|  |
| --- |
| CODE |

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*File : p11.cpp

\*Description : Program for simple RSA algorithm to encrypt and decrypt the data.

\*Author : Arpith (1PE10CS018)

\*Tools : Ubuntu 13.04, gcc 4.7.3 compiler, Code::Blocks

\*Date : 13 September 2013

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RSA ALGORITHM

--------------------------------------------------------------------------------------------------------

1. Generate two large random primes, p and q, of approximately equal size.

2. Compute n=p\*q;

3. Calculate another number Φ = (p-1)\*(q-1)

4. Choose a random number e. Then calculate d so that d\*e=1 mod phi

5. We can now announce e and n to public. Φ and d is kept secret

Hence, the public key is (n, e) and the private key is (Φ, d).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <stdlib.h>

#include <string.h>

#include <math.h>

using namespace std;

long int i, e, d, n, p, q, phi, cipher[50];

int encrypt(char ch);

char decrypt(long int ch);

int gcd(long int a,long int b);

int prime(int a);

int main()

{

int i,len;

char text[50];

cout<<"Enter the text to be encrypted: ";

cin>>text;

len = strlen(text);

do

{

p = rand()%30; // generate prime p

}while(!prime(p));

do

{

q = rand()%30; // generate prime number q

}while(!prime(q));

n=p\*q; //compute n and Î¦

phi=(p-1)\*(q-1);

do

{

e = rand()%phi; //compute E

}while(gcd(e,phi)!=1);

do

{

d = rand()%phi; //compute d

}while(((d\*e)%phi)!=1);

cout<<"\n\nTwo prime numbers (p and q) are: "<<p<<" and "<<q<<endl;

cout<<"n(p\*q) = "<<p<<" \* "<<q<<" = "<<p\*q<<endl;

cout<<" Φ (p-1)(q-1) = ("<<p<<"-1) \* ("<<q<<"-1) = "<<(p-1)\*(q-1)<<endl;

cout<<"Public key (n, e): ("<<n<<", "<<e<<")\n";

cout<<"Private key (Î¦, d): ("<<phi<<", "<<d<<")\n";

//Encrypt the plain text

for(i=0; i<len; i++)

cipher[i]=encrypt(text[i]);

cout<<"\n\nEncrypted message: ";

for(i=0; i<len; i++)

cout<<cipher[i];

//Decrypt the cipher text

for(i=0; i<len; i++)

text[i]=decrypt(cipher[i]);

text[i]='\0';

cout<<"\nDecrypted message: "<<text<<endl;

return 0;

}

int gcd(long int a,long int b)

{

if(a==0) return b;

if(b==0) return a;

return gcd(b ,a%b);

}

int prime(int a)

{

for(i=2; i<a; i++)

if((a%i)==0)

return 0;

return 1;

}

int encrypt(char ch)

{

long int temp=ch;

for(i=1; i<e; i++)

temp=(temp\*ch)%n;

return temp;

}

char decrypt(long int ch)

{

long int temp=ch;

for(i=1; i<d; i++)

ch=(temp\*ch)%n;

return ch;

}

|  |
| --- |
| **OUTPUT** |

