Experiment 06

Write a Program to Implement and analyze RSA cryptosystem

Roll No.	19
Name	Manav Jawrani
Class	D15-A
Subject	Security Lab
LO Mapped	LO1: To apply the knowledge of symmetric cryptography to implement classical ciphers. LO2 : Demonstrate Key management, distribution and user authentication.

Aim: Write a Program to Implement and analyze RSA cryptosystem.

Introduction:

RSA:

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, the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:

A client (for example browser) sends its public key to the server and requests for some data.

The server encrypts the data using the client's public key and sends the encrypted data.

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! 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 keys are 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.

Algorithm of RSA: -

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 n.

$1 < e < \Phi(n)$

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, the 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".

Analysis of RSA:
Advantages of RSA:
The RSA algorithm is safe and secure for its users through the use of complex mathematics. The RSA algorithm is hard to crack since it involves factorization of prime numbers which are difficult to factorize. Moreover, the RSA algorithm uses the public key to encrypt data and the key is known to everyone, therefore, it is easy to share the public key.
Disadvantages of RSA:
The RSA algorithm can be very slow in cases where large data needs to be encrypted by the same computer. It requires a third party to verify the reliability of public keys. Data transferred through the RSA algorithm could be compromised through middlemen who might temper with the public key system.

In conclusion, both the symmetric encryption technique and the asymmetric encryption technique are important in encryption of sensitive

Implementation:

Code:

```
import java.util.*;
import java.math.*;
public class RSA {
   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);
      int p, q, n, z, d = 0, e, i;
      System.out.println("Enter the number to be Encrypted and Decrypted: ");
      int msg = sc.nextInt();
      double c; //cipher text
      BigInteger msgback;
      System.out.println("Enter 1st prime number p : ");
      p = sc.nextInt();
      System.out.println("Enter 2nd prime number q : ");
      q = sc.nextInt();
      n = p * q;
      z = (p - 1) * (q - 1); //phi(n)
      System.out.println("The value of z = " + z);
      for (e = 2; e < z; e++) {
         if (gcd(e, z) == 1) {
            break;
         }
      }
      System.out.println("The value of e = " + e);
      for (i = 0; i <= 9; i++) {
         int x = 1 + (i * z);
         if (x % e == 0) {
            d = x / e;
            break;
```

```
            }
        }
        System.out.println("The value of d = " + d);
        c = (Math.pow(msg, e)) % n;
        System.out.println("Encrypted message is: ");
        System.out.println(c);
        BigInteger N = BigInteger.valueOf(n);
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("\n Decrypted message is: ");
        System.out.println(msgback);
    }
    static int gcd(int e, int z) {
        if (e == 0)
            return z;
        else
            return gcd(z % e, e);
    }
}
```

Results:

```
java -cp /tmp/b5w3DHSSw8 RSA
Enter the number to be Encrypted and Decrypted:
74
Enter 1st prime number p :
3
Enter 2nd prime number q :
5
The value of z = 8
The value of e = 3
The value of d = 3
Encrypted message is:
14.0
Decrypted message is:
14
```

Conclusion:

We have understood and implemented and analyzed the RSA cryptosystem.