Course：Network Security

Homework 1

Name：0653422 林容伊

Explanation：

1. 首先，我們每個人都有 flag.enc，先把他讀進來之後用 base64 解密。

```
with open('./flag.enc','rb') as f:
    flag = f.read().strip() #type:bytes
    flag = base64.b64decode(flag)
```

2. 然後我們每個人身上都會有一把 public key 先從公鑰裡面求得 n 和 e

```
# Get public key
def getpubkey():
    with open('./pub.pem','rb') as f:
        pub = f.read()
        public_key = RSA.importKey(pub)

        n = public_key.n  #140816102882370072753963128960517081965880
        e = public_key.e  #65537

    return public_key,n,e
```

3. 再接著使用 chosen cyphertext attack 的步驟一步步算出加密後的 Y

- Attack steps:
  - choose X where X is relatively prime to n
  - create Y = $C*X^e$ mod n
  - get Z = decrypted Y
  - $Z = Y^d = (C*X^e)^d = C^d*X^{ed} = C^d*X = P^{ed}*X = P*X$ mod n
  - find out X^{-1}, the modular inverse of X
  - $P = Z*X^{-1}$ mod n

```
#str_flag = str(flag, encoding = 'utf-8') # bytes to str

# Use binascii.hexlify to transfer byte string into integer
#flag_enc = int(binascii.hexlify(bytes(flag,'utf-8')),16)
flag_enc = int(binascii.hexlify(flag),16)

#choose X where X is relatively prime to n
X = 997

Y = flag_enc * (X ** e) % n

# use binascii.unhexlify to transfer integer into byte string
Y = binascii.unhexlify(hex(Y)[2:])

Y = base64.b64encode(Y)

Y = str(Y)[2:-1]+'\n'
```

4. 算好加密後的 Y 之後再 send 進 server 去解密

```
HOST = '140.113.194.66'
PORT = 8888
# create socket
# AF_INET 代表使用標準 IPv4 位址或主機名稱
# SOCK_STREAM 代表這會是一個 TCP client
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# client 建立連線
sock.connect((HOST, PORT))

# 接收資料
response_enter = sock.recv(4096)   #接收4096字元
print(response_enter.decode("utf-8"))

# send encrypted message Y to server to decrypted
sock.send(bytes(Y,"utf-8"))  #傳送Y出去
response_receive = sock.recv(4096)
print(response_receive.decode("utf-8"))
response_receive = sock.recv(4096)
print(response_receive.decode("utf-8"))
response_receive = response_receive.strip()
```

5. 解密後的 message 再去算 Z 和 P(照著上面 attack 的步驟)解出最後真正的 flag

```
response_receive = base64.b64decode(response_receive)

# Use binascii.hexlify to transfer byte string into integer
Z = int(binascii.hexlify(response_receive), 16)

X_Inverse = modInv(X , n)
P = (Z * X_Inverse) % n
final_flag = binascii.unhexlify(hex(P)[2:])
#print(str(final_flag, encoding = 'utf-8'))
print(bytes.decode(final_flag))
```

6. 最後 FLAG{S0_y0u_d0_know_th3_cho5en_c1ph3r_4ttack!}