Universidade de A Coruña
Facultade de Informática

Práctica CSAI: Cifrado.
Curso 2023/2024

Calidade Seguridade e Auditoría Informática

Diego Antonio López López diego.a.lopez@udc.es
Alejandro Rodríguez Vaquero a.vaquero@udc.es

Índice General

1. Equipo.

1.1. Nombre del Equipo

- DiegoAleja

1.2. Integrantes

- Integrante 1:
  - Nombre completo Diego Antonio López López
- Login : diego.a.lopez@udc.es
- Integrante 12:
  - Alejandro Rodriguéz Vaquero
  - Login : a.vaquero@udc.es

1.3. Input

Formatos de input que toma el fichero DiegoAleja.py

- python DiegoAleja.py <nombre_del_archivo>

Para hacer funcionar el archivo simplemente se le debe llamar, pasándole como argumento el nombre del archivo (el cual debe estar en el mismo directorio) este abrirá el archivo formateara el texto y generará una posible clave .

2. Autoría

2.1 Autor:

Título repositorio: Vigenere-Cipher-Breaker
Autor: Andrew Paul

2.2 Link al repositorio:

https://github.com/1r0dm480/Vigenere-Cipher-Breaker/tree/master

3. Funcionamiento en detalle.

La implementación del algoritmo vigenere se basó en el siguiente repositorio de github:

3.1. Lectura del Archivo de Entrada

- El script espera un argumento de línea de comandos que especifique el nombre del archivo de entrada.
- Lee el contenido del archivo de entrada usando la función `leer_archivo`, la cual retorna el texto.

3.2. Preprocesamiento

- El script elimina los espacios y saltos de línea del texto utilizando `texto_sin_espacios = texto_original.replace(" ", "").replace("\n", "")`. Este paso asegura que solo se consideren caracteres alfabéticos para el análisis.

3.3. Determinación de la Longitud de la Clave

- El script llama a la función `get_key_length` para determinar la longitud probable de la clave.
- Esta función itera a través de diferentes suposiciones de longitud de clave, calculando el Índice de Coincidencia (IC) promedio para cada suposición. La longitud de clave con el IC más alto se considera la más probable.

3.4. Generación de la Clave

- Una vez determinada la longitud probable de la clave, el script llama a la función `get_key` para generar la clave.
- En `get_key`, se realiza un análisis de frecuencia en cada segmento del texto cifrado para determinar qué letra es más probable para esa parte de la clave.

3.5. Ejecución de la Función Principal

- La función `main` encapsula todo el proceso.
- Toma el texto preprocesado (le realiza otro procesado), determina la longitud probable de la clave y genera la clave utilizando el análisis de frecuencia.

3.6. Salida

- La clave generada se imprime en la consola.

Flujo General

1. Leer el archivo de entrada.
2. Preprocesar el texto (eliminar espacios y saltos de línea).
3. Determinar la longitud probable de la clave.
4. Generar la clave utilizando el análisis de frecuencia.
5. Imprimir la clave generada.

3.7 Cambios al algoritmo

- Se eliminó toda la interacción del usuario
- Se cambió la frecuencia de letras a la del español
- Se añadió una pequeña parte de procesado de texto
- Se añadió la funcionalidad de leer el texto desde la máquina
- Adaptación al inglés (se adaptó el algoritmo mediante sustituciones para que funcione con idiomas como el español o el francés)