

**SSN College of Engineering,
Department of Computer Science and Engineering
IT8761 Security Laboratory**

Exercise6 : To implement the Rivest-Shamir-Adleman(RSA) Algorithm.

Programming Language:Java

Code :

```
import java.io.*;
import java.util.*;
import java.lang.*;
import java.math.*;

class RSA{
    public BigInteger p,q,mess;
    public BigInteger e,d,phi,n;
    public BigInteger[] encr;
    // if message is integer
    public int cipher;
    // if message is string
    public String mess_str;

    public RSA(String m){
        this.mess_str=m;
        //mess = new BigInteger(mess_str);
        encr = new BigInteger[100];

        System.out.println("\n\nGiven message : "+mess_str+"\n\n");

        int limit =10;
        String prime=new String("6");
        Random rand = new Random();
        //generating the prime numbers P,Q
```

```

    p = new BigInteger(prime);
    while(!p.isProbablePrime(1)){
        prime=new String();
        for(int i=0;i<limit;i++){
            int dig = rand.nextInt(10);
            char ch = (char)(48+dig);
            prime = prime+ch;
        }

        p = new BigInteger(prime,10);
    }

    prime=new String("6");
    q = new BigInteger(prime);
    while(!q.isProbablePrime(1)){
        prime=new String();
        for(int i=0;i<limit;i++){
            int dig = rand.nextInt(10);
            char ch = (char)(48+dig);
            prime=prime+ch;
        }

        q = new BigInteger(prime,10);
    }

    n = p.multiply(q);
    phi = p.add(BigInteger.valueOf(-1)).multiply(q.add(BigInteger.valueOf(-1)));

    System.out.println("Generated prime number P =
"+p.toString(16).toUpperCase());
    System.out.println("Generated prime number Q =
"+q.toString(16).toUpperCase());
    System.out.println("N = "+n.toString(16).toUpperCase());
    System.out.println("Phi(p,q) = "+phi.toString(16).toUpperCase());
}

//to generate the e,d values
public void keygen(){
    Random rand = new Random();

```

```

    e = BigInteger.valueOf(1256);
    while(e.gcd(phi).compareTo(BigInteger.ONE) != 0){
        int limit=rand.nextInt(30)+1;
        e = new BigInteger(limit, rand);
    }

    d = e.modInverse(phi);

    System.out.println("\nPublic key :
"+"("+e.toString(16).toUpperCase()+","+n.toString(16).toUpperCase()+")");
    System.out.println("\nPrivate key :
"+"("+d.toString(16).toUpperCase()+","+n.toString(16).toUpperCase()+")");
}

public void encrypt(){
    int len = mess_str.length();
    mess_str = mess_str.toLowerCase();
    for(int i=0;i<len;i++){
        char ch = mess_str.charAt(i);
        mess = BigInteger.valueOf(ch-'a'+1);
        encr[i] = mess.modPow(e,n);
    }

    System.out.println("\n\nEncrypted message : ");
    for(int i=0;i<len;i++){
        System.out.print(encr[i].toString(16).toUpperCase()+" ");
    }
}

public void decrypt(){

    String decr=new String();
    int len = mess_str.length();

    for (int i=0;i<len;i++){
        mess = encr[i].modPow(d,n);
        char ch = (char)(96+mess.intValue());
        decr = decr+ch;
    }
}

```

```
        System.out.println("\n\nDecrypted message : "+decr);
    }
}
```

```
public class RSADriver{

    public static void main(String[] args){
        Scanner in = new Scanner(System.in);

        String m;
        int p,q;
        System.out.println("Enter a message : ");
        m = in.nextLine();

        RSA rsa = new RSA(m);

        rsa.keygen();

        rsa.encrypt();

        rsa.decrypt();
    }
}
```

OUTPUT :

Enter a message :
iamkira

Given message : iamkira

Generated prime number P = 1B0B6399B
Generated prime number Q = 12B79D469
N = 1FA32C271A5B4FC93
Phi(p,q) = 1FA32C26EC984EE90

Public key : (148CAB3,1FA32C271A5B4FC93)

Private key : (17B7FF9CC2DAC89FB,1FA32C271A5B4FC93)

Encrypted message :
493F323C3EFB0A24 80980159EC605DD6 1A5605607F703C546
165017F3DCD6F949E 493F323C3EFB0A24 A21869825C6942A6
80980159EC605DD6

Decrypted message : iamkira
(base) Shankars-MacBook-Pro:Ex14 shankar99\$

Result : Implemented the RSA Algorithm in Java and verified it's correctness.