# EXPERIMENT NO 02

CLASS: TE CMPN A PID: 182027

NAME: REBECCA DIAS ROLL NO. : 19

**Aim**:

a. To implement RSA cryptosystem

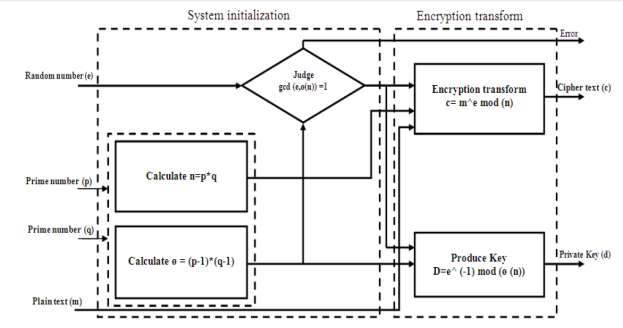
**Theory:**

1. What is public key cryptography?

Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses two mathematically related, but not identical, keys - a public key and a private key. Unlike symmetric key algorithms that rely on one key to both encrypt and decrypt, each key performs a unique function. The public key is used to encrypt and the private key is used to decrypt.

It is computationally infeasible to compute the private key based on the public key. Because of this, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures, and private keys can be kept secret, ensuring only the owners of the private keys can decrypt content and create digital signatures.

2. Block diagram of RSA algorithm



3. Description of RSA algorithm

· RSA algorithm is an asymmetric (public key) cryptography algorithm.

· Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key.

· As the name describes, the Public Key is given to everyone and the Private key is kept private.

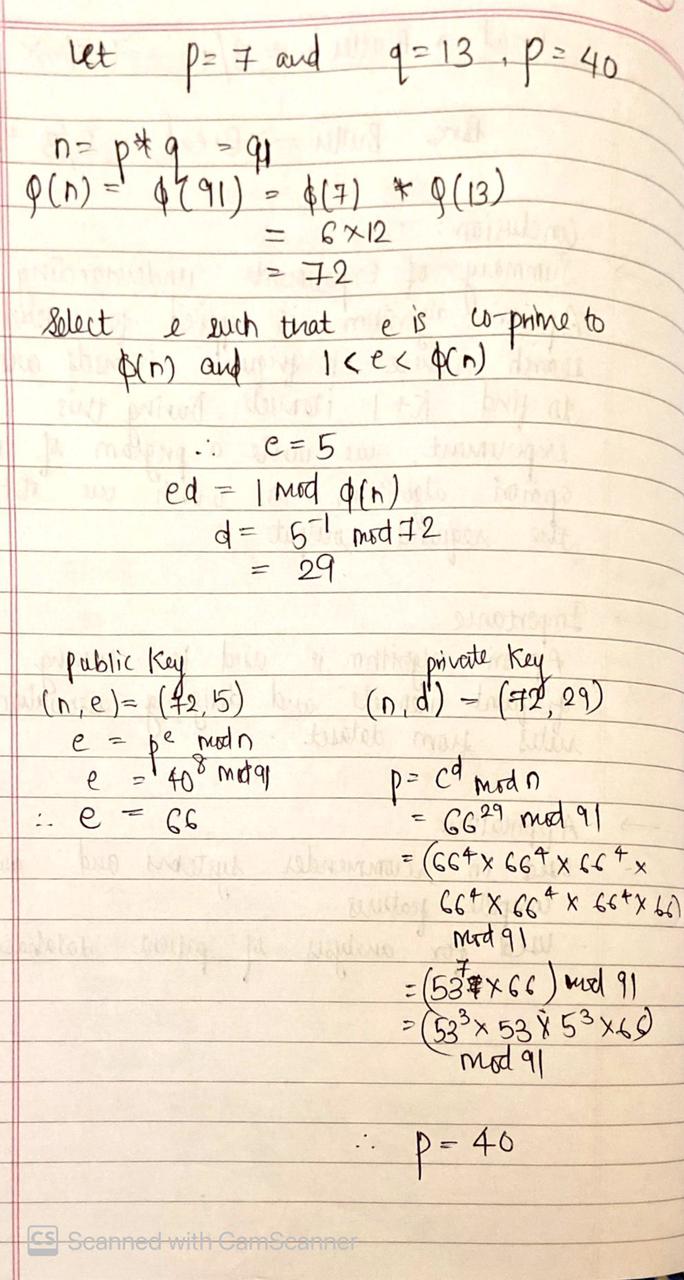
· The RSA algorithm is named after those who invented it in 1978: Ron Rivest, Adi Shamir, and Leonard Adleman.

· The RSA algorithm holds the following features −

a) RSA algorithm is a popular exponentiation in a finite field over integers including prime numbers.

b) The integers used by this method are sufficiently large making it difficult to solve.

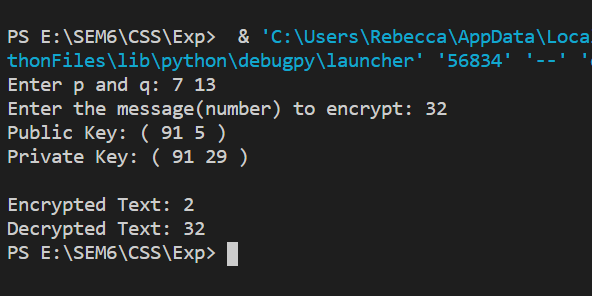
c) There are two sets of keys in this algorithm: private key and public key.

4. Theoretically solve RSA al

**Implementation:**

|  |
| --- |
| import math def mulinv(k):  r1, r2, t1, t2 =26,k,0,1  while r1 != 1:  if r2 ==0:  break  q=r1//r2  r=r1%r2  t = t1-(t2\*q)  r1=r2  r2=r  t1=t2  t2=t  if t1>0:  return t1  else:  return t2+t1  def gcd(p,q):  while q != 0:  p, q = q, p%q  return p   def coprime(x):  prime = [i for i in range(5,x)]  for i in prime:  g =gcd(x,i)  if g == 1:  # print(i)  return i  def key\_gen(n,p,q):  phi = (p-1)\*(q-1)  e = coprime(phi)  for i in range(phi):  if (i\*e)%phi == 1:  d=i  break  return e,d  def rsa\_encrypt(m,e,n):  return (pow(m,e)%n)  def rsa\_decrypt(m,d,n):  return (pow(m,d)%n)  def RSA():  p,q = map(int,input("Enter p and q : ").split(" "))  n=p\*q  e,d =key\_gen(n,p,q)  m = int(input("Enter the message(number) to encrypt : "))  c = rsa\_encrypt(m,e,n)  p = rsa\_decrypt(c,d,n)  print("Public Key: (",n,e,")\nPrivate Key: (",n,d,")\n")  print("Encrypted Text:",c)  print("Decrypted Text:",p)  choice = 0 while choice != 3:  print("Menu:\n1.RSA Encryption Algorithm")  choice = int(input())  if choice == 1:  RSA() |

**Output:**



**Conclusion:**

In this experiment we learnt to implement the RSA Algorithm to encrypt and decrypt the given message text. We also learnt to find the coprime and calculate the totient function.

**Viva Questions:**

1. Explain the working of the RSA algorithm.

The following steps highlight how it works:

* Generating the keys
* Encryption (The pair of numbers (n,e) makes up the public key.)
* Decryption (The pair (n,d) makes up the private key.)

Generating the keys

* Select two large prime numbers, p and q.
* Calculate n=p ∗ q.
* Calculate the totient function; φ(n)
* Select an integer e, such that e is co-prime to φ(n) and 1<e<φ(n).
* Calculate d such that d= e-1 mod φ(n). i.e e.d = 1 mod φ(n)

*Note: d can be found using the extended euclidean algorithm.*

Encryption

* Given a plaintext P, represented as a number, the ciphertext
* C is calculated as:
* C = P​e mod n

Decryption

* Given a Ciphertext C, represented as a number, the plaintext
* P is calculated as:
* P = C​d mod n

2. What are the advantages of RSA algorithm? Explain with an example.

Advantage

1. RSA is stronger than any other symmetric key algorithm.
2. RSA has overcome the weakness of symmetric algorithm i.e. authenticity and confidentiality.

Example:

RSA is still seen in a range of web browsers, email, VPNs, chat and other communication channels.

RSA is also often used to make secure connections between VPN clients and VPN servers. Under protocols like OpenVPN, TLS handshakes can use the RSA algorithm to exchange keys and establish a secure channel.

3. What are the disadvantages of RSA algorithm? Explain with an example

Disadvantage

1. RSA has too much computation.

Example:

RSA is a relatively slow algorithm. Because of this, it is not commonly used to directly encrypt user data. More often, RSA is used to transmit shared keys for symmetric key cryptography, which are then used for bulk encryption-decryption.