**Department Of Computer Science and Applications**

**Panjab University**

**Chandigarh**

****

**Project Report on Text Encryption Using RSA**

*(Session 2019-20)*

**SUBMITTED TO: SUBMITTED BY:**

**Prof. M.Syamala Devi Abhay Singh MCA-II (Evening)**

**Roll No. 51**

**CERTIFICATE**

***TO WHOM IT MAY CONCERN***

This is to certify that **ABHAY SINGH** pursuing Master of Computer Applications, at Panjab University, Chandigarh, undertook a project entitled **“Text Encryption Using RSA algorithm**”, the project is a bona fide work carried out by him under our supervision. This work has not been submitted earlier, either in part or in full, to any other university or institute for the award of degree.

***Prof. M.Syamala Devi***

***(Project Guide)***

***DCSA, Panjab University***

***Chandigarh***

**ABOUT THE PROJECT**

The project is about text-encryption using the RSA algorithm. I have started this project by writing the code for creating the menus in the program for switching between the different functionalities, where each option from the menu calls a function to perform a particular task. After creating the menu driven command line interface, I have written the functionalities corresponding to each function call.

The main feature of this program is to encrypt the text data using the private key and decrypt it using the public key, for this I have implemented three function this first one encrypt the data directly from the keyboard input, the second encrypt data of any text file stored in the hard-drive and last one is used to decrypt the data.

The latest version of C++ supports only integer type of size 64 bits only, but to use the RSA algorithm I needed the data-types which can store very large numbers therefore, to solve this problem I have used the boost multiprecision library which has allowed me user the integer data type of the size up to 4096 bits.

The program is very easy to use and can be used by anyone who wants to encrypt their text document & can only be decrypted by providing the right keys. I have tested this program with different set of inputs and text file and remove all the possible bugs came to my notice.

The program can further modified into GUI program to provide more ease of access to the users.

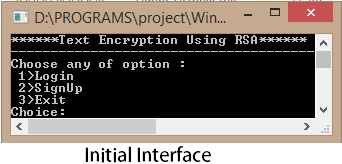
.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Title** | **Page no.** |
| **I** | **Introduction** | **1-2** |
| **II** | **Design** | **3-6** |
| **III** | **Implementation** | **7-9** |
| **IV** | **Testing** | **10-20** |
| **V** | **Conclusions & Scope for further Development** | **21** |
| **VI** | **References** | **22** |

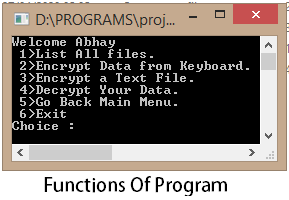
**Teacher’s sign- \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction**



The Text Encryption program is a command line program developed using the c++ programming language. It implements the RSA algorithm to encrypt the text data character by character.

To use the program, the user is required to create a user-account using the sign up option from the program menu. To create an account the user has to provide a unique username and password after which the program will automatically generate the private and the public keys for the user.



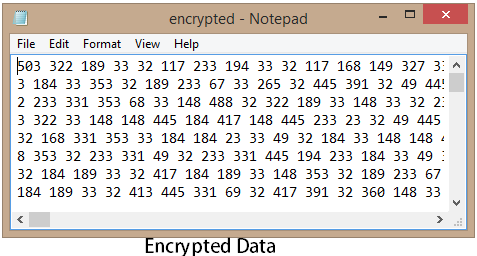
After creating the account, a user can login into the program by providing the username and password. on successful login the user is given six options to use the program, which includes options to encrypt/decrypt data and various other options as show in the picture:

The program is also able to handle any invalid input entered by the user

And prompt the user to enter the correct input till the user enters the correct input or exits the program by choosing the last option and confirming the exit prompt.

***Goal of the Project***

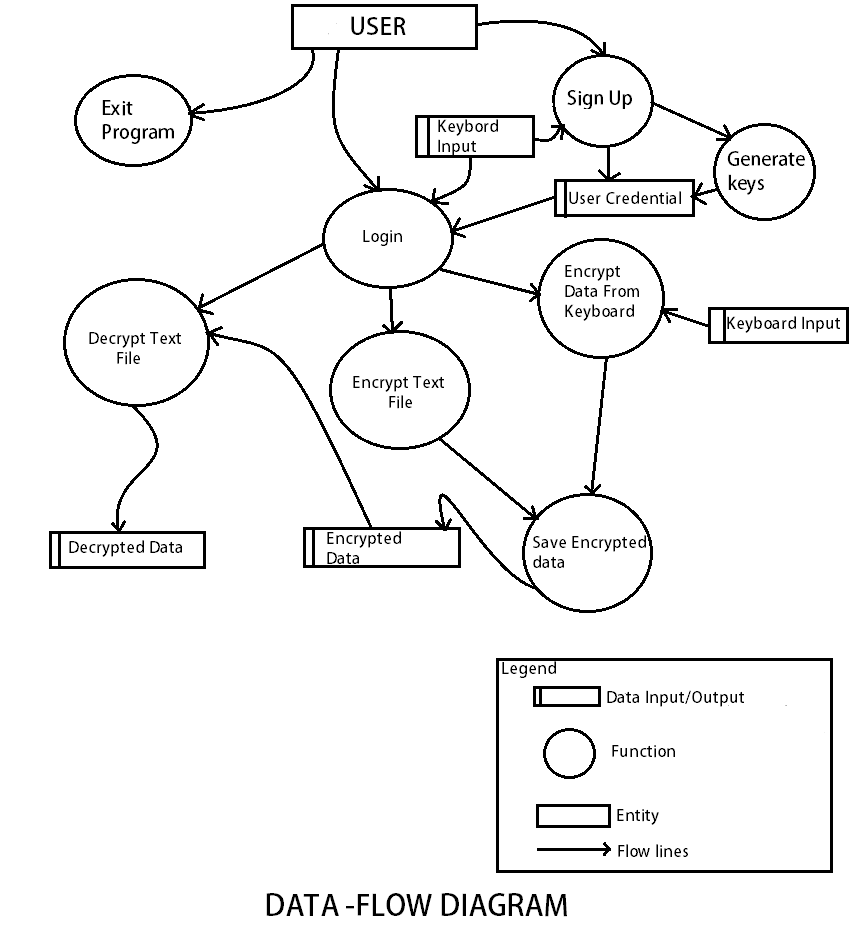
The main goal of the project is to create a very efficient program which can encrypt the text documents providing security and reliability. The encrypted document contains information in unreadable format, since all the information is converted into numerical format by applying the RSA encryption on each character.



***Objectives of the Project***

* **Implementing RSA algorithm:** One of the main objectives of the program is to show the working of the RSA algorithm for data encryption.
* **Saving The Data:** Another objective of the program is to save the encrypted to files by using the input & output stream class of the c++.
* **Security:** The security objective of the program is met by implementing the user-authentication feature in program.
* **Decrypting The Encrypted Data:** This is another relevant objective of the program to retrieve the data into its original format after proper authentication.

**Program Design**



**Program Implementation**

***Programming Language Used: C++***

***IDE Used: Code Blocks 17.12***

***Compiler Used: GNU GCC(c++ version 14)***

***Operating System: Microsoft Windows***

Source Code:

#include<windows.h>***//For Sleep (milliseconds)***

#include<iostream>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

#include<fstream>

#include<cstring>

#include <boost/multiprecision/cpp\_int.hpp> ***//For using larger integer numbers***

using namespace boost::multiprecision;

using namespace std;

namespace mp = boost::multiprecision;

***//For Using the int4096\_t data type***

typedef mp::number<mp::cpp\_int\_backend<4096, 4096, mp::signed\_magnitude, mp::unchecked, void> > int4096\_t;

***//\*\*\*\*\*\*\*\*\*\*\*classSection\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

class Users{

private:

int64\_t p,q,n,phiN,e,d;

char ch,name[20];

int4096\_t temp,ct,msg;

char uname[20],pwd[20];

void credential();

int authenticate();

int64\_t gcd(int64\_t a,int64\_t b);

int64\_t getE();

int64\_t getD(double phiN,double e);

int4096\_t power(int4096\_t x,int4096\_t y);

void genKeys();

void saveFile();

int getRandom();

public:

void login();

void signUp();

int exitProg();

void rsaFileEncrypt();

int rsaEncrypt();

void rsaDecrypt();

void pressKey();

};

int Users::getRandom(){

srand(time(0));

/\*It set the seed different using the time

If srand() is not called the seed is set srand(1) by default\*/

return rand();

}

void Users::pressKey(){

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Press any key to continue..."<<endl;

getch();

}

//GCD function

int64\_t Users::gcd(int64\_t a,int64\_t b){

int64\_t temp;

while(1){

temp=a%b;

if(temp==0)

return b;

a=b;

b=temp;

}

}

//For calculating e(Private key) for encryption

int64\_t Users::getE(){

int64\_t retE=14;

while(retE<phiN)

{

if(gcd(retE,phiN)==1)

break;

retE++;

}

return retE;

}

//For calculating d (public Key) for decryption

int64\_t Users::getD(double phiN,double e){

double d,flag,i=1;

int64\_t temp;

do{

d=((phiN\*i)+1)/e;

temp=d;

flag=d-temp;

i++;

}while(flag!=0.0);

return d;

}

//For find powers of BigIntegers

int4096\_t Users::power(int4096\_t x,int4096\_t y){

int i;

int4096\_t res=1;

for(i=0;i<y;i++){

res=res\*x;

}

return res;

}

***//For generating all RSA key variables***

void Users::genKeys(){

int arr[]={5,7,11,13,17,19,23,29,31,37};

int i,j,rInt;

cout<<"Generating Keys Wait...."<<endl;

again:

n=0;

do{

rInt=getRandom();

i=rInt%10;

rInt=getRandom();

j=rInt%10;

if(i!=j){

p=arr[i];

q=arr[j];

n=p\*q;

}

}while((n<127)||(n>700));

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

e=getE();

d=getD(phiN,e);

if(e==d){

cout<<" Processing...."<<endl;

goto again;

}

}

void Users::saveFile(){

while(true){

cout<<"\n\nSave This Data to File Y/N:";

cin>>ch;

cin.clear();

cin.ignore(100,'\n');

if(ch=='Y'||ch=='y'){

cout<<"Save File as :";

cin>>name;

cin.ignore(100,'\n');

if(rename("encrypted.dat",name)==0){

cout<<"File Saved!"<<endl;

Sleep(2000);

break;

}

else {

cout<<"FIle NOT Saved."<<endl;

cout<<" Check Same File Name Exist\n Try Again!"<<endl;

}

}

else if(ch=='n'||ch=='N'){

break;

}

else{

cout<<"Enter Correct Input Y/N"<<endl;

}

}

}

void Users::rsaFileEncrypt(){

system("cls");

cout<<"Please name file to encrypt : ";

cin>>name;

ofstream out("encrypted.dat",ios::out);

fstream rdChar(name,ios::in);

if(!out){

cout<<"Failed to create ofstream object!";

pressKey();

return;

}

if(!rdChar){

cout<<" File not Found!"<<endl;

cout<<" Enter valid name with extension!"<<endl;

pressKey();

return;

}

else{

cout<<" Found the File!"<<endl;

// cout<<"Now enter your private key : ";

//cin>>e;

}

cout<<"Encrypting Data!"<<endl;

while(rdChar.get(ch)){

int m=ch;

temp=power(m,e);

ct=temp%n;

out<<ct<<' ';

cout<<ct;

}

cout<<"\nData Encryption Complete !"<<endl;

rdChar.close();

out.close();

saveFile();

}

int Users::rsaEncrypt(){

system("cls");

string str;

int i=0;

// cout<<"Please enter your private key : ";

//cin>>e;

cout<<"Now enter the message :";

getline(cin,str);

ch=str[i];

ofstream out("encrypted.dat",ios::out);

if(!out){

cout<<"Failed to create ofstream object!";

exit(-1);

}

cout<<"Encrypting Data!"<<endl;

while(ch!='\0'){

int m=ch;

temp=power(m,e);

ct=temp%n;

out<<ct<<' ';

cout<<ct;

i++;

ch=str[i];

}

cout<<"\nData Encryption Complete !"<<endl;

out.close();

saveFile();

}

void Users::rsaDecrypt(){

system("cls");

int a;

cout<<"Name the File.extension to Decrypt : ";

cin>>name;

fstream encInt(name,ios::in);

if(!encInt){

cout<<"The file You have entered not found!"<<endl;

pressKey();

return;

}

//cout<<"Now enter your public key : ";

//cin>>d;

//Saving the dec into decrypted.dat

fstream decFile("decrypted.dat",ios\_base::out);

while (encInt >> a)

{ ct=a;

temp=power(ct,d);

msg=temp%n;

decFile<<msg<<' ';

}

encInt.close();

decFile.close();

fstream msgFile("decrypted.dat",ios\_base::in);//for Reading decInt

fstream txtFile("originalTxt.dat",ios\_base::out);//For saving char

cout<<"The Decrypted data"<<endl;

while(msgFile>>a){

cout<<(char)a;

txtFile<<char(a);

}

msgFile.close();

txtFile.close();

while(true){

cout<<"\n\nSave The Data? Y/N :";

cin>>ch;

if(ch=='y'||ch=='Y'){

cout<<"Please name the file:";

cin>>name;

if(rename("originalTxt.dat",name)==0)

cout<<"File Saved Successfully!"<<endl;

else

cout<<"Failed to Save the file"<<endl;

break;

}

else if(ch=='n'||ch=='N')

{ cout<<"Going back to previous menu"<<endl;

//delete both files decrypted.dat and originalTxt.dat

break;

}

else

{ cout<<"Invalid Option! Try Again!"<<endl;

}

}

}

//\*\*\*\*\*\*to Close The Program\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int Users::exitProg(){

char choice;

cout<<"Exit The Program (Y/N) :";

cin>>choice;

cin.clear();

cin.ignore(20,'\n');

if(choice=='Y'||choice=='y')

exit(0);

else if(choice=='n'||choice=='N'){

cout<<"Cancelled!"<<endl;

Sleep(1000);

}

else{

cout<<"Please press Y or N"<<endl;

Sleep(1000);

}

return 1;

}

//\*\*\*\*\*\*\*\*\*\*\*AuthenticatingCredential.dat\*\*\*\*\*\*\*\*\*

int Users::authenticate(){

char psw[20],u[20];

//read uname,pwd,pvtkey from file and match with obeject

ifstream rd("credential.dat",ios::in);

if(!rd){

cout<<"The credential Data failed to load"<<endl;

pressKey();

}

while(true){

rd.getline(u,20,'#');//For reading Username

rd.getline(psw,20,'#');//For reading Password

rd>>p;

rd>>q;

rd>>e;

rd>>d;

//red.getline(pk,20,'#');//For Reading Private key

if((strcmp(u,uname)==0)&&(strcmp(psw,pwd)==0))

{ n=p\*q;

return 1;

}

if(rd.tellg()==0||rd.tellg()==-1)

break;

}

cout<<"Check username and password!"<<endl<<" Try Again!"<<endl;

pressKey();

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*credentialInput\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Users::credential(){

int i=0;

ch='\n';

system("cls");

cout<<"Enter Your Details..."<<endl;

cout<<" Username : ";

cin>>uname;

cout<<" Password : ";

do{

ch=getch();

if(ch!=8&&ch!=13&&ch!=' ')

{

cout<<"\*";

pwd[i]=ch;

i++;

}

else if (ch==8&&i>0) // if backspace is presssed

{

cout<<"\b \b"; // moves cursor to the left print <space> again move cursor to left

i--;

}

else

{

}

}

while(ch!=13);

pwd[i]='\0';

cout<<endl;

}

//\*\*\*\*\*\*\*\*\*\*user\_login\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Users::login(){

int choice,flag=0;

credential();

flag=authenticate();

if(flag==1){

cout<<"Login Successfull!"<<endl;

Sleep(1000);

do{ system("cls");

cout<<"Welcome "<<uname<<endl;

cout<<" 1>List All files."<<endl;

//Always show the encypted data to console!

cout<<" 2>Encrypt Data from Keyboard."<<endl;

cout<<" 3>Encrypt a Text File."<<endl;

//Show Data to the console and also Save to file

cout<<" 4>Decrypt Your Data."<<endl;

cout<<" 5>Go Back Main Menu."<<endl;

cout<<" 6>Exit"<<endl;

cout<<"Choice : ";

cin>>choice;

cin.clear();

cin.ignore(100,'\n');

switch(choice){

case 1:system("cls");

cout<<"Files under current directory!"<<endl;

system("dir \*.\*");

cout<<endl<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"Press any key to continue.."<<endl;

getch();

break;

case 2:rsaEncrypt();//keyEncrypt();

break;

case 3:rsaFileEncrypt();

break;

case 4:rsaDecrypt();//decData();

break;

case 5:return;

break;

case 6:exitProg();

default:system("cls");

cout<<"Invalid Input!"<<endl;

break;

}

}while(true);

}

}

//\*\*\*\*\*\*\*\*\*\*user Sign up\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Users::signUp(){

system("cls");

char psw[20],u[20];

int t1,t2,t3,t4;

here:

int i=0; //After the here: to reset i to 0

credential();

cout<<" Re-enter Your Password: ";

do{

ch=getch();

if(ch!=8&&ch!=13&&ch!=' ')

{

cout<<"\*";

psw[i]=ch;

i++;

}

else if (ch==8&&i>0) // if backspace is presssed

{

cout<<"\b \b";

i--;

}

else{}

}

while(ch!=13);

psw[i]='\0';

cout<<endl;

if(strcmp(psw,pwd)!=0){

cout<<"Password do not match try again"<<endl;

pressKey();

goto here;

}

else if(strlen(pwd)<8){

cout<<"Password should be of at least of 8 character!"<<endl;

pressKey();

goto here;

}

else{

fstream out("credential.dat",ios::in);

if(!out){

// cout<<"Can't Verify User-names!"<<endl;

out.close();

}

else{

while(true){

out.getline(u,20,'#');//For reading Username

out.getline(psw,20,'#');//For reading Password

out>>t1;

out>>t2;

out>>t3;

out>>t4;

if((strcmp(u,uname)==0))

{ cout<<"Choose Different Username!"<<endl;

pressKey();

goto here;

}

if(out.tellg()==0||out.tellg()==-1)

break;

}

out.close();

}

fstream wrt("credential.dat",ios::app);

if(!wrt){

cout<<"Unable TO Load Credential.dat"<<endl;

pressKey();

}

else{

genKeys();

wrt<<uname<<'#'<<pwd<<'#'<<'\n';

wrt<<p<<' '<<q<<' '<<e<<' '<<d;

cout<<"Account Created Successfully!"<<endl;

cout<<" Private Key : "<<e<<endl;

cout<<" Public Key : "<<d<<endl;

wrt.flush();

pressKey();

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*dataEncryption\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main(){

int choice,i=0;

Users u1;

do{

system("cls");

cout<<"\*\*\*\*\*\*Text Encryption Using RSA\*\*\*\*\*\*"<<endl;

cout<<"--------------------------------------"<<endl;

cout<<"Choose any of option :"<<endl;

cout<<" 1>Login"<<endl;

cout<<" 2>SignUp"<<endl;

cout<<" 3>Exit"<<endl;

cout<<"Choice: ";

cin>>choice;

cin.clear();//It clear cin error flag which stops further input

cin.ignore(1000,'\n');/\*It ignores 1000 char and stop on encountering newline char

avoid infinite loop on entering non-int and avoid parse failure\*/

switch(choice){

case 1:u1.login();

break;

case 2:u1.signUp();

break;

case 3:u1.exitProg();

break;

default:system("cls");

cout<<"Press 1,2,3 only!"<<endl;

Sleep(1500);

break;

}

}while(true);

return 0;

}

**Testing**

The program with is tested with

**Conclusion**

**Reference**