Experiment no. 05

Aim:

Write a Java/C/C++/Python program to implement RSA algorithm

Objective

Learn how to Perform Encryption and Decryption using RSA algorithm.

Problem Definition

Perform Encryption and Decryption using Asymmetric key cryptography RSA algorithm

Outcome

After completion of this assignment students will be able to understand the Perform Encryption and Decryption using RSA algorithm.

Software and Hardware Requirements

Operating system: 64-bit Windows OS and Linux
RAM :2GB RAM (4GB preferable)
IDE: You have to install **Python3** or higher version or any IDE like **PyCharm**, **Anaconda** ,Dev C++ , eclipse etc

Theory

RSA (Rivest–Shamir–Adleman) algorithm is asymmetric cryptography algorithm. The acronym "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977.Asymmetric actually means that it works on two different keys i.e. Public Key and Private **Key.** As the name describes that the Public Key is given to everyone and Private key is kept private.

An example of asymmetric cryptography :

1. A client (for example browser) sends its public key to the server and requests for some data.
2. The server encrypts the data using client's public key and sends the encrypted data.
3. Client receives this data and decrypts it.

Since this is asymmetric, nobody else except browser can decrypt the data even if a third party has public key of browser.

The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore, encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

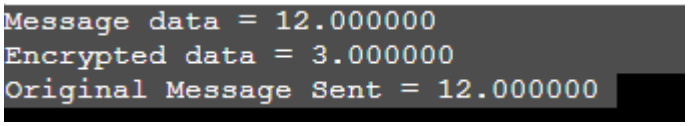
Let us learn the mechanism behind RSA algorithm :

Generating Public Key :

1. Select two prime no's. Suppose **P = 53 and Q = 59**.
2. Now First part of the Public key : **n = P*Q = 3127**.
3. We also need a small exponent say **e** :
4. But e Must be
5.
    a. An integer.
    b. Not be a factor of n.
    c. **1 < e <** Φ(n) [Φ(n) is discussed below],
    d. Let us now consider it to be equal to 3.
6. Our Public Key is made of n and e
    a. >> **Generating Private Key :**
7. We need to calculate Φ(n) :
8. Such that **Φ(n) = (P-1)(Q-1)**
9. so, Φ(n) = 3016
10. Now calculate Private Key, **d** :
11. **d = (k*Φ(n) + 1) / e** for some integer k
12. For k = 2, value of d is 2011.
    a. Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key(d = 2011)
    b. Now we will encrypt **"HI"** :
13. Convert letters to numbers : H = 8 and I = 9
14. Thus **Encrypted Data c = 89^e mod n**.
15. Thus our Encrypted Data comes out to be 1394

16. Now we will decrypt **1394** :
17. **Decrypted Data = c^d mod n**.
18. Thus our Encrypted Data comes out to be 89
19. **8 = H and I = 9 i.e. "HI".**

Output:



Conclusion:

Thus we have studied RSA (Rivest–Shamir–Adleman) algorithm, a asymmetric cryptography algorithm.