Shreyas Patil

BEA-1851011

ASSIGNMENT: 1

Write a program in C++ or Java to implement RSA algorithm for key generation and cipher verification

Input:

package ICS\_PRACT;

import java.math.\*;

import java.security.SecureRandom;

public class RSA\_algo{

private static BigInteger one=new BigInteger("1");

private static BigInteger n;

private static BigInteger phi;

private static BigInteger e;

private static BigInteger d;

public RSA\_algo()

{

BigInteger p=BigInteger.probablePrime(40, new SecureRandom());

BigInteger q=BigInteger.probablePrime(40, new SecureRandom());

n=p.multiply(q);

phi=(p.subtract(one)).multiply(q.subtract(one));

e=new BigInteger("65537"); //only use prime number

d=e.modInverse(phi);

}

public BigInteger encrypt(BigInteger msg)

{

return msg.modPow(e, n);

}

public BigInteger decrypt(BigInteger msg)

{

return msg.modPow(d, n);

}

@Override

public String toString() {

return "Public key="+e.longValue()+"\nPrivate key="+d.longValue()+"\nModuli="+n.longValue();

}

public static void main(String[] args) {

RSA\_algo rsa=new RSA\_algo();

System.out.println(rsa);

String msg="BEIT";

System.out.println("Plaintext="+msg);

BigInteger plaintext=new BigInteger(msg.getBytes());

BigInteger ciphertext=rsa.encrypt(plaintext);

System.out.println("Encrypted message="+new String(ciphertext.toByteArray()));

BigInteger dplaintext=rsa.decrypt(ciphertext);

System.out.println("Decrypted message="+new String(dplaintext.toByteArray()));

}

}

Output:

Public key=65537

Private key=4594109734395052161

Moduli=6214760654083016385

Plaintext=BEIT

Encrypted message=\_2¸Íø‘‡b¢\_

Decrypted message=BEIT