Experiment No:3

**Aim:** Write program to implement RSA algorithm

**Theory:**

RSA is 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, because one of them can be given to everyone. The other key must be kept private. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described it in 1978. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message.

**Operation:**

RSA involves a public key and private key. The public key can be known to everyone; it is used to encrypt messages. Messages encrypted using the public key can only be decrypted with the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two different large random prime numbers p and q
2. Calculate n = pq
3. Calculate the totient: ϕ ( n ) = ( p − 1 ) ( q − 1 )
4. Choose an integer e such that 1 < e < ϕ ( n ), and e is coprime to ϕ ( n ) ie**:** e and ϕ ( n ) share no factors other than 1
5. Compute d to satisfy the congruence relation d e = 1 ( mod ϕ ( n ) ) ie: d e = 1 + k ϕ ( n ) for some integer k.

The **public key** is made of the modulus n and the public (or encryption) exponent e.

The **private key** is made of the modulus n and the private (or decryption) exponent d which must be kept secret.

**Encrypting messages:**

Alice gives her public key ( n & e) to Bob and keeps her private key secret. Bob wants to send message **M** to Alice. First he turns **M** into a number m smaller than n by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to:

c = me mod n.

**Decrypting messages**

Alice can recover m from c by using her private key d in the following procedure:

m = cd mod n

Given m, she can recover the original message **M**.

**Example:**

Here is an example of RSA encryption and decryption.

* Choose two random prime numbers : p = 61 and q = 53;
* Compute n = p q : n = 61 ∗ 53 = 3233
* Compute the totient ϕ ( n ) = ( p − 1 ) ( q − 1 ) : ϕ ( n ) = ( 61 − 1 ) ( 53 − 1 ) = 3120
* Choose e > 1 coprime to 3120 : e = 17
* Choose d to satisfy d e = 1 ( mod ϕ ( n ) ) : d = 2753 i.e, 17 ∗ 2753 = 46801 = 1 + 15 ∗ 3120
* The public key is ( n = 3233, e = 17 ).
* For a padded message m the encryption function is:
  + c = me mod n : m17mod 3233
* The private key is ( n = 3233, d = 2753 ).
* The decryption function is:
  + m = cd mod n : c2753 mod 3233
* For example, to encrypt m = 123 , we calculate
  + c = 12317mod3233 = 855
* To decrypt c = 855, we calculate
  + m = 8552753mod3233 = 123

**Program:**

import java.util.\*;

class RSA {

static int phi, M, n, e, d, C, FLAG;

public static void main(String args[])throws Exception {

int p, q, s;

Scanner sc = new Scanner(System.in);

System.out.print("Enter Two Relatively Prime Numbers: ");

p = sc.nextInt();

q = sc.nextInt();

n = p \* q;

phi = (p - 1) \* (q - 1);

do {

System.out.print("Enter e: ");

e = sc.nextInt();

} while (FLAG == 1);

d = 1;

do {

s = (d \* e) % phi;

d++;

} while (s != 1);

d = d - 1;

System.out.println("Public Key :" + e + " " + n);

System.out.println("Private Key:" + d + " " + n);

System.out.print("Enter The Plain Text: ");

M = sc.nextInt();

encrypt();

System.out.print("Enter the Cipher text: ");

C = sc.nextInt();

decrypt();

sc.close();

}

public static void encrypt()throws Exception {

int i;

C = 1;

for (i = 0; i < e; i++)

C = C \* M % n;

C = C % n;

System.out.println("Encrypted Text:" + C);

}

public static void decrypt()throws Exception {

int i;

M = 1;

for (i = 0; i < d; i++)

M = M \* C % n;

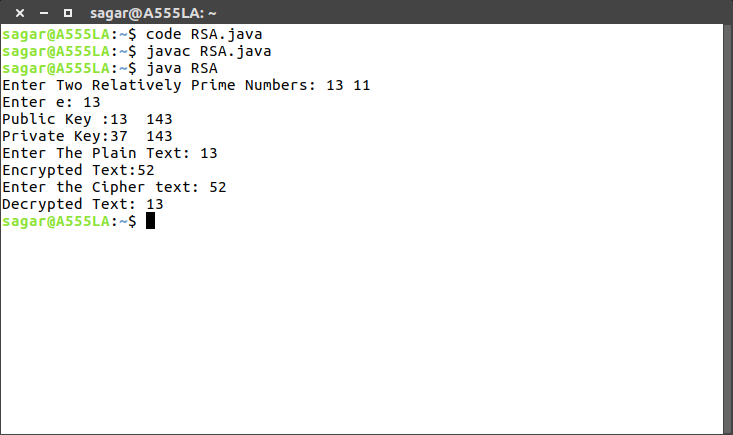
M = M % n;

System.out.println("Decrypted Text: " + M);

}

}

**Output:**



**Conclusion:**

Hence, RSA Algorithm program was executed and verified successfully.