Experiment-3
RSA Algorithm

AIM: Implementation RSA algorithm

DESCRIPTION: RSA algorithm was invented by Ronald L. Rivest, Adi Shamir, and Leonard Adleman in 1977 and released into the public domain on September 6, 2000. It can be used for Authentication, key exchange and encryption. It is an example of public key encryption or asymmetric encryption algorithm with two keys, where one key is used for encryption and other key is used for decryption. The keys are generated using mathematical relation.

The RSA algorithm uses the fact that it's easy to multiply two large prime numbers together and get a product. But you can't take that product and reasonably guess the two original numbers, or guess one of the original primes if only the other is known. The public key and private keys are carefully generated using the RSA algorithm; they can be used to encrypt information or sign it.

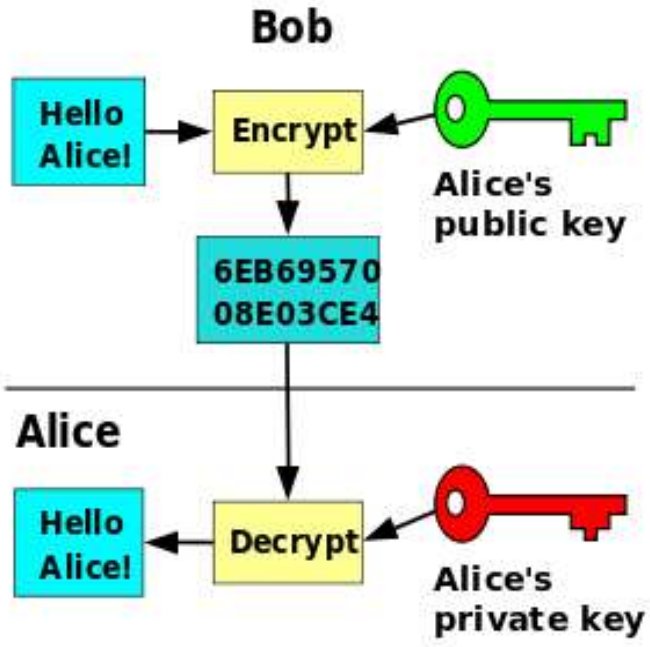

Figure 3.1: Block diagram showing the RSA algorithm

ALGORITHMS:

a. Key Generation:

1) Select two large prime numbers say **p** and **q**, where *p != q;*
2) Computer **n=p x q**;
3) Calculate **n = p × q**;
4) Compute **ø (n) = (p − 1)(q − 1);**
5) Select **e**, so that gcd(e, ø (n)) = 1, 1 < e < ø (n);
6) Calculate d, such that d · e mod ø (n) = 1, i.e. d is the multiplicative inverse of e in mod ø(n);
7) Get public key as **KU = {e, n};**
8) Get private key as **KR = {d, n}.**

b. Encryption

For a given plaintext block **P < n**, its cipher text (C)

C = P^e (mod n)

c. Decryption

For cipher text block **C**, its plaintext (**P**)

P= C^d (mod n)

IMPLEMENTATION:

/* C program for the Implementation of RSA Algorithm Encrypt the text data and Decrypt the same */

```
#include<stdio.h>
#include<conio.h>
int phi,M,n,e,d,C,FLAG;

int check()                    //Function to e is relatively prime to ø (n)
{
   int i;
   for(i=3; e%i == 0 && phi % I == 0; i +2)
    {
       FLAG = 1;
       return;
    }
    FLAG = 0;
}
```

```
void encrypt()                          //Function to encrypt the plain text message M
{
    int i;
   C = 1;
   for(i=0;i< e;i++)
       C=C*M%n;
   C = C%n;
   printf("\n\tEncrypted keyword : %d",C);
}

void decrypt()                          //Function to decrypt the cipher text C into plaintext M
{
    int i;
   M = 1;
   for(i=0;i< d;i++)
   M=M*C%n;
   M = M%n;
   printf("\n\tDecrypted keyword : %d",M);
}

void main()                             //main function
{
int p,q,s;
clrscr();
printf("Enter Two Relatively Prime Numbers\t: ");
scanf("%d%d",&p,&q);
n = p*q;
phi=(p-1)*(q-1);                                    //computation of ø value
printf("\n\tF(n) phi value\t= %d",phi);
do
{
     printf("\n\nEnter e which is prime number and less than phi \t: ",n);
     scanf("%d",&e);
     check();
}while(FLAG==1);
d = 1;
do
{
      s = (d*e)%phi;
      d++;
 }while(s!=1);
d = d-1;
printf("\n\tPublic Key\t: {%d,%d}",e,n);
printf("\n\tPrivate Key\t: {%d,%d}",d,n);
printf("\n\nEnter The Plain Text\t: ");
scanf("%d",&M);
encrypt();
printf("\n\nEnter the Cipher text\t: ");
```

```
scanf("%d",&C);
decrypt();
getch();
}
```

RESULTS AND DISCUSSIONS:

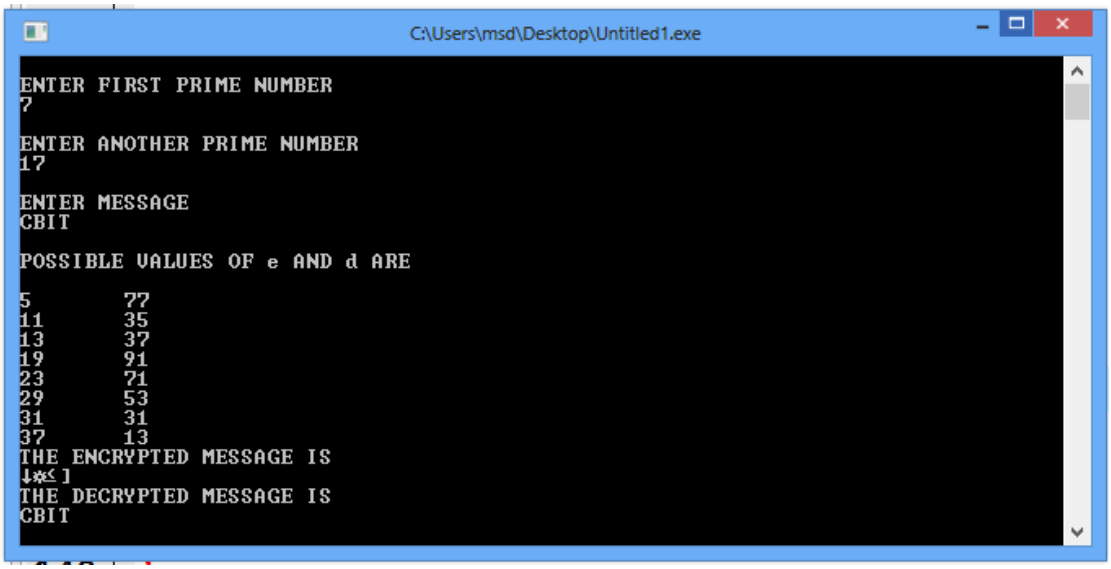

Figure 3.2: Sample screenshot showing the implementation results of RSA algorithm

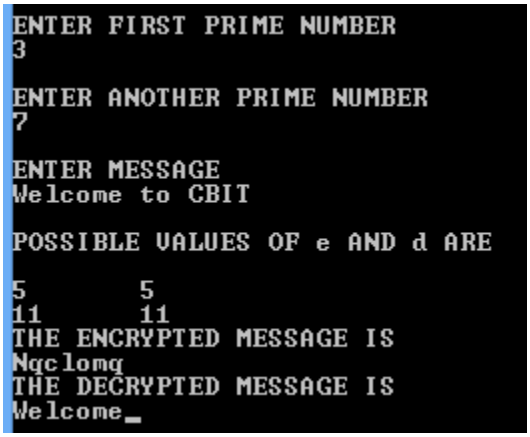

Figure 3.3: Snapshot for which the algorithm has not computed for the full text

- The above program was implemented using GNU C compiler and tested by selecting different key and messages.
- Same code hasn't produced the full plain text when "welcome to CBIT" was given
- The above algorithm also tests for the primality of given numbers

CONCLUSIONS:

- RSA algorithm is implemented successfully and tested with several keys and messages
- In this experiment we have used the RSA algorithm for encryption purpose. It can also be used for key exchange and authentication.

References:

1) https://sourcecode4all.wordpress.com/2012/03/28/rsa-algorithm-in-c/
2) http://www.coders-hub.com/2013/04/c-code-to-encrypt-and-decrypt-message.html#.Vroi9_l97cc