CS1702: Network Security

Assignment -1: RSA Implementation

Submitted by: Siddharth Koppisetty

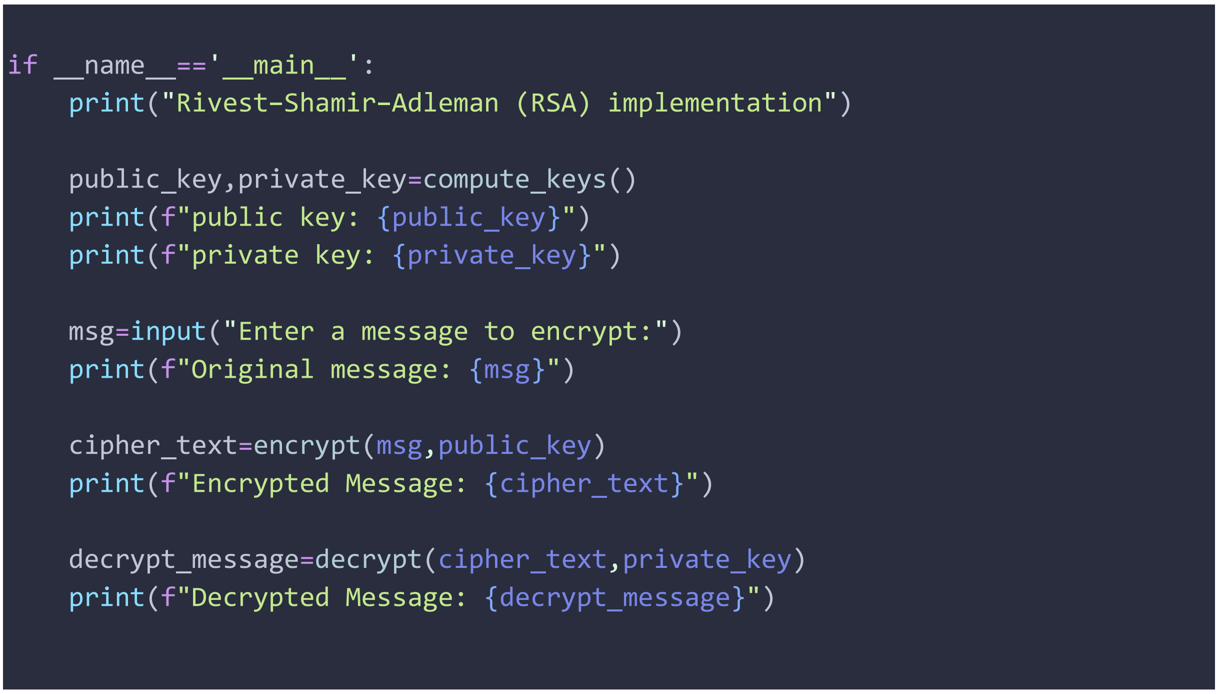
Roll no: CS22B1022

Introduction:

RSA (Rivest–Shamir–Adleman) is a public-key cryptographic algorithm that forms the backbone of many secure communication protocols. It relies on the mathematical hardness of factoring the product of two large prime numbers, a problem for which no efficient solution is known. RSA uses a key pair: a **public key** for encryption and a **private key** for decryption. The keys are generated using modular arithmetic and Euler's totient function. RSA is widely used in secure data transmission, digital signatures, and key exchange mechanisms. Its asymmetric nature allows it to provide both confidentiality and authentication, making it a critical component of modern network security infrastructures.

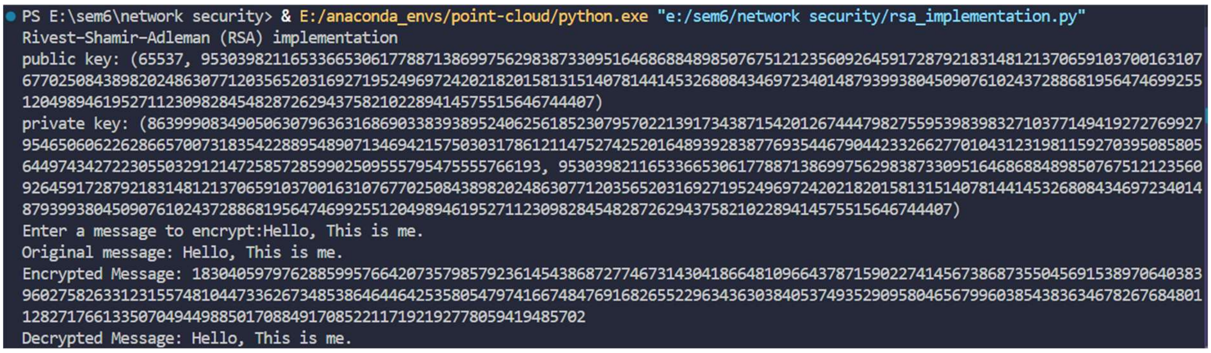
Code (rsa-implementation.py):

```
import random
import sympy

def generate_prime(bits=512):
    return sympy.randprime(2**(bits-1), 2**bits)

def compute_keys():
    p = generate_prime()
    q = generate_prime()

    n = p * q
    phi_n = (p - 1) * (q - 1)

    e = 65537  #public exponent
    d=pow(e,-1,phi_n) #private exponent

    return ((e, n), (d, n))

def encrypt(message, public_key):
    e, n = public_key
    message_int=int.from_bytes(message.encode(), 'big')
    cipher_text=pow(message_int,e,n)
    return cipher_text

def decrypt(cipher_text, private_key):
    d, n = private_key
    decrypted_int=pow(cipher_text,d,n)
    decrypted_message=decrypted_int.to_bytes((decrypted_int.bit_length()+7)//8
, 'big').decode()
    return decrypted_message
```



```python
if __name__=='__main__':
    print("Rivest-Shamir-Adleman (RSA) implementation")

    public_key,private_key=compute_keys()
    print(f"public key: {public_key}")
    print(f"private key: {private_key}")

    msg=input("Enter a message to encrypt:")
    print(f"Original message: {msg}")

    cipher_text=encrypt(msg,public_key)
    print(f"Encrypted Message: {cipher_text}")

    decrypt_message=decrypt(cipher_text,private_key)
    print(f"Decrypted Message: {decrypt_message}")
```

Output:



```
PS E:\sem6\network security> & E:/anaconda_envs/point-cloud/python.exe "e:/sem6/network security/rsa_implementation.py"
Rivest-Shamir-Adleman (RSA) implementation
public key: (65537, 9530398211653366530617788713869975629838733095164686884898507675121235609264591728792183148121370659103700163107
677025084389820248630771203565203169271952496972420218201581315140781441453268084346972340148793993804509076102437288681956474699255
1204989461952711230982845482872629437582102289414575515646744407)
private key: (8639990834905063079636316869033839389524062561852307957022139173438715420126744479827559539839832710377149419272769927
954650606226286657007318354228895489071346942157503031786121147527425201648939283877693544679044233266277010431231981159270395085805
6449743427223055032912147258572859902509555795475555766193, 953039821165336653061778871386997562983873309516468688489850767512123560
926459172879218314812137065910370016310767702508438982024863077120356520316927195249697242021820158131514078144145326808434697234014
8793993804509076102437288681956474699255120498946195271123098284548287262943758210228941457551564674407)
Enter a message to encrypt:Hello, This is me.
Original message: Hello, This is me.
Encrypted Message: 18304059797628859957664207357985792361454386872774673143041866481096643787159022741456738687355045691538970640383
960275826331231557481044733626734853864644642535805479741667484769168265522963436303840537493529095804656799603854383634678267684801
128271766133507049449885017088491708522117192192778059419485702
Decrypted Message: Hello, This is me.
```