MES Wadia College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	**Class: TE Comp**
Semester/Year: 6th	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By: Prof(Dr.)S.K.Wagh	**Experiment No: Part 2-IS-04**

PART 2- LPII-ELII-IS-ASSIGNMENT NO: 04

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

OBJECTIVE:

1. Understand Public Key Cryptography Principles

- Grasp the foundational principles of public key (asymmetric) cryptography, distinguishing it from symmetric cryptography, and understand how these principles are applied in the RSA algorithm.

2. Implement Key Generation

- Develop an algorithm to generate a public-private key pair:

3. Implement the Encryption Process

4. Implement the Decryption Process

5. Ensure Security Practices

- Emphasize the importance of selecting sufficiently large primes and secure random number generation to ensure the cryptographic strength of the key pair.
- Discuss the significance of keeping the private key secure and the implications of its exposure.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative.
- Java, C, C++, or Python

THEORY:

RSA algorithm is an asymmetric cryptography algorithm. 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 the 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 some data.
2. The server encrypts the data using the client's public key and sends the encrypted data.
3. The client receives this data and decrypts it.

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

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 a 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.

RSA Algorithm

Step 1: Generate the RSA modulus

The initial procedure begins with selection of two prime numbers namely p and q, and then calculating their product N, as shown −

N=p*q

Here, let N be the specified large number.

Step 2: Derived Number (e)

Consider number e as a derived number which should be greater than 1 and less than (p-1) and (q-1). The primary condition will be that there should be no common factor of (p-1) and (q-1) except 1

Step 3: Public key

The specified pair of numbers **n** and **e** forms the RSA public key and it is made public.

Step 4: Private Key

Private Key **d** is calculated from the numbers p, q and e. The mathematical relationship between the numbers is as follows −

ed = 1 mod (p-1) (q-1)

The above formula is the basic formula for Extended Euclidean Algorithm, which takes p and q as the input parameters.

Encryption Formula

Consider a sender who sends the plain text message to someone whose public key is **(n,e).** To encrypt the plain text message in the given scenario, use the following syntax −

C = P rest to e mod n

Decryption Formula

The decryption process is very straightforward and includes analytics for calculation in a systematic approach. Considering receiver **C** has the private key **d**, the result modulus will be calculated as −

Plaintext = C rest to d mod n

Mechanism behind the RSA algorithm : >> Generating Public Key:

Select two prime no's. Suppose P = 53 and Q = 59.

Now First part of the Public key : n = P*Q = 3127.

We also need a small exponent say e :

But e Must be An integer.

Not be a factor of Φ(n).

1 < e < Φ(n)

Let us now consider it to be equal to 3.

Our Public Key is made of n and e

>> Generating Private Key:

We need to calculate Φ(n) :

Such that Φ(n) = (P-1)(Q-1)

so, Φ(n) = 3016

Now calculate Private Key, d :

d = (k*Φ(n) + 1) / e for some integer k

For k = 2, value of d is 2011.

Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key(d = 2011)

Now we will encrypt “HI”:

Convert letters to numbers : H = 8 and I = 9

Thus Encrypted Data c = (89e)mod n

Thus our Encrypted Data comes out to be 1394

Now we will decrypt 1394 :

Decrypted Data = (cd)mod n

Thus our Encrypted Data comes out to be 89

8 = H and I = 9 i.e. "HI".

CONCLUSION:

QUESTIONS:

1. What are possible attacks on RSA algorithm? Explain
2. How does the RSA algorithm work for M=5 and primes: p=11 and q=13. Solve it.