ISSN No. 0976-5697

Volume 4, No. 2, Jan-Feb 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Implementation of Public Key RSA Algorithm Using 'C' Language

Sonia Thind
Department of Information Technology
DAV College Sec-10 Chandigarh, India
soniathind04@yahoo.com

Abstract: RSA is one of the most-common used algorithms for public-key cryptography applied for encrypting information in computer-communication systems and suitable for signing and encryption. Implementing RSA algorithm in 'C language' is described in the paper. This paper will be applicable to the educational process in the course of 'Network Security'.

Keywords: RSA; network security; digital signature; encryption; decryption

I. INTRODUCTION

RSA is a public key algorithm invented in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman (RSA). Encryption and Digital Signatures are supported by RSA.[1].

This paper is intended to develop a 'C' program for RSA algorithm for the network security. All the steps in the algorithm are explained in the output of 'C language'. This research consists of 7 steps. In the first step, choose any two prime numbers. If any number entered by the user is not prime the program exits.

II. ALGORITHM

a. Choose two prime numbers 'p' and 'q'.
b. Compute n = p X q
c. 3. Compute phi = (p-1)X (q-1) [Euler's totient function]
d. d X e = 1(mod phi)
   1< e < phi
   'e' should be a prime
   'e' and phi should be co-prime i.e. gcd (phi,e)=1
e. Calculate value of 'd' using "Extended Euclidean Algorithm's Table Method".
   If d > phi Then
   d = d mod phi
   End If
   If d < 0 Then
   d = d + phi
   End If
f. Encryption of plain text 'P' and Encrypted value is C=Pe (mod n)
g. Decryption of 'C' at receiving end to get 'P' is given by P=Cd (mod n)

III. IMPLEMENTATION OF RSA IN 'C' LANGUAGE

```
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
int phi,p,n,e,d,c,flag;
int checke()
{
        int i;
        for(i=3;e%i==0&&phi%i==0;i+2)
        {
 flag=1;
                return 0;
        }
        flag=0;
        return 0;
}
void checkprime(int n)
{
        int i;
        for(i=2;i<=n-1;i++)
        {
                if(n%i==0)
                {
                        printf("\n Not prime no's");
                        getch();
                        exit(0);
                }
        }
}
int encrydecry(unsigned long long int base,unsigned long long int exponent,unsigned long long int n)
{
int result=1;
 while(exponent>0)
{
 if(exponent==0)
 break;
 if(exponent%2==1)
 {
 result=(result*base)%n;
 }
 base=(base*base)%n;
 exponent=exponent/2;
}
return result;
}
void main()
{
        int p,q,s;
        clrscr();
        printf("\n STEP I \n");
        printf("\t Enter two relatively prime numbers");
        scanf("%d%d",&p,&q);
        checkprime(p);
        checkprime(q);
```

```
n=p*q;
printf("\n STEP II \n");
printf("\n \t  n=%d",n);
phi=(p-1)*(q-1);
printf("\n STEP III \n");
printf("\n\t Phi=%d",phi);
printf("\n STEP IV \n");
do
{
        printf("\n\t Choose  e\t");
        scanf("%d",&e);
        checkprime(e);
        checke();
 }while(flag==1);
if(flag==1)
{
printf("\n \t\t Wrong value of e");
printf("\n \t\t\t***END***");
getche();
exit(0);
}
printf("\n STEP V");
printf("\n \t Calculate value of 'd' using Extended Euclidean Algorithm's Table Method");
int a[100], b[100],d1[100],k[100];
a[1]=1,a[2]=0,b[1]=0,b[2]=1,d1[1]=phi,d1[2]=e;
printf("\n ID \t a \t b \t d \t k");
printf("\n 1 \t %d \t %d \t %d",a[1],b[1],d1[1]);
k[2]=d1[1]/d1[2];
printf("\n 2 \t %d \t %d \t %d \t %d",a[2],b[2],d1[2],k[2]);
int i;
for(i=3;d1[i-1]>1;i++)
{
        a[i]=a[i-2]-(a[i-1]*k[i-1]);
        b[i]=b[i-2]-(b[i-1]*k[i-1]);
        d1[i]=d1[i-2]-(d1[i-1]*k[i-1]);
        k[i]=d1[i-1]/d1[i];
        printf("\n %d \t %d \t %d \t %d \t %d",i,a[i],b[i],d1[i],k[i]);
}
int d=b[i-1];
printf("\n value of d= %d",d);
if(d<0)
{
d=d+phi;
printf("\n As d=negative therefore Actual value of d= %d",d);
}
if(d>phi)
{
d=fmod(d,phi);
printf("\n As d<phi, therefore Actual value of d= %d",d);
}
printf("\n\t Public key{ %d,%d}",e,n);
printf("\n\t Private key{%d,%d}",d,n);
printf("\n\n Enter the plain text\t");
scanf("%d",&p);
int x=encrydecry(p,e,n);
printf("\n \t Encrypted keyword is %d",x);
printf("\n \t Decrypted keyword is %d",encrydecry(x,d,n));
getch();
}
```

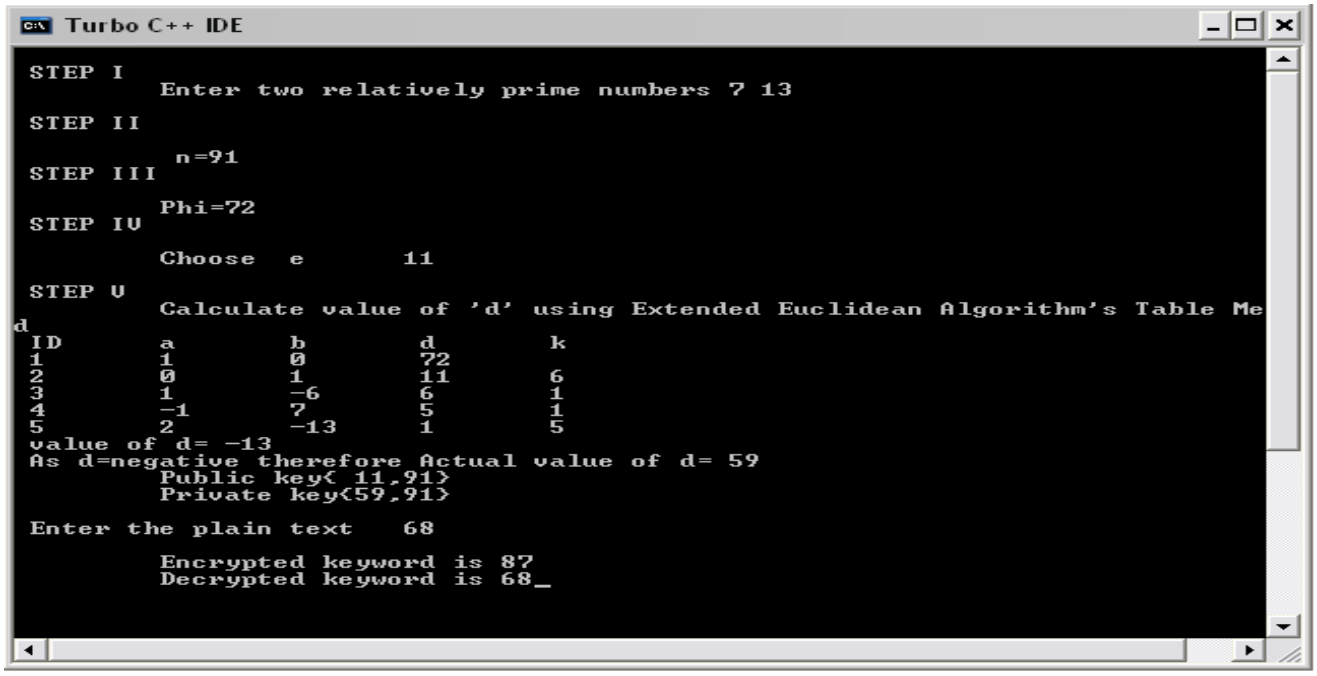

Figure: 1 Output of RSA algorithm in 'C'.

Here plain text up to 185 is given by user and corresponding to plain text, data comes in the encrypted form. Encryption is the correct method to implement confidentiality for Internet traffic [2].

RSA is an asymmetric key algorithm. It is based on the concept of a key pair. i.e. one key can encrypt information that only the other key can decrypt [3]. The key pair is associated to one another. Asymmetric key pairs use confidentiality and digital signatures [4].

IV. CONFIDENTIALITY AND DIGITAL SIGNATURES

A. Confidentiality Using Asymmetric Key Pairs (Encryption):

a. 'A' desires to send a confidential message to 'B'
b. 'A' retrieves B's public key and encrypts the message with it [5].

B. Confidentiality Using Asymmetric Key Pairs (Decryption):

'B' receives the message and decrypts the message with the secretly held, private key

The only key that can possibly decrypt a message that is encrypted with B's public key is B's private key [5].

C. Digital Signatures Using Asymmetric Key Pairs (Encryption):

'A' desires to send a digitally signed message to 'B'

'A' uses their own private key to encrypt a part of the message

'A' sends the encrypted part of the message to 'B' [5].

D. Digital Signatures Using Asymmetric Key Pairs (Decryption):

'B' receives Trading Partner A's message and obtains A's public key

'B' tries to decrypt the encrypted portion of A's message

If it decrypts, Then 'B' knows it has to be from A because the only thing A's public key will decrypt is something encrypted with A's private key and only 'A' has access to that private key [5].

V. CONCLUSION

Implementation of RSA algorithm in 'C language' is presented in paper. Here plain text is in decimal number system form. For example in the output shown 68 is plain text, encrypted keyword for 68 is 87 and then decryption is applied on encrypted keyword (87) and get back 68 as decrypted keyword. This paper will help students to learn how RSA algorithm is implemented and step wise output of the program is given that help better and easily understandable.

VI. REFERENCES

[1] A.J. Menezes , P.C. Van Oorschot y S.A. Vanstone, "Handbook of Applied Cryptography", CRC Press 5th Printing (August 2001) ISBN 0-8493-8523-7, page no.15-28 and. 283-291.

[2] Douglas R. Stinson, "Cryptography: Theory and Practice, Third edition, CRC Press. 2005, ISBN 10: 1584885084.

[3] SV Kartalopoulos, " A primer on cryptography in communications",IEEE Communications Magazine, vol. 44, Fourth edition , 2006, page no. 146–151.

[4] W. Stallings, "Cryptography and Network Security: Principles and Practice", Prentice-Hall, Second Edition, 1999, ISBN 0-13-869017-0, page no.592-593.

[5] Wade Trappe and Washington, L., "Introduction to Cryptography with Coding Theory", Prentice Hall , Second Edition, 2006, ISBN-10: 0131862391, , page no. 285-288