

**UNIVERSIDAD DEL VALLE DE GUATEMALA**

Cifrado de información

Sección 10

Ludwing Cano



## **Laboratorio 2**

Parte B

Abner Iván García Alegría 21285

## Requerimiento:

1. Implementar una función que haga la operación XOR, bit a bit, con dos cadenas de texto.

```
clave = b'si'

resultado = encriptar_con_xor(texto, clave)

Proceso de encriptación XOR:
-----
Mensaje original: b'hola'
Keystream original: b'si'
Keystream ajustado: b'sisi'

Operación XOR byte por byte:

Posición 0:
Byte mensaje: 01101000 (h)
Byte key: 01110011 (s)
Resultado: 00011011 (0x1b)

Posición 1:
Byte mensaje: 01101111 (o)
Byte key: 01101001 (i)
Resultado: 00000110 (0x6)

Posición 2:
Byte mensaje: 01101100 (l)
```

## Problemas a resolver

2. Dada la imagen XOR\_Imagen, y la llave “cifrados\_2025” encontrar el valor original de la

imagen.

a. Deben de convertir la imagen a base64 y aplicarle un xor con la llave para encontrar su valor

```
imagen = "./imagen_xor.png" # Imagen a cifrar

clave = b'cifrados_2025' # Clave para cifrar la imagen

with open(imagen, "rb") as f: # Leer imagen
    imagen_bytes = f.read() # Convertir imagen a bytes

imagen_resultado = encriptar_con_xor(imagen_bytes, clave) # Cifrar imagen

with open("./imagen_xor_cifrada.png", "wb") as f: # Guardar imagen cifrada
    f.write(imagen_resultado) # Guardar imagen cifrada
    print("\nImagen cifrada guardada en /imagen_xor_cifrada.png") # Mensaje de confirmación
```



### 3. Investigar porque al aplicar XOR con una llave de texto la imagen se corrompe.

R// Al aplicar XOR a una imagen con una llave de texto, lo que hace es modificar el valor de cada píxel de la imagen basándose en el valor de la llave. Es como si cada color de la imagen original se mezclara con un color "aleatorio" de la llave, resultando en colores completamente nuevos y sin relación aparente con la imagen original. Por ejemplo, imaginemos que estamos pintando un retrato con una paleta de colores que cambia constantemente y sin control; el resultado final sería una imagen irreconocible y "corrompida".

Este texto fue sacado en base a la referencia de

Khan Academy. (s. f.).  
<https://es.khanacademy.org/computing/computer-science/cryptography/ciphers/a/xor-bitwise-operation>.

**4. Investigar cómo aplicar un xor a 2 imágenes. Para esto deben de seleccionar 2 imágenes, luego proceder hacer un xor entre las dos imágenes. Esto significa que una imagen es la original y la otra se utilizará como llave para aplicar el xor.**

**a. Mostrar las imágenes utilizadas y el resultado, así mismo explique qué inconvenientes encontró al momento de realizar el xor.**

“Recordemos que cuando aplicamos la operación XOR a una imagen, estamos realizando una manipulación de bits en cada píxel de la imagen. La operación XOR combina dos conjuntos de bits en una posición correspondiente y produce un nuevo conjunto de bits como resultado. La peculiaridad del XOR es que devuelve 1 en una posición si los bits son diferentes y 0 si son iguales.”

Tips:

- Pueden utilizar la librería pillow (Images)  
o resize : tener el mismo tamaño de pixeles  
o fromarray: toma un array y crea una imagen
- Pueden utilizar numpy  
o bitwise\_xor: aplica un xor a nivel de bits de las imágenes

Imagen 1



Imagen 2



Imagen XOR imagen1 XOR imagen2



## Mis complicaciones al aplicar XOR

R// Al realizar la operación XOR entre dos imágenes, me encontré con varios inconvenientes.

Primero, las imágenes deben tener las mismas dimensiones. Si las imágenes tienen tamaños diferentes, no se puede realizar la operación XOR directamente, ya que no hay una correspondencia uno a uno entre los píxeles lo cual me toco que convertirlo a las mismas dimensiones cada imagen.

Otro problema surgió con el tipo de datos de las imágenes. La operación XOR se realiza bit a bit, por lo que las imágenes deben estar en un formato que lo permita y deben ser el mismo formato, hice pruebas con diferentes formatos y tendía a no generar la imagen XOR por lo cual deben estar en el mismo formato.

Además, la operación XOR es sensible al ruido. Si las imágenes contienen ruido, este se verá amplificado en la imagen resultante, lo que puede dificultar la visualización de la información original.