

Universidad Intercanional del
Ecuador

Christian Salinas



The screenshot shows a terminal window with the following content:

```
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ sudo python Cipher.py
[sudo] password for kali:
Message
Key
b'0{\xb3\xa7\xea{.\x90\xf4\x15\xbb'
Encryption with XOR
BÆÊ[bz
Decryption with XOR
oola mundo

└─(kali㉿kali)-[~]
$
```

The terminal shows the execution of a Python script named `Cipher.py`. It prompts for a password, which is used to generate a key. The key is displayed as a byte string: `b'0{\xb3\xa7\xea{.\x90\xf4\x15\xbb'`. The script then performs XOR encryption on the message "BÆÊ[bz" using this key, resulting in the ciphertext "oola mundo". Finally, it performs XOR decryption on the same ciphertext using the same key, recovering the original message.

Tenemos al texto `\x14BÆÊ\x16[b\x90z` con caracteres especiales en la codificación latin-1

Tambien aplicamos el proceso XOR que se usó para cifrar, ya que el cifrado XOR es reversible (es decir, $a \wedge b \wedge b = a$).

Y tendremos en la terminar el resultado mas parecido o el mensaje en si.

```
GNU nano 8.3                                         Ciph...
```

```
import random

def encode_system(text, key):
    encrypted_text = bytes([a ^ b for a, b in zip(key, list(text.encode(encoding='latin-1')))])
    return encrypted_text.decode(encoding='latin-1')

def decode_system(text, key):
    encrypted_bytes = text.encode('latin-1')
    decrypted = bytes([a ^ b for a, b in zip(key, encrypted_bytes)])
    return decrypted.decode('latin-1')

if __name__ == '__main__':
    print("Message")
    print()
    print("Key")
    key = bytes([79, 123, 179, 167, 234, 123, 46, 144, 244, 21, 187])
    print(key)
    print()
    print("Encryption with XOR")
    encrypted_text = "\x14\x8E\x16[p\x90z"
    print(encrypted_text)
    print()
    print("Decryption with XOR")
    decrypted_text = decode_system(encrypted_text, key)
    print(decrypted_text)
    print()
```