

ABSTRACT

Unauthorized access to your precious files and folders can be carried out remotely as well as physically. By Physical access, we mean an intruder can attempt to exploit your private information stored in your personal computer in your absence. The attacker can even get hold of your hardware through your files. Through the Folder a locker we aim to achieve our goal of providing the user with layers of physical security.

Keywords: Security, Intruder, Physical security

TABLE OF CONTENT

Introduction	1
Problem Statement	2
Objective	2
Literature Review	3
Methodology	5
Software Requirements.....	6
Use Case.....	7
Pert Chart.....	8
Result	9
Conclusion.....	10
References.....	11
Appendix.....	12

INTRODUCTION

App locks have been designed for android based device such as phones but such app locks are currently not available for the windows. This product would be beneficial to an individual as much as to an organization. People generally store their work related documents on their personal computer's. Hence they would feel much secure if their valuable information is stored in a secure vault. People really struggle to come up with strong passwords that can prevent basic brute force and Dictionary attacks. By implementing ciphers we can help the user to generate passwords with high entropy measure. Also we will be implementing innovative as well as secured password retrieval as well as intruder all mechanisms, to make our product as useful as possible.

We have come up with the idea by keeping the following parameters in our mind:-

1. Assessing the risks
2. Identifying the viable assets
3. Providing solutions

a.) **Assessing the risks**

Organizations and individuals sometimes underestimate the importance of keeping their offices and equipment physically secure. Even those who take steps to protect hardware like computers and backup storage devices from theft, severe weather and other physical threats often fail to document these steps in a written security policy. As discussed above our main aim is to prevent an intruder from physically exploiting our personal as well as professional files.

b.) **Identifying the viable assets**

Information regarding the employees, assets, customer, and new products are very essential to the organization. Any loss to this data will result into financial loss to the company. This type of data can be stored in the simplest form possible that is text files.

c.) **Providing Solutions**

We can provide the user with a mechanism that provides him ways of generating stronger passwords. For this we can use trivial encryption methods. Another way of making the file or folder secure is by providing with an option of hiding the file or the folder.

PROBLEM STATEMENT

As we all are from the cyber security domain we feel obliged to contribute to the same domain. Hence we were highly motivated to work on a concept concerned with the field of cyber security. “Folder and File Protection” is a topic of high conflict.

Folders are used to organize information. In the DOS and UNIX world, folders are called directories. Folder locking software is a security tool.

Locking folders effectively protects you from malicious programs, such as viruses, worms, and Trojans. Locking is the best way to guarantee that nobody accidentally or intentionally gets access to your financial, health, private, and confidential information. This software acts like a shield. And because locking your folders makes them inaccessible, they cannot be deleted, damaged, or harmed in any other way.

OBJECTIVE

The objective of this project is to create a folder locker to maintain the Confidentiality and integrity of data present inside the folder in the form of text files and encrypt and decrypt confidential files using RSA Algorithm.

This software to act like a shield and protect system files and folders. And because locking your folders makes them inaccessible, they cannot be deleted, damaged, or harmed in any other way.

LITERATURE REVIEW

As this is our first minor project we haven't worked on any formal project but we have implemented some Cryptographic algorithms in c language, they are as follows :-

a.) Caesar Cipher

The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials. Thus to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down. [7]

The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1, ..., Z = 25. Encryption of a letter by a shift n can be described mathematically as.

FOR ENCRYPTION:- $E(x) = (x+n) \bmod 26$
FOR DECRYPTION:- $D(x) = (x-n) \bmod 26$

b.) Vignere Cipher

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vignere Table. [5]

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

For encryption:- First a keyword is decided according to which the alphabets in the plaintext are to be shifted. We pair each letter of the plaintext with the corresponding Letter of the keyword.

For decryption:- Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VIGNERE TABLE

C) HASHING

Hashing is generating a value or values from a string of text using a mathematical function. Hashing is one way to enable security during the process of message transmission when the message is intended for a particular recipient only. A formula generates the hash, which helps to protect the security of the transmission against tampering. Hashing is also a method of sorting key values in a database table in an efficient manner.[6]

Hash Table and Hash Function

Earlier when this concept introduced programmers used to create “Direct address table”. Direct address table means, when we have “n” number of unique keys we create an array of length “n” and insert element “i” at ith index of the array. That array is called **Hash Table**. But due to this method even we have 10 elements of each range 1 lack, we should create table of size 1 lack for only 10 elements. Which is going to be waste of memory.

To avoid this problem we fix the size of hash table (array) and map our elements into that table using a function, called **Hash function**. This function decides where to put a given element into that table. If we want to search also first apply hash function decide whether the element present in hash table or not

METHODOLOGY

In this section we describe a way to solve the problem of lack of security measures being provided in operating systems such as Microsoft. Through this we customize the way we secure our valuable assets. We will be applying one of the most secure cryptographic that is RSA.

RSA(Rivest,Shamir, Adelman) cryptographic algorithm:

RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. **Public Key** and **Private Key**. As the name describes that the Public Key is given to everyone and Private key is kept private.[1]

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. [4]

Following calculations are necessary for implementing RSA Algorithm:-

a.) For generating Public Key -

For this we will select two prime numbers, P and Q.

The first part of the public key will be calculated as : $n=P*Q$

For instance, let us assume $P=53$ and $Q=59$. Hence $n = 53*59=3127$

A small exponent e is also required. The constraint for choosing e is as follows:-

- 1.)It should be an integer
- 2.)It shouldn't be a factor of n
- 3.) $1 < e < \text{function of } (n)$

b.) For generating Private key-

Now we need to calculate function(n): Such

that $\text{function}(n) = (P-1)(Q-1)$

so, $\text{function}(n)=3016$

Calculating Private key, d:
 $d = (k * \phi(n) + 1) / e$ for some integer k for k
=2, value of d is 2011.

-

We will encrypt the text by first converting all the to their Ascii values and then we will apply the following formula, $f^e \bmod n$, where f is the ascii value of the alphabet to be encrypted.

To decrypt the encrypted alphabet, we will use the following formula, $c^d \bmod n$, where c is the encrypted alphabet. Our motive to use cipher algorithms is to generate stronger passwords. We can create our own unique cipher algorithms. For example we can first apply a monoalphabetic cipher to our text before using RSA. This generates a higher level of password entropy. [7]

FOLDER LOCKING

In computing, `cacls` and its replacement, `icacls`, are Microsoft Windows native command line utilities capable of displaying and modifying the security descriptors on folders and files. An access control list is a list of permissions for securable object, such as a file or folder, that controls who can access it.[8]

To lock your file or folder type `cacls "File Path" [/t] /e /p everyone:n`. The `/t` is used to lock all folders and files within a folder. My file is located at `C:\Users\Michael\Desktop\Cool\testdoc.txt`, so I will type `cacls "C:\Users\Michael\Desktop\Cool\testdoc.txt" /e /p everyone:n`. [2]

`har mych2[]="\" /t /e /p everyone:n";` -To lock the folder

`har mych2[]="\" /t /e /p everyone:f";` -To Unlock the folder

Software Requirements:-

- 1.) Microsoft Visual Studio
- 2.) GCC(Gnu C Compiler)

Hardware Requirements:- 1.)

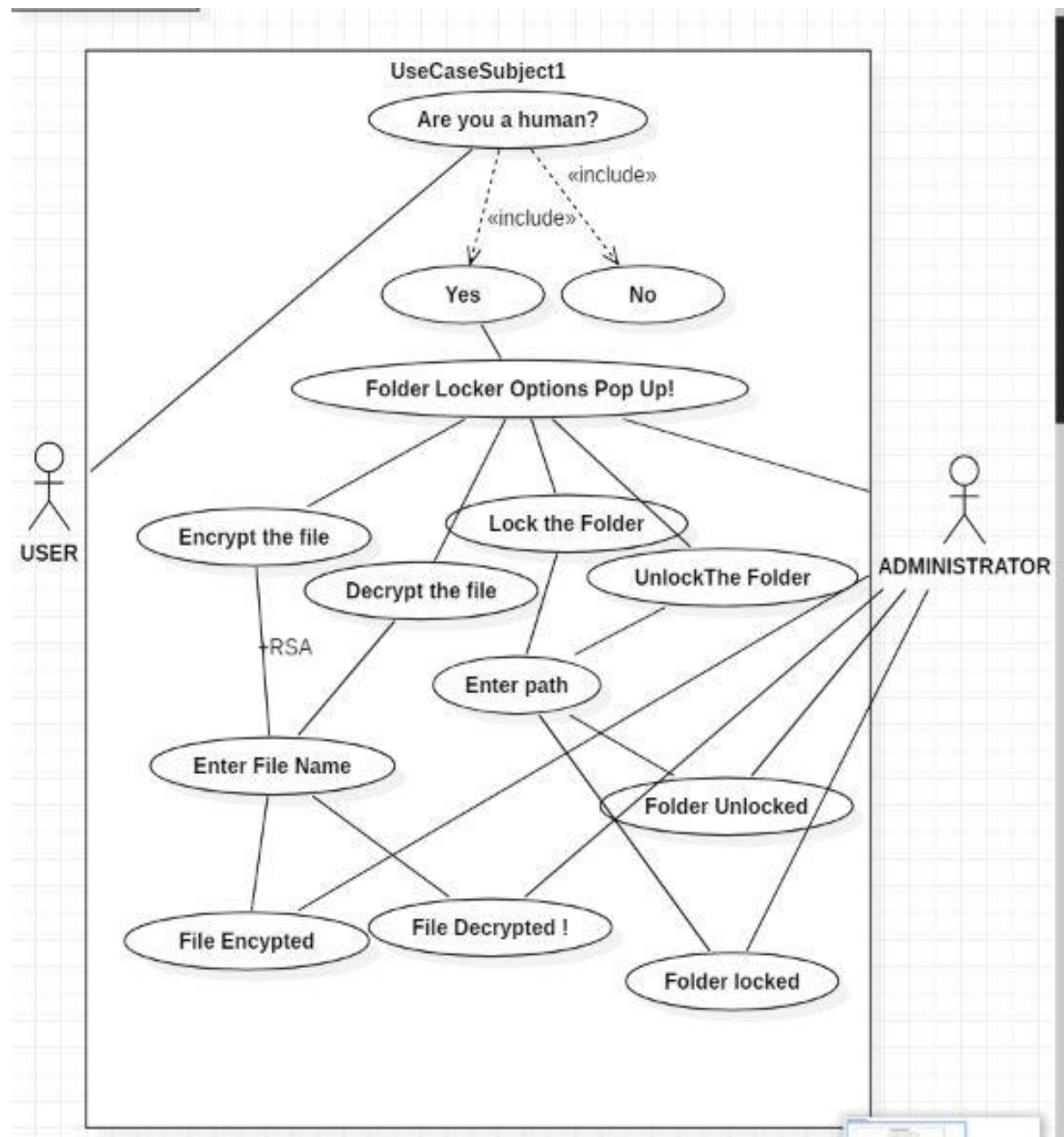
2Ghz Processor

2.) 4Gb Ram

Programming Language used:-

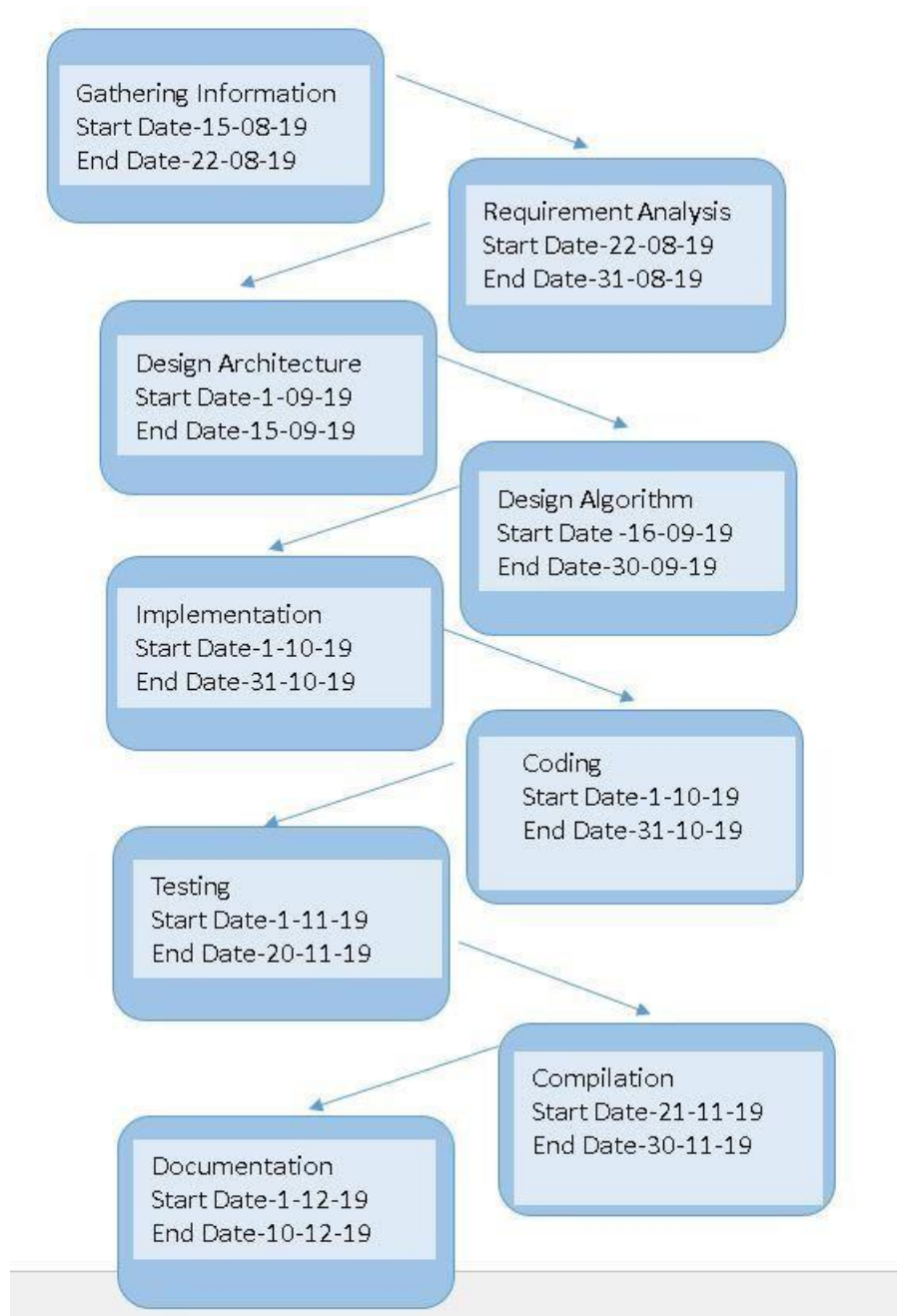
C

USE CASE



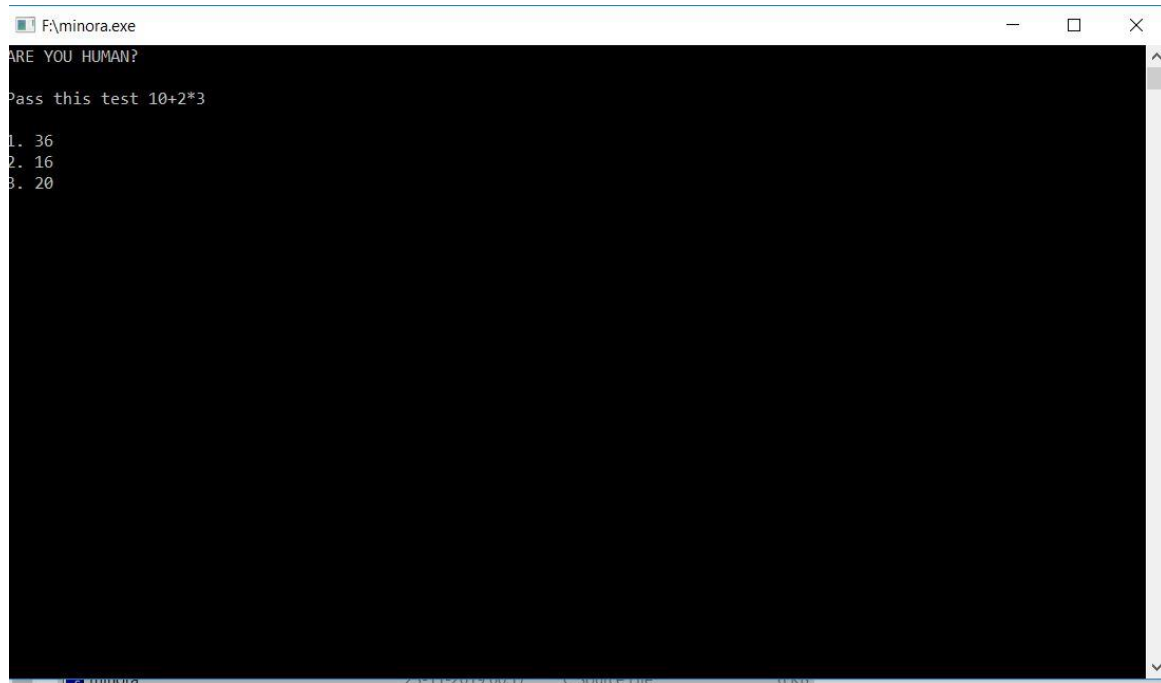
PERT CHART

The following PERT chart displays our month wise plan:-



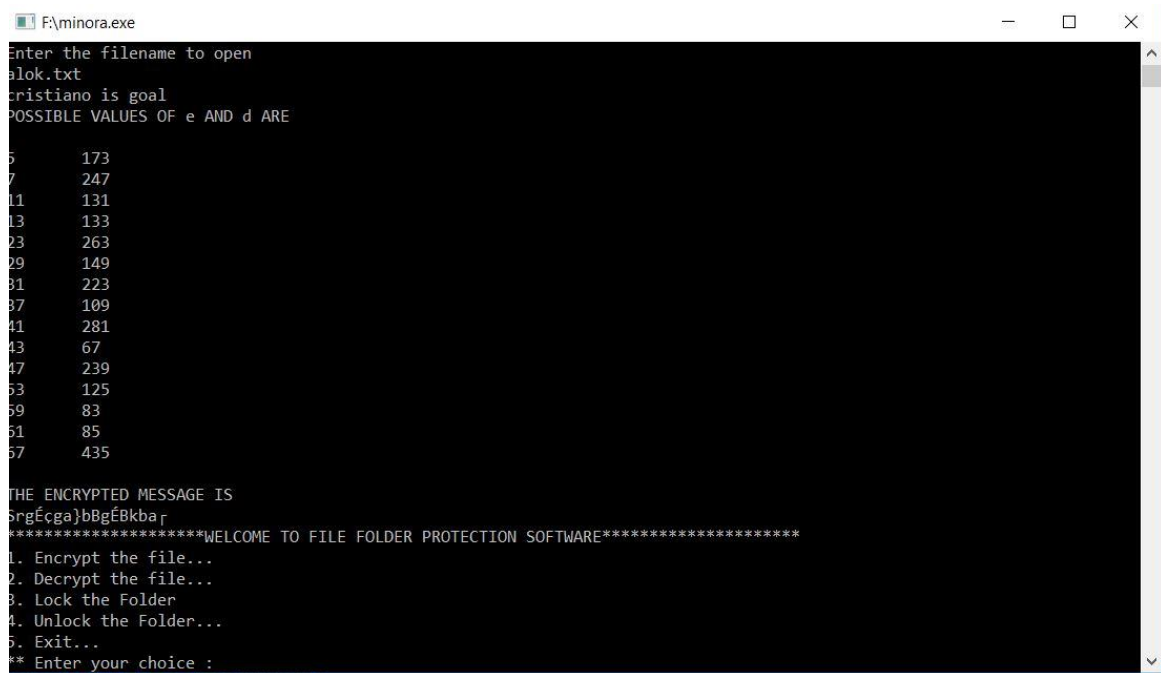
RESULT

RECOGNIZING HUMAN



```
F:\minor.exe
ARE YOU HUMAN?
Pass this test 10+2*3
1. 36
2. 16
3. 20
```

RSA ALGORITHM IMPLEMENTATION



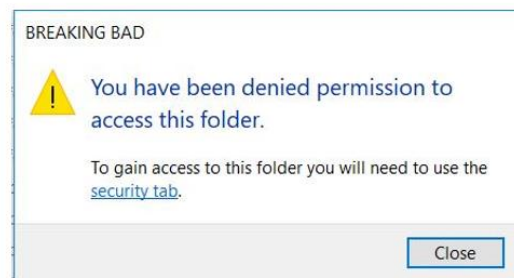
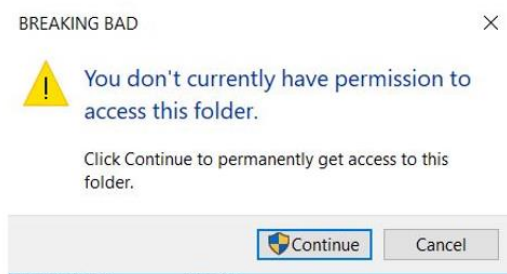
```
F:\minor.exe
Enter the filename to open
alok.txt
cristiano is goal
POSSIBLE VALUES OF e AND d ARE
5      173
7      247
11     131
13     133
23     263
29     149
31     223
37     109
41     281
43     67
47     239
53     125
59     83
61     85
67     435

THE ENCRYPTED MESSAGE IS
SrgEcga}bBgEBkba r
*****WELCOME TO FILE FOLDER PROTECTION SOFTWARE*****
1. Encrypt the file...
2. Decrypt the file...
3. Lock the Folder
4. Unlock the Folder...
5. Exit...
** Enter your choice :
```

ENCRYPTING THE FILE

```
F:\minora.exe
ARE YOU HUMAN?
Pass this test 10+2*3
1. 36
2. 16
3. 20
2
*****WELCOME TO FILE FOLDER PROTECTION SOFTWARE*****
1. Encrypt the file...
2. Decrypt the file...
3. Hide the Folder
4. Unhide the Folder...
5. Exit...
** Enter your choice :1
ENTER FIRST PRIME NUMBER
17
ENTER SECOND PRIME NUMBER
19
Enter the filename to open
test.txt
```

FOLDER PROTECTION



CONCLUSION

We have successfully implemented folder locker and encrypted and decrypted the Files using RSA Algorithm.

REFERENCES

1. <https://ieeexplore.ieee.org/abstract/document/6021216>
2. <https://www.instructables.com/id/How-to-Lock-a-Folder-in-Windows-Using-Cmd/>
3. <https://scialert.net/fulltextmobile/?doi=itj.2013.1818.1824>
4. *Shireen Nisha, Mohammed Farik , RSA Public Key Cryptography Algorithm – A Review , 07, JULY 2017*
5. *Aized Amin Soofi,Irfan Riaz, Umair Rasheed, An Enhanced Vigenere Cipher For Data Security, 03, MARCH 2016*
6. *Rajeev Sobti, G.Geetha, Cryptographic Hash Functions: A Review, March 2012*
7. *Tonni Limbong, Testing the Classic Caesar Cipher Cryptography using of Matlab, 02, February-2017 .*
8. <https://en.m.wikipedia.org/wiki/Cacls>

APPENDIX

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
#include<process.h>
#include<conio.h>

int x, y, n, t, i, flag;
long int e[10000], d[10000], temp[10000], j, m[10000], en[10000];
char msg[10000];
char filename[100];
int prime(long int);
//function for storing the key
void encryption_key();
long int cd(long int);
//function for encrypting the text file
void encrypt();
//function for decrypting the text file
void decrypt();
void create(char [],long);

void prim();
//Function for detecting human
int selection();
void main()
{
    selection();
}

int main12()
{
    int choice=0;
    char mych[]="cacls \\";
    char mych2[]="\ /t /e /p everyone:n";
    char mych3[]="\ /t /e /p everyone:f";
    char file[100],temp;
    while(choice!=5)
    {
```

```

printf("\n*****WELCOME TO FILE FOLDER PROTECTION
SOFTWARE*****\n");
printf("1. Encrypt the file...\n");
printf("2. Decrypt the file...\n");
printf("3. Lock the Folder\n");
printf("4. Unlock the Folder...\n");
printf("5. Exit...\n");
printf("** Enter your choice :");
scanf("%d",&choice);

switch(choice)
{
    case 1:
        encrypt();
        break;
    case 2:
        decrypt();break;
    case 3:
        printf("Enter the filename");
        scanf("%c",&temp);
        fgets(file,100,stdin);
        size_t len = strlen(file);
        if (len > 0 && file[len-1] == '\n') {
            file[--len] = '\0';
        }
        strcat(myh,file);
        strcat(myh,myh2);
        puts(myh);
        system(myh);
        break;
    case 4:
        printf("Enter the filename");
        scanf("%c",&temp);
        fgets(file,100,stdin);
        size_t lene = strlen(file);
        if (lene > 0 && file[lene-1] == '\n') {
            file[--lene] = '\0';
        }
        strcat(myh,file);
        strcat(myh,myh3);
        puts(myh);
        system(myh);
        break;
    case 5:

```

```

        exit(0);
    default:
        printf("\n Invalid choice !!!");
    }
}
return 0;
}

void prim()
{
    printf("\nENTER FIRST PRIME NUMBER\n");
    scanf("%d", &x);
    flag = prime(x);
    if(flag == 0)
    {
        printf("\nINVALID INPUT\n");
        exit(0);
    }
    printf("\nENTER SECOND PRIME NUMBER\n");
    scanf("%d", &y);
    flag = prime(y);
    if(flag == 0 || x == y)
    {
        printf("\nINVALID INPUT\n");
        exit(0);
    }
    FILE *fptr;
    char c;
    printf("Enter the filename to open \n");
    scanf("%s", filename);
    // Open file
    fptr = fopen(filename, "r");
    if (fptr == NULL)
    {
        printf("Cannot open file \n");
        exit(0);
    }

    // Read contents from file
    c = fgetc(fptr);
int k=0;
    while (c != EOF)
    {
        msg[k]=c;
        k=k+1;

```



```

        c = fgetc(fptr);
    }
    msg[k]='\0';
    printf("%s",msg);
    fclose(fptr);
    for(i = 0; msg[i] != '\0'; i++)
m[i] = msg[i];

n = x * y;
t = (x-1) * (y-1);
encryption_key();
printf("\nPOSSIBLE VALUES OF e AND d ARE\n");
for(i = 0; i < j-1; i++)
    printf("\n%ld\t%ld", e[i], d[i]);
}
int prime(long int pr)
{
    int i;
    j = sqrt(pr);
    for(i = 2; i <= j; i++)
    {
        if(pr % i == 0)
            return 0;
    }
    return 1;
}

//function to generate encryption key
void encryption_key()
{
    int k;
    k = 0;
    for(i = 2; i < t; i++)
    {
        if(t % i == 0)
            continue;
        flag = prime(i);
        if(flag == 1 && i != x && i != y)
        {
            e[k] = i;
            flag = cd(e[k]);
            if(flag > 0)
            {
                d[k] = flag;

```

```

        k++;
    }
    if(k == 99)
        break;
    }
}
}
long int cd(long int a)
{
    long int k = 1;
    while(1)
    {
        k = k + t;
        if(k % a == 0)
            return(k / a);
    }
}

//function to encrypt the message
void encrypt()
{
    prim();
    FILE *fptr2;
    long int pt, ct, key = e[0], k, len;
    i = 0;
    printf("%d",key);
    len = strlen(msg);
    while(i != len)
    {
        pt = m[i];
        pt = pt - 96;
        k = 1;
        for(j = 0; j < key; j++)
        {
            k = k * pt;
            k = k % n;
        }
        temp[i] = k;
        ct = k + 96;
        en[i] = ct;
        i++;
    }
    en[i] = -1;
    fptr2 = fopen(filename, "w");

```

```

if (fptr2 == NULL)
{
    printf("Cannot open file %s \n", filename);
    exit(0);
}
printf("\n\nTHE ENCRYPTED MESSAGE IS\n");
for(i = 0; en[i] != -1; i++)
{
    fputc(en[i], fptr2);
    printf("%c", en[i]);
}
fclose(fptr2);
create(filename,d[0]);
}

```

//function to decrypt the message

```
void decrypt()
```

```

{
    FILE *fptr3;
    long int pt, ct,key,k;
    i = 0;

```

```

printf("\nEnter the key please");
scanf("%ld",&key);

```

```

while(en[i] != -1)
{
    ct = temp[i];
    k = 1;
    for(j = 0; j < key; j++)
    {
        k = k * ct;
        k = k % n;
    }
    pt = k + 96;
    m[i] = pt;
    i++;
}
m[i] = -1;
printf("\n\nTHE DECRYPTED MESSAGE IS\n");
fptr3 = fopen(filename, "w");
if (fptr3 == NULL)
{
    printf("Cannot open file %s \n", filename);

```

```

        exit(0);
    }

    for(i = 0; m[i] != -1; i++)
    {
        fputc(m[i], fptr3);
        printf("%c", m[i]);
    }
    fclose(fptr3);
    printf("\n");
}

void create(char file[100],long k)
{
    FILE *fptr;
    fptr = fopen("key.txt", "a");
    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
    fprintf(fptr,"%s %ld\n", file,k);
    fclose(fptr);
}

int selection()
{
    int i, chc;
    printf("ARE YOU HUMAN?");
    printf("\n\n");
    printf("Pass this test 10+2*3");
    printf("\n\n");
    printf("1. 36");
    printf("\n");
    printf("2. 16");
    printf("\n");
    printf("3. 20");
    printf("\n");
    scanf("%d",&chc);
    if(chc==2)
    {
        main12();
    }
}

```