

Experiment-4

RSA Algorithm

4.1 AIM: Implementation RSA algorithm

4.2 DESCRIPTION: Encryption algorithms are basically used for providing confidentiality to the information. They are basically divided into **private key** (symmetric) or **public key** (Asymmetric) encryption algorithms. Private Key encryption algorithms use the same key for encryption and decryption while public key algorithms use two keys where one key is used for encryption and the other is used for decryption. Some of the examples for private key encryption algorithms are: *DES, IDEA, AES, Blowfish, RC4, RC5* etc. *RSA, Digital signatures* etc. are examples of public key encryption algorithms.

RSA algorithm was invented by Ronald L. Rivest, Adi Shamir, and Leonard Adelman in 1977 and released into the public domain on September 6, 2000. It can be used for **Authentication, key exchange** and **encryption**. 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.

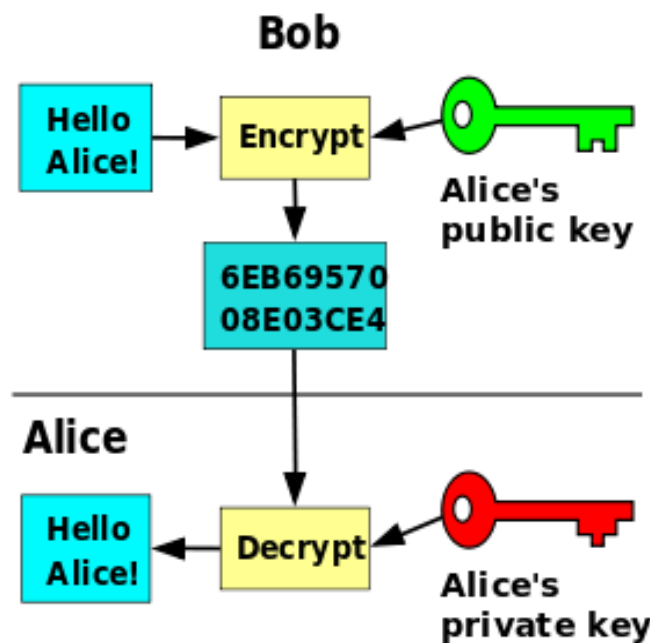


Figure 4.1: Block diagram showing the RSA algorithm

4.3 ALGORITHMS

a. Key Generation:

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

b. Encryption

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

$$C = P^e \pmod{n}$$

c. Decryption

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

$$P = C^d \pmod{n}$$

4.4 IMPLEMENTATION:

/* C program for the Implementation of RSA Algorithm */

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

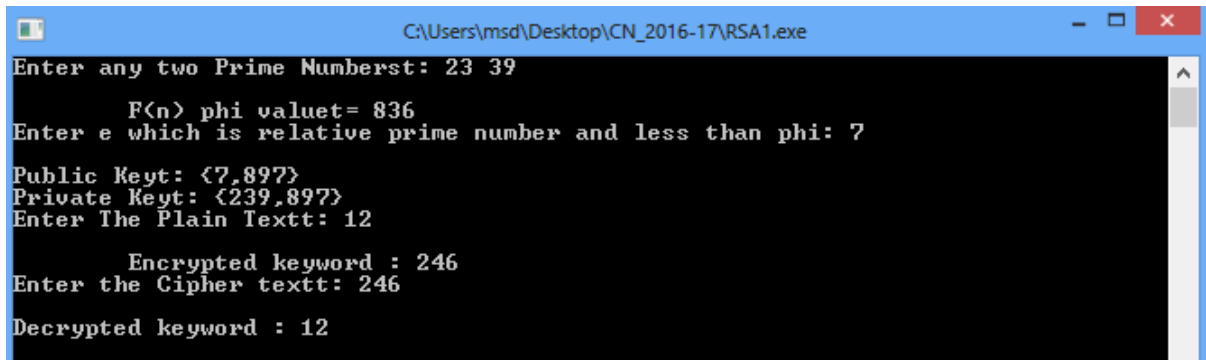
int check()                                //Function that checks whether e is relatively prime to φ(n)
{
    int i;
    for(i=3; e%i == 0 && phi % i == 0; i +2)
    {
        FLAG = 1;
        return;
    }
    FLAG = 0;
}

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);
}

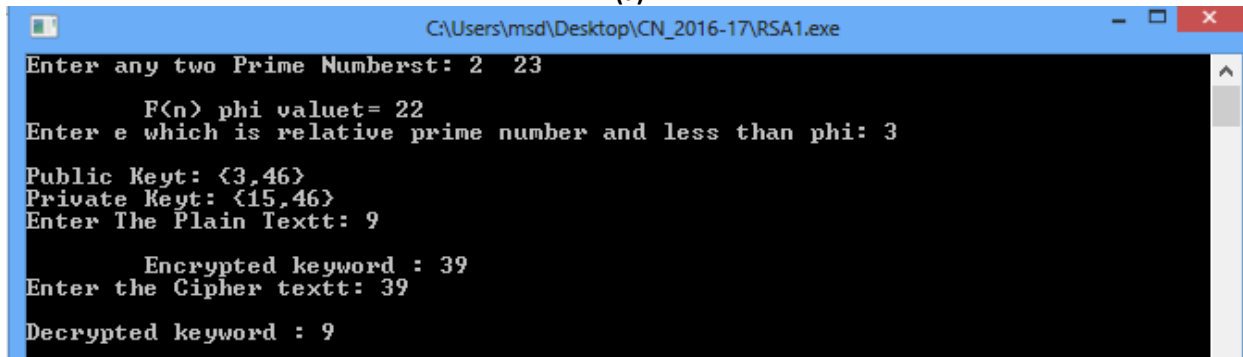
void main()                                  //main function
{
    int p,q,s;
    clrscr();
    printf("Enter any two Prime Numbers\t: ");
    scanf("%d%d",&p,&q);
    n = p*q;
    phi=(p-1)*(q-1);                          //computation of φ value
    printf("\n\tF(n) phi value\t= %d",phi);
    do
    {
        printf("\n\nEnter e which is relatively prime 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();
}
```

4.4 RESULTS AND DISCUSSIONS



```
C:\Users\msd\Desktop\CN_2016-17\RSA1.exe
Enter any two Prime Numberst: 23 39
      F(n) phi valuet= 836
Enter e which is relative prime number and less than phi: 7
Public Key: {7,897}
Private Key: {239,897}
Enter The Plain Textt: 12
      Encrypted keyword : 246
Enter the Cipher textt: 246
Decrypted keyword : 12
```

(a)



```
C:\Users\msd\Desktop\CN_2016-17\RSA1.exe
Enter any two Prime Numberst: 2 23
      F(n) phi valuet= 22
Enter e which is relative prime number and less than phi: 3
Public Key: {3,46}
Private Key: {15,46}
Enter The Plain Textt: 9
      Encrypted keyword : 39
Enter the Cipher textt: 39
Decrypted keyword : 9
```

(b)

Figure 4.2: Snapshot for which the algorithm has not computed for the full text

Tasks to be performed:

1. Check the above program with various inputs and identify where it fails
2. Modify the above program that encrypts and decrypts text messages.
(<http://proprogramming.org/program-to-implement-rsa-algorithm-in-c/>)
3. Test the program, take the snapshot, incorporate your comments in the record and submit

4.5 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) http://www.coders-hub.com/2013/04/c-code-to-encrypt-and-decrypt-message.html#.Vroi9_I97cc
- 2) <https://sourcecode4all.wordpress.com/2012/03/28/rsa-algorithm-in-c/>
- 3) <http://proprogramming.org/program-to-implement-rsa-algorithm-in-c/>