3. Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.
4. Compute the secret exponent d, 1 < d < phi, such that ed ≡ 1 (mod phi).
5. The public key is (n, e) and the private key is (n, d). The values of p, q, and phi should also be kept secret.

- n is known as themodulus.
- e is known as the public exponent or encryptionexponent.
- d is known as the secret exponent or decryptionexponent.

Note: It is possible to find a smaller d by using lcm(p-1,q-1) instead of phi, lcm(p-1,q-1) = phi /gcd(p-1,q-1) ).

**Encryption**

Sender A does the following:-

1. Obtains the recipient B's public key (n,e).
2. Represents the plaintext message as a positive integerm.
3. Computes the ciphertext c = m^e modn.
4. Sends the ciphertext c toB.

**Decryption**

Recipient B does the following:-

1.Uses his private key (n, d) to compute m = c^d modn.

2.Extracts the plaintext from the integer representativem.

**/* RSA Key Generation */**

**Source Code:**

```
import java.util.*;
import java.io.*;
public class rsa
{
  static int gcd(int m,int n)
  {
  while(n!=0)
  {
        int r=m%n;
        m=n;
        n=r;
  }
  return m;
  }
public static void main(String args[])
{
```

```
int p=0,q=0,n=0,e=0,d=0,phi=0;
int nummes[]=new int[100];
int encrypted[]=new int[100];
int decrypted[]=new int[100];
int i=0,j=0,nofelem=0;
Scanner sc=new Scanner(System.in);
String message ;
System.out.println("Enter the Message tobe encrypted:");
message= sc.nextLine();
System.out.println("Enter value of p and q\n");
p=sc.nextInt();
q=sc.nextInt();
n=p*q;
phi=(p-1)*(q-1);
for(i=2;i<phi;i++)
    if(gcd(i,phi)==1)
        break;
    e=i;
for(i=2;i<phi;i++)
    if((e*i-1)%phi==0)
    break;
d=i;
for(i=0;i<message.length();i++)
{
    char c = message.charAt(i);
    int a =(int)c;
    nummes[i]=c-96;
}
nofelem=message.length();
for(i=0;i<nofelem;i++)
{
    encrypted[i]=1;
    for(j=0;j<e;j++)
    encrypted[i] =(encrypted[i]*nummes[i])%n;
}
System.out.println("\n Encrypted message\n");
for(i=0;i<nofelem;i++)
{
    System.out.print(encrypted[i]);
    System.out.print((char)(encrypted[i]+96));
}
for(i=0;i<nofelem;i++)
{
    decrypted[i]=1; for(j=0;j<d;j++)
```

```
        decrypted[i]=(decrypted[i]*encrypted[i])%n;
    }

    System.out.println("\n Decrypted message\n ");
    for(i=0;i<nofelem;i++)
        System.out.print((char)(decrypted[i]+96)); return;
  }
}
```

**Output**

Enter the text:
hello
Enter the value of P and Q :
5
7
Encrypted Text is: 8 h 10 j 17 q 17 q 15 o
Decrypted Text is: hello

**Program Outcome**

- Implement data link layer protocols. Identify and apply the operation of RSA algorithm.

**Viva Questions:**

- What is RSA? Explain its algorithm.
- What do you mean by encryption and decryption of data?