

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Bull Temple Road, Basavanagudi, Bangalore-560019



A project report on

“ENCRYPTION/DECRYPTION CIPHER”

Submitted in partial fulfilment of the requirements for the
award of degree

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

By

CHIRANJEEVI NAYAK B, 1BM20IS402

SRINIVAS BA, 1BM20IS410

SUSHMITHA R, 1BM20IS412

Under the guidance of

Chandrakala G Raju, Assistant professor

Department of Information Science and Engineering

2020-21

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)



Department of Information Science and Engineering

CERTIFICATE

This is to certify that the project entitled “*ENCRYPTION/DECRYPTION CIPHER*” is a bona-fide work carried out by **CHIRANJEEVI NAYAK B (1BM20IS402), SRINIVAS BA (1BM20IS410), SUSHMITHA R (1BM20IS412)** in partial fulfillment for the award of degree of Bachelor of Engineering in **Information Science and Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Signature of the Faculty

Signature of the HOD

LIST OF CONTENT

SLNO	CONTENT	PAGE NO
1	Abstract	1
2	Introduction	2
3	Problem statement	3
4	Tools used	3
5	Objectives	3
6	Methodology	4
7	Source Code	5-34
8	Output snapshots	35
9	Cheat Sheet	36-37
10	Limitations of project	38
11	Learning Outcome	38
12	Conclusion	38
13	References	39

ABSTRACT

Data security is a major issue which we are facing today in this digital world of communication. As we know that today hackers are almost at every corner in search of our useful data which can be hacked by them for their different purposes. Even the risk gets doubled when come to the data of any country's government. So, a system or terminology is must required to make that data safe forever by any means during communication.

Data protection can be accomplished by changing the original data by any means to some other un useful data so that if someone gets that data then also it must remain in un useful bits. This process can be achieved by Encrypting that data by some means of algorithms which are known to the sender and the similar Decryption algorithms to be known to only the desired receiver so that it can convert that encrypted data back to the user understandable form. Today as it is a need to develop such kind of applications which performs the specified task but along with it should be very much user friendly so that no special skills need to be required to learn in order to use that application or project.

In the application the user has to select either wants to send something by encrypting or wants to receive by decrypting. If it wants to send then it have to select source file previously designed or type some message which is to Encrypt and then transfer. Whereas on the receiver side again the receiver have to select the file which is to be received from the sender along with a decryption key to decrypt that message. Decryption key can be selected either manually if told by the sender or selecting key sent by the sender along with the encrypted data to avoid further delay in processing the message.

INTRODUCTION

This project has made us understand the concepts and implementations of package in a better way. The key idea behind this project is to build a portable cipher for daily and general use. Most of the Ciphers available in industry are based on some Algorithms (RSA Algo- Most famous one). They somehow provide some strong and secure method of data encryption/decryption. But this Cipher built by us shows uniqueness in itself. We have built this Cipher without any algo or logic, which makes it impossible to decode by cyber attackers. One can decode it by using the Cheat Sheet only owned by the developer itself.

Today as it is a need to develop such kind of applications which performs the specified task but along with it should be very much user friendly so that no special skills need to be required to learn in order to use that application or project.

We have implemented this Cipher in an Applet. One can build a jar file out of it, and can implement this cipher in messaging and communication systems (Web Apps) to hide their texts and privacy.

PROBLEM STATEMENT

- To Design a program that encrypts and decrypts the word or sentence, given input by the user.
- At present there are many encryption and decryption, especially in the communication system provided in a variety of application.
- An Encryption/Decryption program consider the problem of encrypting a sequence of characters into another sequence of characters.
- For Decrypting, we can retrieve the characters back by the inverse process

TOOLS USED

1. Java software kit (JDK/ JRE - Ver._1.8)
2. Text Editor (Notepad/ Eclipse IDE)
3. MS-DOS

OBJECTIVES

- The main objective of this project is to build a portable program for Encryption and Decryption using Packages.
- Users can implement these packages into their program to run Cipher. Here, we have shown a glimpse of its implementation in a fully functional applet programming.

METHODOLOGY

- There are two source files kept in the package serving different purposes; namely:
 1. Cipher.java: containing parameterized methods encrypt() and decrypt(). Its class file is implemented in the Applet programming.
 2. CipherImplement.java: containing non-parameterized methods encrypt() and decrypt(), both of which are taking inputs of String through Scanner() method.
- Structure of the package:

PBL_TESTING\ -----Base Directory
 Com->pbl->cipher -----Package name
- Two Test Files, namely:
 1. expack.java: Implementing the CipherImplement.class file. Program is tested on MS-DOS after compiling and interpreting the source file.
 2. CipherApplet.java: Implementing the Cipher.class file. Program is tested on an Applet. Two buttons are given to encrypt the message thrown in text box, and then another button to decrypt that encrypt message to receive the original message again.
- This program is capable to encrypt/ decrypt a single word or a complete sentence as per requirement.
- To run DOS Test file:

>> java Testcipher
- To run Applet program from DOS (Applet viewer):

>> appletviewer CipherApplet.java

SOURCE CODE

1.Cipher.java

```
package com.pbl.cipher;

import java.util.Scanner;

public class Cipher {

    String ins;

    char[] outs;

    char get, put, at, on;

    //Scanner KB=new Scanner(System.in);

    public void match(char at){

        this.at=at;

        if(get=='A'){

            get='B';

        }

        else if(get=='B'){

            get='N';

        }

        else if(get=='C'){
```



```

        get='Z';

    }

    else if(get=='D'){

        get='2';

    }

    else if(get=='E'){

        get='K';

    }

    else if(get=='F'){

        get='A';

    }

    else if(get=='G'){

        get='9';

    }

    else if(get=='H'){

        get='Q';

    }

    else if(get=='T'){

        get='J';

    }

    else if(get=='J'){

        get='T';
    
```

```

    }

    else if(get=='K'){

        get='D';

    }

    else if(get=='L'){

        get='O';

    }

    else if(get=='M'){

        get='U';

    }

    else if(get=='N'){

        get='I';

    }

    else if(get=='O'){

        get='8';

    }

    else if(get=='P'){

        get='C';

    }

    else if(get=='Q'){

        get='H';

    }

```

```

else if(get=='R'){

    get='5';

}

else if(get=='S'){

    get='7';

}

else if(get=='T'){

    get='F';

}

else if(get=='U'){

    get='M';

}

else if(get=='V'){

    get='0';

}

else if(get=='W'){

    get='6';

}

else if(get=='X'){

    get='E';

}

else if(get=='Y'){

```

```

        get='P';
    }

    else if(get=='Z'){

        get='T';

    }

    else if(get=='0'){

        get='V';

    }

    else if(get=='1'){

        get='G';

    }

    else if(get=='2'){

        get='4';

    }

    else if(get=='3'){

        get='L';

    }

    else if(get=='4'){

        get='S';

    }

    else if(get=='5'){

```

```

        get='W';
    }

    else if(get=='6'){

        get='3';

    }

    else if(get=='7'){

        get='Y';

    }

    else if(get=='8'){

        get='R';

    }

    else if(get=='9'){

        get='X';

    }

}

public void scratch(char on){

    this.on=on;

    if(put=='A'){

        put='F';

    }

    else if(put=='B'){

        put='A';
    
```

```

    }

    else if(put=='C'){

        put='P';

    }

    else if(put=='D'){

        put='K';

    }

    else if(put=='E'){

        put='X';

    }

    else if(put=='F'){

        put='T';

    }

    else if(put=='G'){

        put='I';

    }

    else if(put=='H'){

        put='Q';

    }

    else if(put=='I'){

        put='J';

    }

```

```

else if(put=='J'){

    put='I';

}

else if(put=='K'){

    put='E';

}

else if(put=='L'){

    put='3';

}

else if(put=='M'){

    put='U';

}

else if(put=='N'){

    put='B';

}

else if(put=='O'){

    put='L';

}

else if(put=='P'){

    put='Y';

}

else if(put=='Q'){

```

```

        put='H';
    }

    else if(put=='R'){

        put='8';

    }

    else if(put=='S'){

        put='4';

    }

    else if(put=='T'){

        put='Z';

    }

    else if(put=='U'){

        put='M';

    }

    else if(put=='V'){

        put='0';

    }

    else if(put=='W'){

        put='5';

    }

    else if(put=='X'){

        put='9';
    
```



```

    }

    else if(put=='Y'){

        put='7';

    }

    else if(put=='Z'){

        put='C';

    }

    else if(put=='0'){

        put='V';

    }

    else if(put=='1'){

        put='N';

    }

    else if(put=='2'){

        put='D';

    }

    else if(put=='3'){

        put='6';

    }

    else if(put=='4'){

        put='2';

    }

```

```

        else if(put=='5'){

            put='R';

        }

        else if(put=='6'){

            put='W';

        }

        else if(put=='7'){

            put='S';

        }

        else if(put=='8'){

            put='O';

        }

        else if(put=='9'){

            put='G';

        }

    }

}

public String encrypt(String s){

    //System.out.println("\n\t\t\t enter ur input:");

    //ins= KB.next();

```

```

        s=s.toUpperCase();

        outs=new char[s.length()];

        for(int i=0; i<s.length();i++){

            get=s.charAt(i);

            match(get);

            outs[i]=get;

        }

        //System.out.print("*****!!! ");

        //System.out.print(outs);

        //System.out.println("  !!!*****");

        String str=String.valueOf(outs);

    return str;

    }

    public String decrypt(String s){

        //System.out.println("\n\n\n\t\t\t enter ur input:\t\t\t");

        //ins= KB.next();

        outs=new char[s.length()];

        for(int i=0; i<s.length();i++){

            put=s.charAt(i);

            scratch(put);

            outs[i]=put;

        }
    
```

```

        //System.out.print("*****!!! ");

        //System.out.print(outs);

        //System.out.println("  !!!*****");

        String str=new String(outs);

        return str;

    }

}

```

2.CipherImplement.java

```

package com.pbl.cipher;

import java.util.Scanner;

public class CipherImplement {

    String ins;

    char[] outs;

    char get, put, at, on;

    Scanner KB=new Scanner(System.in);

    public void match(char at){

        this.at=at;

        if(get=='A'){

            get='B';

```

```

    }

    else if(get=='B'){

        get='N';

    }

    else if(get=='C'){

        get='Z';

    }

    else if(get=='D'){

        get='2';

    }

    else if(get=='E'){

        get='K';

    }

    else if(get=='F'){

        get='A';

    }

    else if(get=='G'){

        get='9';

    }

    else if(get=='H'){

        get='Q';

    }

```

```

else if(get=='I'){

    get='J';

}

else if(get=='J'){

    get='I';

}

else if(get=='K'){

    get='D';

}

else if(get=='L'){

    get='O';

}

else if(get=='M'){

    get='U';

}

else if(get=='N'){

    get='1';

}

else if(get=='O'){

    get='8';

}

else if(get=='P'){

```

```

        get='C';
    }
    else if(get=='Q'){
        get='H';
    }
    else if(get=='R'){
        get='5';
    }
    else if(get=='S'){
        get='7';
    }
    else if(get=='T'){
        get='F';
    }
    else if(get=='U'){
        get='M';
    }
    else if(get=='V'){
        get='0';
    }
    else if(get=='W'){
        get='6';
    }

```

```

    }

    else if(get=='X'){

        get='E';

    }

    else if(get=='Y'){

        get='P';

    }

    else if(get=='Z'){

        get='T';

    }

    else if(get=='0'){

        get='V';

    }

    else if(get=='1'){

        get='G';

    }

    else if(get=='2'){

        get='4';

    }

    else if(get=='3'){

        get='L';

    }

```



```

        else if(get=='4'){

            get='S';

        }

        else if(get=='5'){

            get='W';

        }

        else if(get=='6'){

            get='3';

        }

        else if(get=='7'){

            get='Y';

        }

        else if(get=='8'){

            get='R';

        }

        else if(get=='9'){

            get='X';

        }

    }

    public void scratch(char on){

        this.on=on;
    }

```

```

if(put=='A'){

    put='F';

}

else if(put=='B'){

    put='A';

}

else if(put=='C'){

    put='P';

}

else if(put=='D'){

    put='K';

}

else if(put=='E'){

    put='X';

}

else if(put=='F'){

    put='T';

}

else if(put=='G'){

    put='I';

}

else if(put=='H'){

```

```

        put='Q';
    }
    else if(put=='I'){
        put='J';
    }
    else if(put=='J'){
        put='I';
    }
    else if(put=='K'){
        put='E';
    }
    else if(put=='L'){
        put='3';
    }
    else if(put=='M'){
        put='U';
    }
    else if(put=='N'){
        put='B';
    }
    else if(put=='O'){
        put='L';
    }

```

```

    }

    else if(put=='P'){

        put='Y';

    }

    else if(put=='Q'){

        put='H';

    }

    else if(put=='R'){

        put='8';

    }

    else if(put=='S'){

        put='4';

    }

    else if(put=='T'){

        put='Z';

    }

    else if(put=='U'){

        put='M';

    }

    else if(put=='V'){

        put='0';

    }

```

```

else if(put=='W'){
    put='5';
}
else if(put=='X'){
    put='9';
}
else if(put=='Y'){
    put='7';
}
else if(put=='Z'){
    put='C';
}
else if(put=='0'){
    put='V';
}
else if(put=='1'){
    put='N';
}
else if(put=='2'){
    put='D';
}
else if(put=='3'){

```

```

        put='6';
    }
    else if(put=='4'){
        put='2';
    }
    else if(put=='5'){
        put='R';
    }
    else if(put=='6'){
        put='W';
    }
    else if(put=='7'){
        put='S';
    }
    else if(put=='8'){
        put='O';
    }
    else if(put=='9'){
        put='G';
    }
}

```

```

public void encrypt(){

    System.out.println("\n\t\t\t Enter ur input for encryption:");

    ins= KB.next();

    ins=ins.toUpperCase();

    outs=new char[ins.length()];

    for(int i=0; i<ins.length();i++){

        get=ins.charAt(i);

        match(get);

        outs[i]=get;

    }

    System.out.print("*****!!! ");

    System.out.print(outs);

    System.out.println("  !!!*****");

    //String str=String.valueOf(outs);

    //return str;

}

public void decrypt(){

    System.out.println("\n\n\n\t\t\t Enter ur input for decryption:\t\t\t");

    ins= KB.next();

    ins=ins.toUpperCase();

```

```

        outs=new char[ins.length()];

        for(int i=0; i<ins.length();i++){

            put=ins.charAt(i);

            scratch(put);

            outs[i]=put;

        }

        System.out.print("*****!!! ");

        System.out.print(outs);

        System.out.println("  !!!*****");

        //String str=new String(outs);

        //return str;

    }

}

```


TEST FILES

3.CipherApplet.java

```
import com.pbl.cipher.Cipher;

import java.applet.Applet;

import java.awt.*;

import java.awt.event.*;

import java.awt.Color;

import java.awt.Graphics;

/*

<applet code="CipherApplet.class" height=200 width=550>

</applet>

*/

public class CipherApplet extends Applet implements ActionListener

{

    Image picture;

    TextField t1;

    TextField t2;

    TextField t3;

    Button e;

    Button d;

    Button c;
```

String s,e1,d1;

Cipher ob;

public void init()

{

 picture=getImage(getDocumentBase(),"images.jpg");

 t1=new TextField(20);

 t2=new TextField(20);

 t3=new TextField(20);

 ob=new Cipher();

 e=new Button("Encrypt");

 e.setBackground(Color.green);

 d=new Button("Decrypt");

 d.setBackground(Color.red);

 c=new Button(" Reset ");

 c.setBackground(Color.blue);

 add(t1);

 add(e);

 add(t2);

 add(d);

 add(t3);

```

        add(c);

        e.addActionListener(this);

        d.addActionListener(this);

        c.addActionListener(this);

    }

    public void paint(Graphics g)

    {

        setBackground(Color.black);

        g.drawImage(picture,530,100,this);

    }


    public void actionPerformed(ActionEvent ae)

    {

        if(ae.getSource()==e)

        {

            s=String.valueOf(t1.getText());

            e1=ob.encrypt(s);

            t2.setText(e1);

        }

        if(ae.getSource()==d)

        {

            d1=ob.decrypt(t2.getText());

```

```

        t3.setText(d1);
    }
    if(ae.getSource()==c)
    {
        t1.setText(" ");
        t2.setText("");
        t3.setText("");
    }
}
}
}

```

4.expack.java

```
import com.pbl.cipher.CipherImplement;

class Testcipher{

    public static void main(String[] args){

        CipherImplement ob=new CipherImplement();

        System.out.println("\n YOU MUST TRY TO ENCRYPT AND DECRYPT
THE CODE:\n");

        ob.encrypt();

        System.out.println("\n SUCCESSFULLY ENCRYPTED.. \n");

        ob.decrypt();

        System.out.println("\n SUCCESSFULLY DECRYPTED.. \n");

    }

}
```

OUTPUT (SNAPSHOTS)

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.1083]
(c) Microsoft Corporation. All rights reserved.

C:\Cipher-Decipher-using-Java-master>set path=C:\Program Files (x86)\Java\jdk1.8.0_161\bin
C:\Cipher-Decipher-using-Java-master>javac expack.java
C:\Cipher-Decipher-using-Java-master>java Testcipher

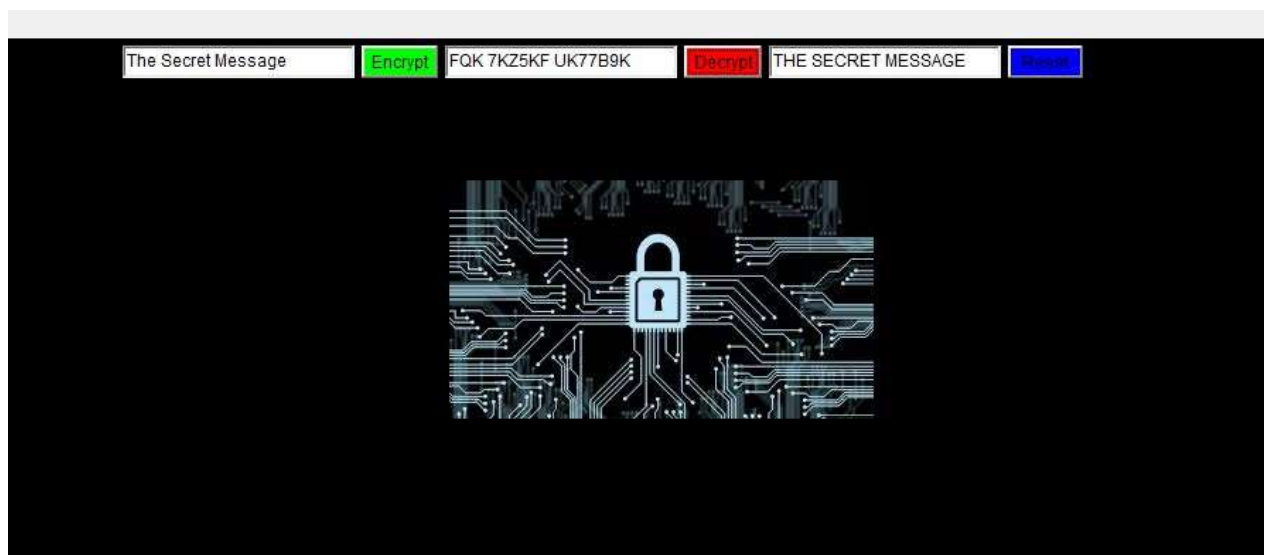
YOU MUST TRY TO ENCRYPT AND DECRYPT THE CODE:

                Enter ur input for encryption:
MESSAGE
*****!!!    UK77B9K    !!!*****

SUCCESSFULLY ENCRYPTED..

                Enter ur input for decryption:
UK77B9K
*****!!!    MESSAGE    !!!*****

SUCCESSFULLY DECRYPTED..
    
```



CHEAT SHEET

<u>Encryption</u>		<u>decryption</u>	
A	B	A	F
B	N	B	A
C	Z	C	P
D	2	D	K
E	K	E	X
F	A	F	T
G	9	G	1
H	Q	H	Q
I	J	I	J
J	I	J	I
K	D	K	E
L	O	L	3
M	U	M	U
N	1	N	B
O	8	O	L
P	C	P	Y
Q	H	Q	H

ENCIPHERMENT & DECRYPTER CIPHER

R	5	R	8
S	7	S	4
T	F	T	Z
U	M	U	M
V	0	V	0
W	6	W	5
X	E	X	9
Y	P	Y	7
Z	T	Z	C
0	V	0	V
1	G	1	N
2	4	2	D
3	L	3	6
4	S	4	2
5	W	5	R
6	3	6	W
7	Y	7	S
8	R	8	O
9	X	9	G

LIMITATIONS OF THIS PROJECT

We have restricted this project to some limitations:

- The Dos implementation does work for a complete Sentence input.
- Any further development in this project is impossible without developer's presence.
- Lower Case letters give wrong output in DOS implementation. (Case conversation is not applied here). Though, it works well in Applet Implementation.

LEARNING OUTCOME

Through this project based Learning methodology, we have actually got an opportunity to explore the subjects taught to us in our Academics. We have gained an experience to develop something totally functional out of raw academic stuffs.

This project can be further expanded to its next level. We can put into action for a chat messenger app, or any web application based on communication between two terminals. Additionally, we have gained the in depth concepts behind Cipher technologies.

CONCLUSION

Data security is an essential component of an organization in order to keep the information safe from various competitors. It helps to ensure the privacy of a user's personal information from others. Secured and timely transmission of data is always an important aspect for an organization. Strong encryption algorithms and optimized key management techniques always help in achieving confidentiality, authentication and integrity of data and reduce the overheads of the system. Used a technique encryption and decryption without any algo or logic to avoid unauthorized access of data. which makes it impossible to decode by cyber attackers. One can decode it by using the Cheat Sheet only owned by the developer itself.

REFERENCES

1. <http://www.javatpoint.com/package>
2. <http://www.dreamincode.net/forums/topic/85570-positioning-the-buttons-in-an-applet/>
3. <http://stackoverflow.com/>
4. https://www.tutorialspoint.com/awt/awt_event_handling.htm