BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA, BENGALURU - 560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Report on RSA Algorithm

Name	Raghavendra K M
USN	1BY18IS093
Semester/Section	5B
Course Code	18CSL57
Course Name	Computer Network Laboratory
Faculty	Prof. Gireesh babu C N
Title	RSA Algorithm
Date	12-11-2020

Signature of a Student **Signature of a Faculty**

Experiment No. 11

Write a program for simple RSA algorithm to encrypt and decrypt the data.

RSA Algorithm:

RSA algorithm is a public key encryption technique and is considered as the most secure way of encryption. It was invented by Rivest, Shamir and Adleman in year 1978 and hence name RSA algorithm.

- RSA is an algorithm used by modern computers to encrypt and decrypt messages.
- It is an Asymmetric Cryptographic Algorithm. Asymmetric means that there are two different keys. This is also called public key cryptography.
- A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key.
- Anyone can use the public key to encrypt a message but only someone with knowledge of the prime factors can feasibly decode the message.

Algorithm:

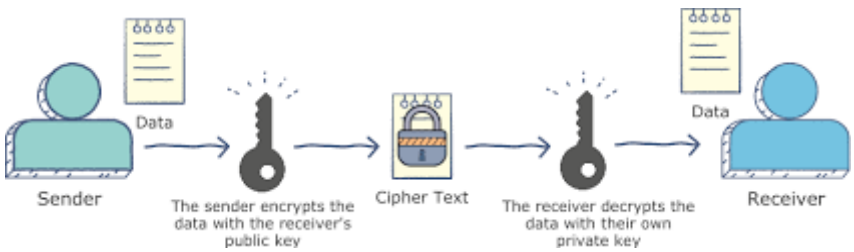
The RSA algorithm involves four steps:

key generation, key distribution, encryption and decryption.

1. Choose two distinct prime numbers p and q.
2. Compute n = pq
3. Compute z= (p-1)(q-1)
4. Choose an integer e, 1<e<z, such that GCD (e, z) = 1
5. Compute the secret exponent d, 1<d<z,

   such that, e * d ≡ 1 (mod z)

1. The public key is (n, e) and the private key is (n, d).
2. Encrypting messages: $c = m^e \bmod n$
3. Decrypting messages: $m = c^d \bmod n$



Source code:

RSA.java

```
import java.math.BigInteger;
import java.io.*;

public class RSA {
    BigInteger p, q, privateKey, publicKey, n, z;
    // BigInteger is used to store and manipulate very large numbers
    // p and q -> Large Prime Numbers
    BufferedReader keyIn = new BufferedReader(new InputStreamReader(System.in));
    String senderMsg, decryptedMsg, encryptedMsg;
    int size; //Size of the Message
    BigInteger m, c;
    // m and c -> used to store the ASCII values of individual characters in
     BigInteger format to encrypt or decrypt

    void read() throws IOException {
        System.out.println(
        "Enter the large prime numbers(p and q: Such that p*q > 127):");
        p = new BigInteger(keyIn.readLine()); // Read p
        q = new BigInteger(keyIn.readLine()); // Read q
        n = p.multiply(q); // n = p*q
        z = (p.subtract(BigInteger.ONE)).multiply(q.subtract(BigInteger.ONE));
        // z = (p-1)(q-1)
        System.out.println("Enter the public exponent (e):");
        publicKey = new BigInteger(keyIn.readLine()); // Read Public Key (e)
        privateKey = new BigInteger("0"); // Set Private Key (d) to 0
        BigInteger temp;
        // do-while loop to generate private key (d) such that (e*d) mod z = 1
        do {
            privateKey = privateKey.add(BigInteger.ONE); // d++
            temp = (privateKey.multiply(publicKey)).mod(z);
                // computing (e*d) mod z
        } while (!temp.equals(BigInteger.ONE));
                 // the value of d is accepted if temp==1
        System.out.println("Enter Message to Encrypt:");
        senderMsg = keyIn.readLine();
        size = senderMsg.length();
        encryptedMsg = "";
        decryptedMsg = "";
    }

    void encrypt() {
        for (int i = 0; i < size; i++) {
            m = BigInteger.valueOf((int) senderMsg.charAt(i));
                //Copy each character of msg into m
            c = m.modPow(publicKey, n); // Calculate c = (m^e) mod n
            encryptedMsg += (char) c.intValue();
```

```
                // Add encrypted character into code
        }
    }

    void decrypt() {
        for (int i = 0; i < size; i++) {
            c = BigInteger.valueOf((int) encryptedMsg.charAt(i));
                //Copy each character of code into c
            m = c.modPow(privateKey, n); // Calculate m = (c^d) mod n
            decryptedMsg += (char) m.intValue();
                // Add decrypted character into message
        }
    }

    void show() {
        System.out.println
        ("\nThe Message Entered at Sender's end is \"" + senderMsg + "\"");
        System.out.println
       ("The Encrypted Message sent to the Receiver is \"" + encryptedMsg + "\"");
        System.out.println
        ("The Decrypted Message at Receiver's end is \"" + decryptedMsg + "\"");
    }

    public static void main(String[] args) throws IOException {
        RSA obj = new RSA();
        obj.read();
        obj.encrypt();
        obj.decrypt();
        obj.show();
    }
}
```

Outputs:

C:\Windows\System32\cmd.exe

```
D:\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\RSA algorithm>javac RSA.java

D:\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\RSA algorithm>java RSA
Enter the large prime numbers(p and q: Such that p*q > 127):
29
17
Enter the public exponent (e):
5
Enter Message to Encrypt:
raghu

The Message Entered at Sender's end is "raghu"
The Encrypted Message sent to the Receiver is "?%4??"
The Decrypted Message at Receiver's end is "raghu"

D:\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\RSA algorithm>
```