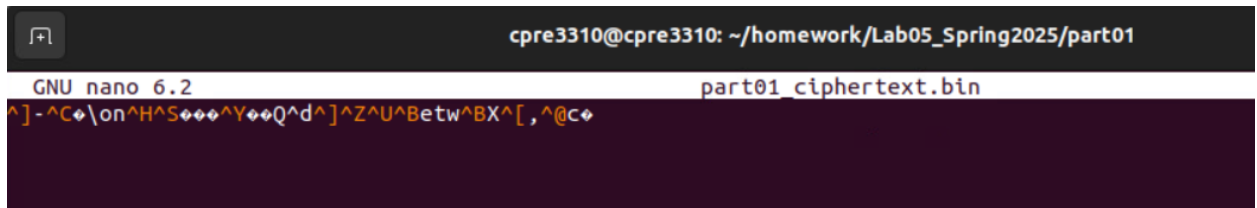


Lab 05 Template

Part 01

1) Open the part01_ciphertext.bin file and include a screenshot of the result in your lab report.

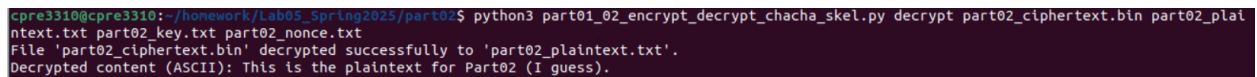
(15 points)



Part 02

2) Include a screenshot in your lab report of the resulting ascii text printed on your screen when you decrypted from part02_ciphertext.bin.

(15 points)



3) Upload your working part01_02_encrypt_decrypt_chacha_skel.py code as a separate file.

(20 points)

uploaded.

Part 03

4) Include a screenshot in your lab report of the resulting ascii text printed on your screen which is part03_plaintext2.txt printed on your screen.

(15 points)

```
cpre3310@cpre3310:~/homework/Lab05_Spring2025/part03$ python3 part03_xor_nonce_reuse_skel.py part03_ciphertext1.bin part03_ciphertext2.bin part03_plaintext1.txt part03_plaintext2.txt
Recovered Plaintext 2 (ASCII): Oh no! Can you see me?
Recovered plaintext written to 'part03_plaintext2.txt'.
```

5) Questions related to reusing nonce, but not the key

a. **Would you generate a unique keystream if the nonce was reused, but the key changed?**

(5 points)

Yes, we would generate a unique keystream if nonce was reused but the key changed. Changing either one will result in a different keystream. The keystream in ChaCha20 is derived from both the key and nonce.

b. **Would you really want to implement ChaCha20 reusing the nonce? Why or why not?**

(10 points)

We would not want to implement ChaCha20 reusing nonce. It could lead to confidentiality or security attacks. An attacker could potentially decrypt sensitive information. Using a unique nonce is the best practice as it reduces risk of accidental key reuse.

6) **Upload your working part03_xor_nonce_reuse_skel.py code as a separate file.**

(20 points)

Uploaded.