Lab 06 Template

Part 01

1. **Upload python code for babyFeistel cipher**

(10 points)

Uploaded.

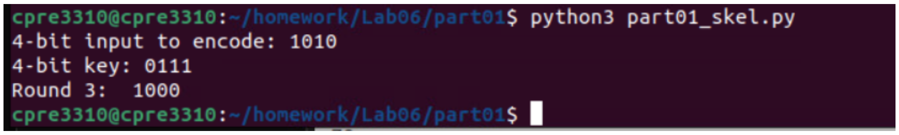
2. **Final 4-bit ciphertext**

(5 points)

1000 is the final 4 bit ciphertext

3. **Screenshot of encrypting**

(5 points)

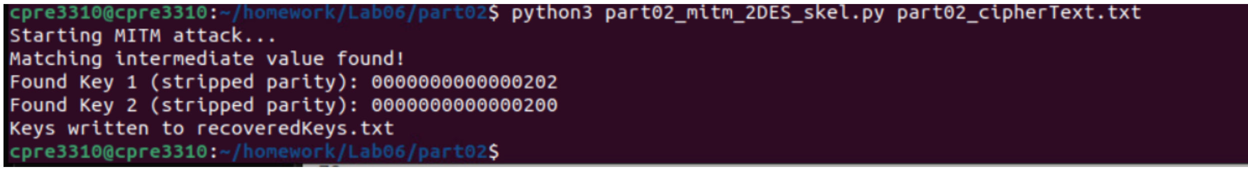


```
cpre3310@cpre3310:~/homework/Lab06/part01$ python3 part01_skel.py
4-bit input to encode: 1010
4-bit key: 0111
Round 3:  1000
cpre3310@cpre3310:~/homework/Lab06/part01$
```

Part 02

4. **Include a screenshot of the two keys in your lab report.**

(10 points total, 5 points each key)



```
cpre3310@cpre3310:~/homework/Lab06/part02$ python3 part02_mitm_2DES_skel.py part02_cipherText.txt
Starting MITM attack...
Matching intermediate value found!
Found Key 1 (stripped parity): 0000000000000202
Found Key 2 (stripped parity): 0000000000000200
Keys written to recoveredKeys.txt
cpre3310@cpre3310:~/homework/Lab06/part02$
```

5. **What block cipher mode is being used? What is a known flaw with this mode?**

(10 points total, 5 points each question)

The block cipher mode being used is ECB(Electronic Codebook). Known flaw would be that it does not provide semantic security. Identical plaintext blocks always encrypt the same ciphertext blocks and can lead to pattern leakage. This makes it vulnerable to pattern leakage to attackers.

6. **Why is known plaintext important in a meet-in-the-middle attack? What role does it play in recovering the encryption keys?**

(10 points total, 5 points each question)

It serves as a point of comparison between the results of encryption and decryption at the intermediate stage. The attacker encrypts the known plaintext with all possible keys for the first encryption stage and stores it. At the same time the attacker decrypts with all possible keys and stores it. By finding matching values in the stored results the attacker can find the keys. Without known plaintext, the attacker would have no way to correlate intermediate values, making the attack infeasible.

7. Could this attack be extended to 2-key 3DES? What about 3-key 3DES? What differences or additional challenges would arise in attacking 3DES with a similar approach?

(10 points total, 5 points each question)

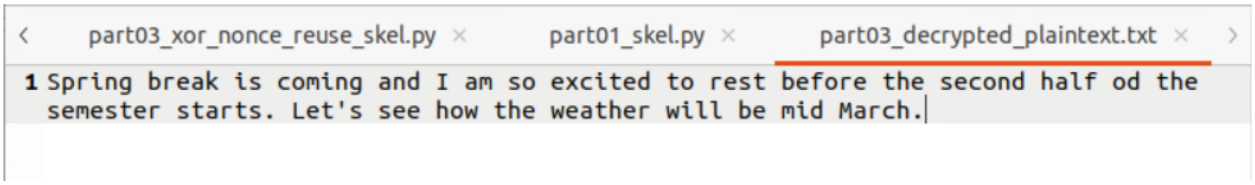
This could possibly be extended to a 2-key 3DES but it will be more complex. The number of possible intermediate values increases and the storage will increase too. This would not be efficient to extend it to 3-key 3DES as the storage would be significantly higher and there would be two separate intermediate results to match. The total number of possible keys would be 2^168.

The challenges would be the that there would be two separate intermediate results to match, storage increases, and possible keys increases significantly.

Part03

8. **Include a screenshot of the plaintext result in your lab report.**

(5 points)


part03_xor_nonce_reuse_skel.py
part01_skel.py
part03_decrypted_plaintext.txt

```
Spring break is coming and I am so excited to rest before the second half od the
semester starts. Let's see how the weather will be mid March.
```

9. **What block cipher mode are you using in this implementation of Triple DES?**

(5 points)

We use CBC(Cipher Block Chaining). It provides better security than ECB.

10. **In 2-key 3DES, what is the *effective key length* for encryption? How does it compare to the effective key length of 3-key 3DES? What are those other bits used for?**

(15 points total, 5 points each question)

2-key 3DES - Effective length is 112 bits as each key is 56 bits.

3-key 3DES - Effective key length is 168 bits as it uses 3 56 bit keys.

The remaining bits are used for key parity checks in both the cases.

11. Historically, there has been a way to use a single key in Triple DES where you set key1=key2=key3. Why would you want to do that? What is the effective key size when key1=key2=key3?

(15 points total, 5 points each question)

This reduces 3DES to single DES. This reduces the effective key size to 56 bits. This would reduce the security but we would use this for backward compatibility with older DES systems. If a system uses a single DES and we are transforming to a 3DES system then we would want to use this. This would ensure backward compatibility with older systems.