

Nombre: Edlith Alejandra Vinueza Zambrano

Ejercicio 1

Indicaciones:

Realice el cifrado de un mensaje por permutación de filas, teniendo como clave 5 filas y la cantidad de columnas que sean necesarias (garantice al menos 3). Los espacios del mensaje original se sustituyen con el carácter "-". Si en la matriz de cifrado sobran espacios, estos deben llenarse con el carácter "*".

Lenguaje empleado: Python

Cifrado por Permutación de Filas

Creamos un método llamado: `cifrado_por_permutacion` el recibe 5 filas como clave

```
def cifrar_por_permutacion(mensaje, filas=5):
```

1. Sustitución de Espacios:

- En primer lugar, reemplazamos todos los espacios en blanco del mensaje por el carácter "-".

```
# 1) Reemplazar los espacios con '-'
mensaje = mensaje.replace(' ', '-')

```

2. Cálculo de las Columnas:

- Calculamos cuántas columnas necesitamos para la matriz. Si el número de caracteres no es un múltiplo exacto de 5, rellenamos el mensaje con el carácter * para completar la matriz.

```
# 2) Cálculo de cuántas columnas se necesitan
# Al menos 3 columnas, y el número total de celdas debe ser múltiplo de 5
columnas = max(3, (len(mensaje) + filas - 1) // filas) # Número de columnas necesarias

# Rellenar con '*' si el mensaje no llena la matriz
total_caracteres = filas * columnas
mensaje = mensaje.ljust(total_caracteres, '*')

```

3. Creación de la Matriz:

- Una vez que sabemos cuántas columnas necesitamos, colocamos el mensaje en una matriz de 5 filas.

```
# 3) Construir la matriz de cifrado
matriz = [list(mensaje[i:i+columnas]) for i in range(0, len(mensaje), columnas)]

# Mostrar la matriz antes de la permutación (opcional)
print("Matriz antes de permutar:")
for fila in matriz:
    print("".join(fila))

```

4. Permutación de las Filas:

- Cambiamos el orden de las filas en la matriz de manera aleatoria.

```
# 4): Permutar las filas de manera aleatoria
filas_permutadas = list(range(filas))
random.shuffle(filas_permutadas)

# Crear la matriz permutada
matriz_permutada = [matriz[i] for i in filas_permutadas]

# Mostrar la matriz permutada (opcional)
print("\nMatriz después de permutar:")
for fila in matriz_permutada:
    print("".join(fila))
```

5. Lectura por Columnas:

- Finalmente, para obtener el mensaje cifrado, leemos la matriz columna por columna. Es decir, tomamos el primer carácter de cada fila, luego el segundo de cada fila, y así sucesivamente, hasta completar todo el mensaje cifrado.

```
# 5): Leer la matriz por columnas
mensaje_cifrado = ''.join(matriz_permutada[fila][col] for col in range(columnas) for fila in range(filas))
return mensaje_cifrado
```

Imprimir resultados en consola:

```
print("RESULTADOS")

mensaje_original = "eavinuezaz1"
print("Mensaje original: " + mensaje_original)
mensaje_cifrado = cifrar_por_permutacion(mensaje_original)
print("\nMensaje cifrado:", mensaje_cifrado)
```

Resultados

Mensaje original

```
C:\Users\fing.labcom\Desktop\eavinuezaz1>C:/Users/fing.labcom/AppData/Local/Microso
RESULTADOS
Mensaje original: eavinuezaz1
```

Matriz (antes y después)

```
Matriz antes de permutar:  
eav  
inu  
eza  
z1*  
***
```

```
Matriz después de permutar:  
z1*  
eza  
inu  
eav  
***
```

Mensaje cifrado

```
Mensaje cifrado: zeie*1zna**auv*
```

```
C:\Users\fing.labcom\Desktop\eavinuezaz1>
```