombre del fragmento seleccionado. No es necesario incluir el fragmento seleccionado.
   b) Traten de obtener el texto en claro correspondiente, aplicando Kasiski. Indiquen los pasos que se han seguido.
   c) Muestren el texto descifrado.
   d) Indiquen la clave de cifrado.

9) Implementen en Python una herramienta que automatice lo máximo posible el criptoanálisis usando Kasiski.
   a) Expliquen el mecanismo de funcionamiento del algoritmo.
   b) Guarden el código implementado como `kasiski.py` (o `kasiski.zip`, en caso de que se utilice más de un archivo).
      - El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `kasiski.py`
   c) Indiquen la sintaxis de uso del comando `kasiski.py`.
   d) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.

10) Implementen en Python el algoritmo RC4, con las siguientes consideraciones:
    - Debe mostrar por pantalla el valor inicial de S, el valor de S después de la fase inicial y cómo va cambiando S con la generación del keystream. La representación de los valores debe hacerse en formato binario.
    - La clave debe introducirse en formato hexadecimal
    - Para el cifrado:
        - El texto a cifrar se introducirá por consola y se irá cifrando y mostrando el resultado, carácter a carácter, con cada pulsación del teclado.
        - Los caracteres se interpretarán como ASCII.
        - Para cada carácter introducido, se mostrará su codificación en ASCII, en binario, el valor del keystream en binario y el resultado de la operación de cifrado en binario y en hexadecimal.
    - Para el descifrado:
        - El texto a descifrar se introducirá por consola en formato hexadecimal.
        - El descifrado se realizará en un solo paso.
        - El resultado se mostrará en formato ASCII.

    a) Guarden el código implementado en un archivo denominado `RC4.py.` (o `RC4.zip`, en caso de que utilice más de un archivo).
        - El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `RC4.py`
    b) Indiquen la sintaxis de uso del comando `RC4.py`.
    c) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.

Modo y fecha de entrega

Los archivos cifrados correspondientes a los ejercicios 4 y 6 deben subirse a la carpeta P2/Retos de Teams, no más tarde del **viernes 30 de septiembre a las 13:30**.

La memoria y resto de archivos, se entregarán vía Moodle, no más tarde del lunes **10 de octubre a las 10:00**. Sólo una entrega por grupo. Un único archivo P1_GN.zip, que debe contener:

- Memoria, en formato .pdf, explicando lo que se pide en cada uno de los apartados.
    - Copien el enunciado de cada ejercicio tal cual, sin modificaciones, y respondan dentro de cada apartado (a), b), etc.)
- Archivos indicados en el enunciado de la práctica, respetando la nomenclatura indicada.

IMPORTANTE: respondan con claridad a lo que se pide en cada apartado y respeten la nomenclatura de los archivos, así como las indicaciones del modo de entrega.

Defensa

Posteriormente a la entrega, el profesor indicará una fecha para la defensa de la práctica. En la defensa deben estar presentes todos los integrantes del grupo.