**LAB 5**

**OBJECTIVE**

To implement RSA encryption algorithm

**THEORY**

RSA (Rivest–Shamir–Adleman) 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, because one of the keys can be given to anyone. The other key must be kept private. The algorithm is based on the fact that finding the factors of a large composite number is difficult: when the factors are prime numbers, the problem is called prime factorization. It is also a key pair (public and private key) generator.

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.

**CODE**

# Python program to implement RSA algorithm.

import pyinputplus as pyip

import math, random

def is\_prime(n):

count = 0

for i in range(1, n + 1):

if n % i == 0:

count += 1

if count == 2:

return True

else:

return False

def calculate\_d(totient\_n, e):

for i in range(2, totient\_n):

if (e \* i) % totient\_n == 1:

return i

while True:

print("Enter two prime numbers : ")

p = pyip.inputNum()

q = pyip.inputNum()

if is\_prime(p) and is\_prime(q) and p != q:

break

n = p \* q

print("n : ", n)

totient\_n = (p - 1) \* (q - 1)

print("totient(n) : ", totient\_n)

e\_list = []

for i in range(2, totient\_n):

if math.gcd(totient\_n, i) == 1:

e\_list.append(i)

# print("e\_list: ", e\_list)

while True:

#e = pyip.inputInt("Select an integer from e\_list: ")

e = random.choice(e\_list)

if e in e\_list:

break

print("e : ", e)

d = calculate\_d(totient\_n, e)

print("d : ", d)

PU = [e, n]

print("Public key : ", PU)

PR = [d, n]

print("Private key : ", PR)

m = pyip.inputInt("Enter message : ", lessThan = n)

c = m \*\* e % n

print("Cipher text : ", c)

m = c \*\* d % n

print("Decrypted text : ", m)

**OUTPUT**

Enter two prime numbers :

89

97

n : 8633

totient(n) : 8448

e : 2759

d : 5879

Public key : [2759, 8633]

Private key : [5879, 8633]

Enter message : 6794

Cipher text : 1625

Decrypted text : 6794

**CONCLUSION**

In this lab, we got familiar with the RSA encryption technique, implemented it using python programming language and tested for different prime input sets and different messages.