Práctica 1. Criptología y aplicaciones (1,4 puntos)

Esta práctica se realiza en grupos de 4 personas. Es obligatorio inscribirse en el documento habilitado a tal efecto en Teams **antes del jueves 26 de septiembre a las 17:00 horas**.

NOTAS:

1) 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.
2) En la nomenclatura de los archivos, reemplacen *GN* por el código asignado a su grupo de prácticas (p.ej: 1.1).

Criptografía clásica

1) Instalen Cryptool 1
    a. Realicen pruebas de cifrado y descifrado con los algoritmos vistos en clase de teoría: César, Rot-13, Playfair, …. Aporten evidencias.
    b. 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 reemplazados. 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 para su posterior entrega como *GN*`_fragmento1.txt`, *GN*`_fragmento2.txt` y *GN*`_fragmento3.txt`, dentro de una carpeta `fragmentos`.

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. Incluyan en la explicación un breve ejemplo de cifrado (una palabra o una frase corta) y otro de descifrado.
        - 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.
        - El tamaño del texto cifrado no puede superar en más de 2 veces el tamaño del texto plano.
        - Deben utilizar los conocimientos adquiridos en clase de teoría, para proponer un algoritmo lo más robusto (y a la vez sencillo) posible.
    b) El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `monoalfabeto.py` y pasándole los argumentos y parámetros necesarios.
        - El parámetro `--help` debe mostrar la sintaxis y las instrucciones de uso. Muestren en la memoria la salida del comando con este parámetro.
        - Guarden el código implementado para su posterior entrega en una carpeta denominada `monoalfabeto`.

c) Cifren, utilizando el algoritmo implementado, el texto de los archivos *GN*_fragmento1.txt y *GN*_fragmento2.txt. Guarden los textos cifrados para su posterior entrega como *GN*_fragmento1_cif.txt y *GN*_fragmento2_cif.txt respectivamente y súbanlos a la carpeta P1_Retos_Monoalfabeto en Teams, dentro del plazo indicado (ver nota al final).
    - Indiquen en la memoria el comando concreto que han usado para la operación de cifrado y el que se necesita para la operación de descifrado.

4) Implementen en Python el algoritmo de Vigenère.
    a) Expliquen el funcionamiento del código implementado.
        - 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) El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `vigenere.py` y pasándole los argumentos y parámetros necesarios.
        - El parámetro `--help` debe mostrar la sintaxis y las instrucciones de uso. Muestren en la memoria la salida del comando con este parámetro.
        - Guarden el código implementado para su posterior entrega en una carpeta denominada `vigenere`.
    c) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.
    d) Cifren, utilizando el algoritmo implementado el texto del archivo *GN*_fragmento3.txt. Guarden el texto cifrado como *GN*_fragmento3_cif.txt y súbanlo a la carpeta P1_Retos_Vigenere en Teams, dentro del plazo indicado (ver nota al final).
        - Indiquen en la memoria el comando concreto que han usado para la operación de cifrado y el que se necesita para la operación de descifrado.
        - La contraseña puede tener, como máximo, 7 caracteres.

Criptoanálisis

5) Seleccionen dos fragmentos de la carpeta P1_Retos_Monoalfabeto en Teams.
    a) En la memoria deben indicar el nombre de los fragmentos seleccionados, pero no es necesario que los incluyan.
    b) Traten de obtener el texto en claro correspondiente, aplicando análisis de frecuencias. Indiquen los pasos que se han seguido. Se valora la explicación y justificación del proceso. Dicho proceso no puede automatizarse con el uso de herramientas existentes. Aunque sí pueden utilizarse herramientas para los cálculos intermedios necesarios.
    c) Muestren el texto descifrado.
    d) Indiquen la clave de sustitución.
    e) Expliquen, si es posible, el funcionamiento del algoritmo de cifrado analizado.

6) Seleccionen un fragmento de la carpeta P1_Retos_Vigenere en Teams.
    a) En la memoria deben indicar el nombre del fragmento seleccionado, pero no es necesario que lo incluyan.
    b) Traten de obtener el texto en claro correspondiente, aplicando Kasiski. Indiquen los pasos que se han seguido. Se valora la explicación y justificación del proceso. Dicho proceso no puede automatizarse con el uso de herramientas existentes. Aunque sí pueden utilizarse herramientas para los cálculos intermedios necesarios.
    c) Muestren el texto descifrado.
    d) Indiquen la clave de cifrado.

7) Implementen en Python una herramienta que automatice lo máximo posible el criptoanálisis usando Kasiski.
   a) Expliquen el funcionamiento del código implementado.
   b) El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `kasiski.py` y pasándole los argumentos y parámetros necesarios.
      - El parámetro `--help` debe mostrar la sintaxis y las instrucciones de uso. Muestren en la memoria la salida del comando con este parámetro.
      - Guarden el código implementado para su posterior entrega en una carpeta denominada `kasiski`.
   c) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.

Criptografía moderna. Cifrado simétrico

8) Implementen en Python el algoritmo RC4, con las siguientes consideraciones:
   - Al ejecutar el algoritmo, debe mostrarse 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 de S debe hacerse en formato decimal.
   - Para el cifrado:
     - La clave de cifrado debe introducirse en formato hexadecimal.
     - El texto a cifrar debe leerse por consola e irse cifrando, carácter a carácter, con cada pulsación del teclado.
     - Los caracteres introducidos se interpretarán como ASCII.
     - Para cada carácter introducido, se mostrará: 1) su codificación en ASCII y en binario, 2) el valor del *keystream* en decimal y en binario y 3) el resultado de la operación de cifrado en binario y en hexadecimal.
     - Finalizado el proceso de cifrado, se mostrará el resultado completo del texto cifrado, en formato hexadecimal.
   - Para el descifrado:
     - El texto a descifrar se introducirá por consola en formato hexadecimal. Todo el texto de una vez.
     - En este caso no se mostrarán los pasos intermedios, como en el cifrado. Sino que se mostrará directamente el texto descifrado, en formato ASCII.

   a) El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `RC4.py` y pasándole los argumentos y parámetros necesarios.
      a. El parámetro `--help` debe mostrar la sintaxis y las instrucciones de uso. Muestren en la memoria la salida del comando con este parámetro.
      b. Guarden el código implementado para su posterior entrega en una carpeta denominada `rc4`.
   b) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.

9) Revisen las animaciones de DES y AES en cryptool, con el objetivo de comprender su funcionamiento general. Con respecto a la arquitectura, ¿cuál es la diferencia principal entre ambos algoritmos?

10) Utilizando OpenSSL:
    a) Indiquen los comandos necesarios para cifrar el archivo `secreto.txt` con el algoritmo DES utilizando una clave de 56 bits y con el algoritmo AES, utilizando una clave de 256 bits, sin hacer uso de un modo de operación específico. Indique el tiempo que tarda en ejecutarse cada comando.
    b) Repita ahora el apartado a), pero usando los modos ECB y CBC para ambos algoritmos. Analice y comente los resultados.
    c) ¿Es posible usar otros modos de operación con OpenSSL?
    d) ¿Cómo son los tiempos de cifrado y descifrado de DES y AES para el mismo texto y modo de operación?
    e) Busquen y comenten sobre alguna vulnerabilidad sobre AES

11) Utilizando el paquete PyCryptodome, implementen en Python un algoritmo que simule el funcionamiento de OpenSSL para los casos vistos en el ejecicio 10
    a) El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `cipher.py` y pasándole los argumentos y parámetros necesarios.
        - El parámetro `--help` debe mostrar la sintaxis y las instrucciones de uso. Muestren en la memoria la salida del comando con este parámetro.
        - Guarden el código implementado para su posterior entrega en una carpeta denominada `cipher`.

Modo y fecha de entrega

Los archivos cifrados correspondientes a los ejercicios 3 y 4 deben subirse a las carpetas de Teams correspondientes, no más tarde del **lunes 7 de octubre**.

La memoria y resto de archivos, se entregarán vía Campus Virtual, no más tarde del **lunes 21 de octubre a las 10:00**. Sólo una entrega por grupo. Un único archivo *GN*`_P1.zip`, que debe contener:

- Memoria, en formato .pdf:
    - No agrupen ni cambien el orden de ejercicios ni apartados. Mantengan la numeración de los ejercicios y apartados y respondan con claridad a lo que se pide en cada uno de ellos.
    - Aporten evidencias suficientes del trabajo realizado (capturas de pantalla, comandos concretos utilizados, explicaciones, etc.).
- Archivos indicados en el enunciado de la práctica, respetando la nomenclatura de archivos y carpetas indicada.

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, cada uno con su ordenador y todo lo necesario para poder mostrar cualquier ejercicio de la práctica.