Name: Drishti Bhatia **Roll Number: 07**
SYMCA **Block Chain Lab**

Practical No: 02

Aim: To implement RSA and show the encryption as well as decryption process.

Theory:

❖ RSA

- RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e., **Public Key** and **Private Key.** As the name describes that the Public Key is given to everyone and 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 client's public key and sends the encrypted data.
  - Client receives this data and decrypts it.

- In RSA cryptography, both the public and the private keys can encrypt a message. The opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method to assure the confidentiality, integrity, authenticity, and non-repudiation of electronic communications and data storage.

- RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total -- or factoring -- is considered infeasible due to the time it would take using even today's supercomputers.

- In RSA, public key is used for encryption and private key is used for decryption.

❖ Description of the Algorithm

1. Choose two large prime number P and Q.
2. Calculate ***N = P x Q.***
3. Select the public key (i.e., the encrypt ion key) E such that it is not a factor of ***(P - 1) and (Q - 1).***
4. Select the private key (i.e., the decryption key) D such that the following equation is true: ***(D x E) mod (P- 1) x (Q - 1) = 1***
5. For encryption, calculate the Cipher text CT from the plain text PT as follows: ***CT=PT E mod N***
6. Send CT as the cipher text to the receiver.
7. For decryption. Calculate the plain text PT from the cipher text CT as follows: ***PT =CTD mod N***

Name: Drishti Bhatia **Roll Number: 07**
SYMCA **Block Chain Lab**

Code:

```
import math

def gcd(m, n):
  if n==0:
    return abs(m)
  else:
    return gcd(n,m%n)


p = int(input("Enter a prime number(p) : "))
q = int(input("Enter a prime number(q) : "))
n = p*q
e = int(input("Enter public key: "))
phi = (p-1)*(q-1)

while (e < phi):
  if(gcd(e, phi) == 1):
    break
  else:
    e = e+1

d = 2 #Calculating private key : d
while d < phi:
  if ((d*e) % phi)==1:
    break
  d += 1

pt = float(input("Enter plain text : "))
print("Plain Text = ", pt)

# Encryption c = (pt ^ e) % n
c = pow(pt, e)
c = math.fmod(c, n)
print("Encrypted data = ", c)

# Decryption m = (c ^ d) % n
m = pow(c, d)
m = math.fmod(m, n)
print("Original Message Sent = ", m)
```

Output:

```
Enter a prime number(p) : 3
Enter a prime number(q) : 11
Enter public key: 7
Enter plain text : 2
Plain Text =  2.0
Encrypted data =  29.0
Original Message Sent =  2.0
```

Conclusion: We have successfully implemented RSA and shown the encryption as well as decryption process.