EE8209: INFORMATION SECURITY

ASSIGNMENT 02

NAME	:	MOYURA M.L.
REG NO.	:	EG/2018/3397
SEMESTER	:	08
DATE	:	17/08/2023

Problem

Implement RSA from scratch for your assignment 2. You can use any programming language. Similar to assignment 1, you need to encrypt your name and decrypt it back. All the intermediate states should be printed. The code should be properly commented. Note that you need to select two large prime numbers for p and q.

Introduction

The Massachusetts Institute of Technology's (MIT) Ron Rivest, Adi Shamir, and Leonard Adleman first presented the RSA asymmetric public key encryption method in 1977. One of the oldest, most used, and safest public key cryptography algorithms in existence. The last names of the three creators provide the basis for the abbreviation RSA. A public key is used for message encryption and a private key is used for message decryption in the RSA encryption process [1].

The RSA algorithm's key generation steps are shown below figure,

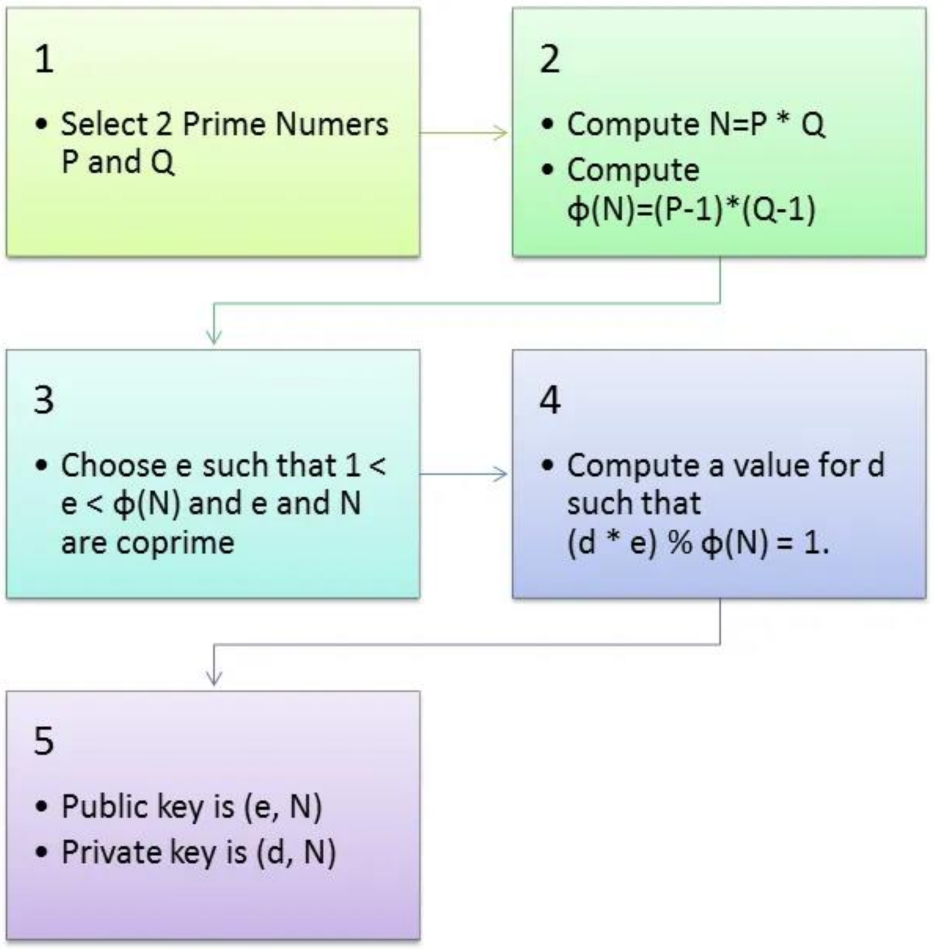

1
Select 2 Prime Numers P and Q
2
Compute N=P * Q
Compute ϕ(N)=(P-1)*(Q-1)
3
Choose e such that 1 < e < ϕ(N) and e and N are coprime
4
Compute a value for d such that (d * e) % ϕ(N) = 1.
5
Public key is (e, N)
Private key is (d, N)

Figure 1: Steps of generating Public and Private Keys [2]

According to Figure 1,

- The RSA algorithm works by using the properties of prime numbers. A prime number is a number that is only divisible by itself and 1. The RSA algorithm first selects two prime numbers, p and q. It then calculates their product and N.

- The algorithm then calculates the value of $\Phi(N) = (p-1)(q-1)$, which is the number of positive integers less than or equal to N that are relatively prime to N. A number is relatively prime to N if it has no common factors with N other than 1.

- The algorithm then chooses an integer e such that $1 < e < \Phi(N)$ and e and N are coprime. This means that e and N have no common factors other than 1. e is the public exponent (public key). The algorithm then calculates the private exponent d such that $d * e = 1 \bmod \Phi(N)$. d is the private key.

- To encrypt a message, the sender uses the public key to encrypt the message. This is done by raising the message to the power of e mod N and to decrypt the message, the recipient uses the private key to decrypt the message as shown in Figure 2. This is done by raising the encrypted message to the power of d mod N.

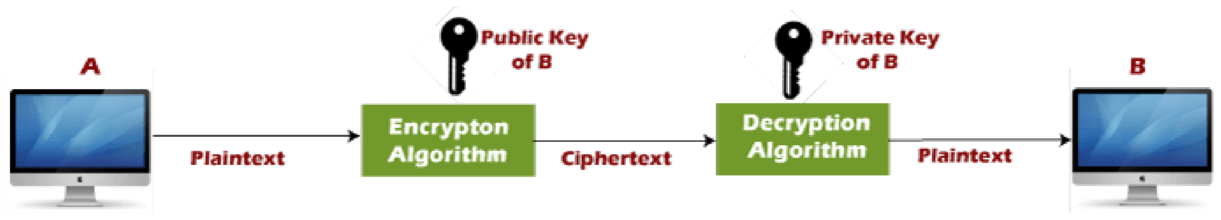


Figure 2: Use of Public and Private key in RSA algorithm [3]

The strength of the RSA method, which offers security for data transfer and digital signatures, is a result of the difficulty in factoring huge numbers. But it performs more slowly and might be vulnerable to future quantum computers because it depends on lengthy key lengths. Its shortcomings in perfect forward secrecy and padding vulnerabilities make key management a necessity. Due to higher security and shorter keys, modern options like elliptic curve cryptography are gaining popularity [1].

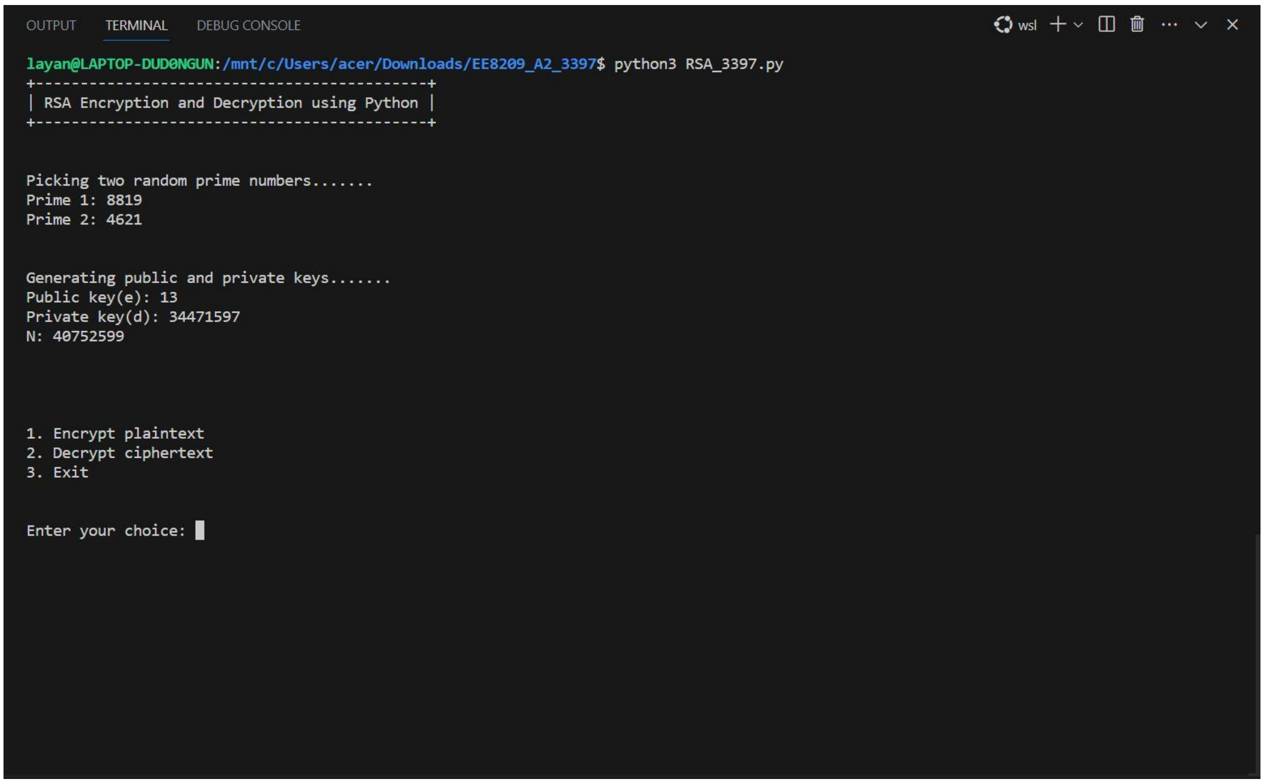
Implementation

RSA algorithm was implemented using Python language referring to methods provided in [1]. First, functions are defined to establish public and private keys, check for primes, and choose random primes. Two 4-digit prime numbers are randomly chosen for each run. The primary function presents the user with a menu option to either encrypt plaintext using the freshly created public key or decrypt ciphertext using the associated private key. The program encodes and decodes the data after converting it from plaintext to ASCII, then uses the recently created public key to encrypt and encode each character. For decryption, it performs the procedure backwards by deciphering the ciphertext, decrypting each character with its associated private key, and producing the decrypted output. The efficient encryption and decryption in the code is achieved through modular exponentiation.

In summary, The user interacts with the program through the terminal menu. The application provides a straightforward user interface to demonstrate the RSA process with various prime numbers and step-by-step encryption and decryption procedures.

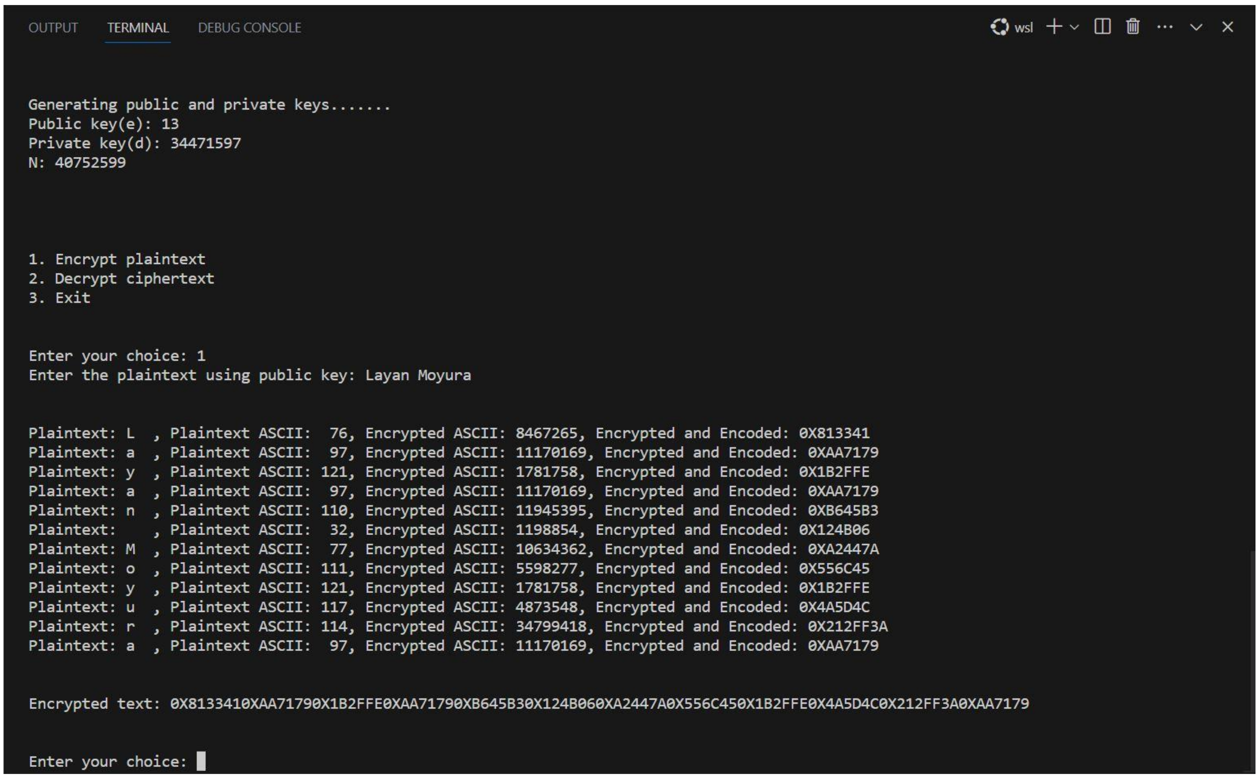
Results

1. Initiation of public and private keys.



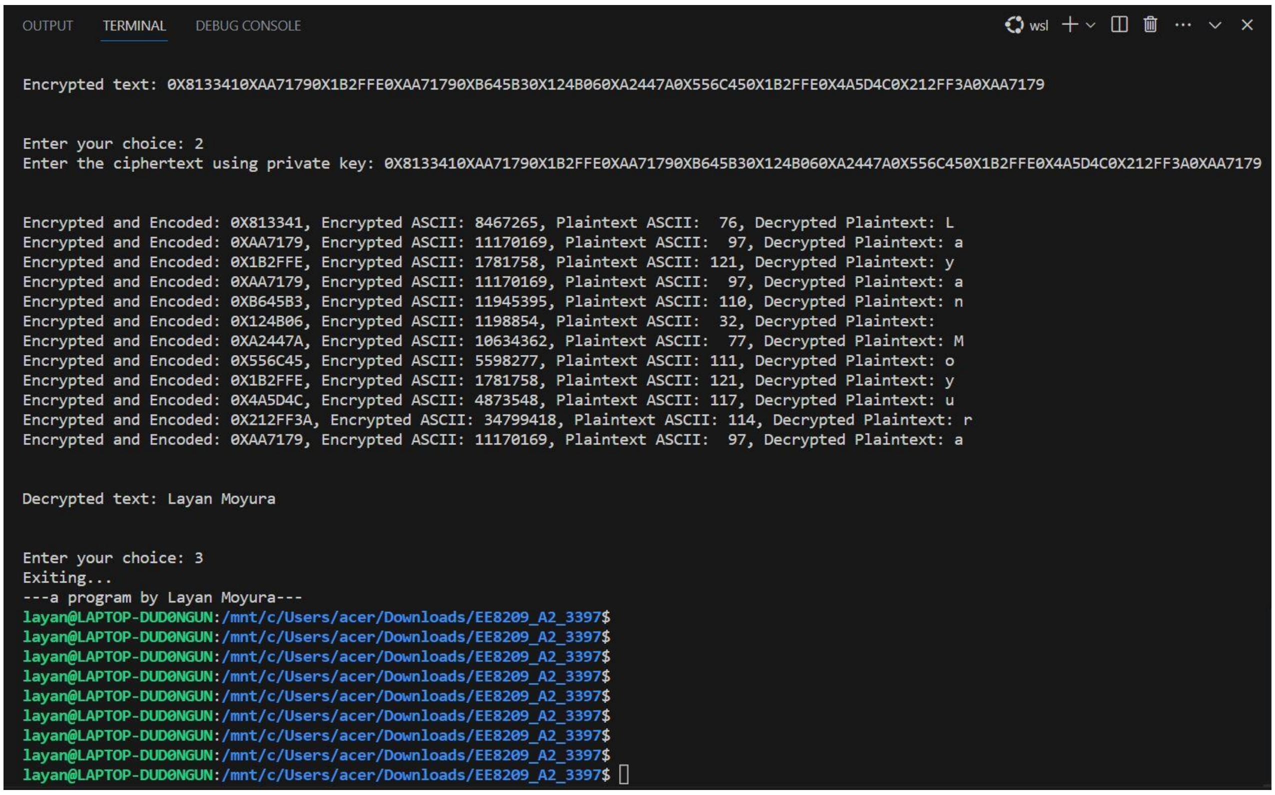
```
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$ python3 RSA_3397.py
+-------------------------------------------+
| RSA Encryption and Decryption using Python |
+-------------------------------------------+


Picking two random prime numbers.......
Prime 1: 8819
Prime 2: 4621


Generating public and private keys.......
Public key(e): 13
Private key(d): 34471597
N: 40752599




1. Encrypt plaintext
2. Decrypt ciphertext
3. Exit


Enter your choice: 
```

2. User encrypting plaintext(name) after choosing option 1.



```
Generating public and private keys.......
Public key(e): 13
Private key(d): 34471597
N: 40752599



1. Encrypt plaintext
2. Decrypt ciphertext
3. Exit


Enter your choice: 1
Enter the plaintext using public key: Layan Moyura


Plaintext: L  , Plaintext ASCII:  76, Encrypted ASCII: 8467265, Encrypted and Encoded: 0X813341
Plaintext: a  , Plaintext ASCII:  97, Encrypted ASCII: 11170169, Encrypted and Encoded: 0XAA7179
Plaintext: y  , Plaintext ASCII: 121, Encrypted ASCII: 1781758, Encrypted and Encoded: 0X1B2FFE
Plaintext: a  , Plaintext ASCII:  97, Encrypted ASCII: 11170169, Encrypted and Encoded: 0XAA7179
Plaintext: n  , Plaintext ASCII: 110, Encrypted ASCII: 11945395, Encrypted and Encoded: 0XB645B3
Plaintext:    , Plaintext ASCII:  32, Encrypted ASCII: 1198854, Encrypted and Encoded: 0X124B06
Plaintext: M  , Plaintext ASCII:  77, Encrypted ASCII: 10634362, Encrypted and Encoded: 0XA2447A
Plaintext: o  , Plaintext ASCII: 111, Encrypted ASCII: 5598277, Encrypted and Encoded: 0X556C45
Plaintext: y  , Plaintext ASCII: 121, Encrypted ASCII: 1781758, Encrypted and Encoded: 0X1B2FFE
Plaintext: u  , Plaintext ASCII: 117, Encrypted ASCII: 4873548, Encrypted and Encoded: 0X4A5D4C
Plaintext: r  , Plaintext ASCII: 114, Encrypted ASCII: 34799418, Encrypted and Encoded: 0X212FF3A
Plaintext: a  , Plaintext ASCII:  97, Encrypted ASCII: 11170169, Encrypted and Encoded: 0XAA7179


Encrypted text: 0X8133410XAA71790X1B2FFE0XAA71790XB645B30X124B060XA2447A0X556C450X1B2FFE0X4A5D4C0X212FF3A0XAA7179


Enter your choice: 
```

3. User decrypting cyphertext obtained from the previous option through the second option and exiting the program.



```
Encrypted text: 0X8133410XAA71790X1B2FFE0XAA71790XB645B30X124B060XA2447A0X556C450X1B2FFE0X4A5D4C0X212FF3A0XAA7179


Enter your choice: 2
Enter the ciphertext using private key: 0X8133410XAA71790X1B2FFE0XAA71790XB645B30X124B060XA2447A0X556C450X1B2FFE0X4A5D4C0X212FF3A0XAA7179


Encrypted and Encoded: 0X813341, Encrypted ASCII: 8467265, Plaintext ASCII:  76, Decrypted Plaintext: L
Encrypted and Encoded: 0XAA7179, Encrypted ASCII: 11170169, Plaintext ASCII:  97, Decrypted Plaintext: a
Encrypted and Encoded: 0X1B2FFE, Encrypted ASCII: 1781758, Plaintext ASCII: 121, Decrypted Plaintext: y
Encrypted and Encoded: 0XAA7179, Encrypted ASCII: 11170169, Plaintext ASCII:  97, Decrypted Plaintext: a
Encrypted and Encoded: 0XB645B3, Encrypted ASCII: 11945395, Plaintext ASCII: 110, Decrypted Plaintext: n
Encrypted and Encoded: 0X124B06, Encrypted ASCII: 1198854, Plaintext ASCII:  32, Decrypted Plaintext:
Encrypted and Encoded: 0XA2447A, Encrypted ASCII: 10634362, Plaintext ASCII:  77, Decrypted Plaintext: M
Encrypted and Encoded: 0X556C45, Encrypted ASCII: 5598277, Plaintext ASCII: 111, Decrypted Plaintext: o
Encrypted and Encoded: 0X1B2FFE, Encrypted ASCII: 1781758, Plaintext ASCII: 121, Decrypted Plaintext: y
Encrypted and Encoded: 0X4A5D4C, Encrypted ASCII: 4873548, Plaintext ASCII: 117, Decrypted Plaintext: u
Encrypted and Encoded: 0X212FF3A, Encrypted ASCII: 34799418, Plaintext ASCII: 114, Decrypted Plaintext: r
Encrypted and Encoded: 0XAA7179, Encrypted ASCII: 11170169, Plaintext ASCII:  97, Decrypted Plaintext: a


Decrypted text: Layan Moyura


Enter your choice: 3
Exiting...
---a program by Layan Moyura---
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$
layan@LAPTOP-DUD0NGUN:/mnt/c/Users/acer/Downloads/EE8209_A2_3397$ 
```

References

[1] RSA algorithm in cryptography (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/ (Accessed: 15 August 2023).

[2] Somani, S. (no date) RSA algorithm , C# Corner. Available at: https://www.c-sharpcorner.com/UploadFile/75a48f/rsa-algorithm-with-C-Sharp2/ (Accessed: 16 August 2023).

[3] RSA encryption algorithm - javatpoint www.javatpoint.com. Available at: https://www.javatpoint.com/rsa-encryption-algorithm (Accessed: 17 August 2023).