JAVA ENCRYPTOR

Final Project Report

Submitted in fulfillment of the 1 month winter INTERNSHIP

Bharti Airtel LTD, Gurgaon

UMANG CHAUDHARY – 15BIT0074

B.Tech – Information Technology (2nd year)

VIT Vellore, Tamil Nadu

Under the guidance of

Mr. NARESH PATEL	**Mr. ATUL ANAND**	**Mr. ARUN GUPTA**
(Mentor) IBM Private LTD.	**(IT Head) IBM Private LTD.**	**(IT Head) Bharti Airtel**

December, 2016

airtel

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

ACKNOWLEDGEMENT

It gives me a great pleasure to present before you "**JAVA ENCRYPTOR**". The overwhelming response of my friends and the keen interest shown by my learned teachers, promoted me to make a meaningful, complete, compact and comprehensive project.

It gives me immense pleasure to have an opportunity to express my heartiest gratitude to Senior Software Engineers at BHARTI AIRTEL who guided me in the completion of the project on Encryption/Decryption desktop application.

The cooperation of my Mentor Mr. NARESH PATEL was excellent. It was the coordinated efforts of him only that the project was able to complete well within the time. I am also thankful to the company for lending me the support and providing me with all the required facilities.

- **UMANG CHAUDHARY**

CONTENTS

CHAPTER 1

ABOUT THE ORGANISATION: BHARTI AIRTEL LTD

Bharti Airtel Limited is an Indian global telecommunications Services Company headquartered in New Delhi, India. It operates in 20 countries across South Asia, Africa, and the Channel Islands. Airtel provides GSM, 3G and 4G LTE mobile services, fixed line broadband and voice services depending upon the country of operation. It is the largest mobile network operator in India and thethird largest in the world with 325 million subscribers. Airtel was named India's second most valuable brand in the first ever Brandz ranking by Millward Brown and WPP plc.

Airtel is the one of the largest mobile operator in the world in terms of subscriber base and has a commercial presence in 20 countries and the Channel Islands, Baysquare Technology developed a Settlement and Reconciliation Tool (SRT) to reconcile from various data streams. The system was developed to match the calls being captured by the network elements and the calls getting rated, i.e. ensuring that operator is billing all calls its serves and also it is paying out to other operators the correct billing amounts.

Its area of operations includes:

- The Indian Subcontinent:
  - Airtel India, in India
  - Airtel Sri Lanka, in Sri Lanka
  - Airtel Bangladesh, in Bangladesh
  - Airtel Africa, which operates in 17 African countries:
  - Burkina Faso, Chad, Democratic Republic of the Congo, Republic of the Congo, Gabon, Ghana, Kenya, Madagascar, Malawi, Niger, Nigeria, Rwanda, Seychelles , Sierra Leone, Tanzania, Uganda and Zambia.
- The British Crown Dependency islands of Jersey and Guernsey, under the brand name Airtel-Vodafone, through an agreement with Vodafone.

Telemedia

Under the Telemedia segment, Airtel provides broadband internet access through DSL, internet leased lines as well as MPLS (multiprotocol label switching) solutions, as well as IPTV and fixed line telephone services. Until 18 September 2004, Bharti provided fixed line telephony and broadband services under the *Touchtel* brand. Bharti now provides all telecom services including fixed line services under a common brand *airtel*. As of September 2012, Airtel provides

Telemedia services to 3.3 million customers in 87 cities. As on 30 November 2012, Airtel had 1.39 million broadband subscribers.

Airtel Broadband provides broadband and IPTV services. Airtel provides both capped as well as unlimited download plans. However, Airtel's unlimited plans are subject to free usage policy (FUP), which reduces speed after the customer crosses a certain data usage limit. In most of the plans, Airtel provides only 512kbit/s beyond FUP, which is lower than the TRAI specified limit of half the subscriber's original speed. The maximum speed available for home users is 16Mbit/s.

In May 2012, Airtel Broadband and some other Indian ISPs temporarily blocked file sharing websites such as vimeo.com megavideo.com, thepiratebay.se, etc. with out giving any legal information to the customers.

Digital television

The Digital television business provides Direct-to-Home (DTH) TV services across India under the brand name Airtel digital TV. It started services on 9 October 2008 and had about 7.9 million customers at the end of December 2012.

Enterprise

The Enterprise business provides end-to-end telecom solutions to corporate customers and national and international long-distance services to telcos through its nationwide fibre optic backbone, last mile connectivity in fixed-line and mobile circles, VSATs, ISP and international bandwidth access through the gateways and landing stations. It has two sections under it.

Mobile data service

The different services under mobile data are BlackBerry services, a web-enabled mobile email solution working on 'Push Technology', USB modem that helps in getting instant access to Internet and corporate applications, Airtel Data Card that gives the liberty to access the internet anytime, Easy Mail is a platform that provides access to personal/corporate e-mails independent of handset operating system and application services that shorten the queues at the billing section, off-load the pressure on the billing staff and bring convenience to the user.

Enterprise business solutions

There are two kind of solutions offered by Airtel. One is GPRS Based Solutions like mobile applications tools for enterprise, TrackMate, automatic meter reading solutions etc. and the other is SMS Based Solutions like interactive sms, bulk sms, inbound call center solutions.

Android-based tablet

Beetel Teletech Ltd., a unit of Bharti Enterprises Ltd., on 18 August launched a ₹9,999 ($220) 7-inch tablet in India based on Google Inc.'s Android operating system. The offering is intended to capitalise on the expected demand for cheap computing devices in the world's fastest-growing and second-largest mobile phone market.[39]

Domestic operations

Airtel is the largest provider of mobile telephony and second largest provider of fixed telephony in India, and is also a provider of broadband and subscription television services. It offers its telecom services under the "airtel" brand, and is headed by Sunil Bharti Mittal.

CHAPTER 2

ABOUT CRYPTOGRAPHY & ALGORITHMS

CRYPTOGRAPHY

Cryptography is the method of protecting secret information from unauthorized people by storing and transmitting information in an unreadable format.

Some encryption and decryption released cryptographic terms are

- Ciphertext
- Plaintext
- Decipher
- Encipher and
- Work factor

Some key cryptographic terms are

- Algorithms
- Cryptosystem
- Cryptology

Some more cryptographic terms are

- Key
- Key clustering
- Keyspace
- Data origin authentication and
- Entity authentication

A cipher is an algorithm that helps represent plaintext units or single letters in the form of ciphertext using arbitrary symbols or groups of symbols.

Encryption emerged as an art but was later used to protect secret information related to warfare, commerce, and government arenas. Encryption is the key process used in cryptography and it has become an integrated part of the computing world with the advent of the internet.

History of Cryptography

Cryptography has undergone many changes through the centuries according to advances in technology.

Table 1: Developments in Cryptography

Period	Development
2000 BC	In Egypt, hieroglyphics were used in inscriptions.
500-600 BC	Hebrews used the atbash method for encryption. In this method, each letter of the alphabet mapped to a different letter to hide the true meaning of a word.
487 BC	The Spartans used the scytale for encryption – messages were written on paper wrapped around a wooden rod. The paper was then unwrapped and sent. The recipient could read the message only by wrapping this paper on a rod of the same length and diameter.
100-44 BC	Julius Caesar used an encryption method similar to the atbash method. He shifted each letter of the alphabet by a fixed number of places to send encrypted messages.
1379	Gabrieli di Lavinde developed the nomenclator.
1466-1467	The first polyalphabetic cipher was invented, which was much stronger than the nomenclator.
1518	Johannes Trithemius invented a steganographic cipher in which each letter was represented as a word taken from a succession of columns.
1553	Giovan Batista Belaso introduced the use of a passphrase as the key for a repeated polyalphabetic cipher. In 1563, Giovanni Battista Porta introduced the digraphic cipher and classified ciphers as transposition, substitution, and symbol substitution.
1585	Blaise de Vigenere developed the polyalphabetic substitution cipher. William Frederick Friedman published a book on Crytography and is called the "Father of Modern Cryptography".
1623	Sir Francis Bacon described the biliteral cipher – known as 5-bit binary encoding. This advanced the steganographic cipher by using variation in type face to carry out encoding.
1917	Gilbert S. Vernam invented a polyalphabetic cipher machine that used a never repeating random key. In 1919, a rotor-based cipher machine was invented, which led

Table 1: Developments in Cryptography

Period	Development
	to the invention of the popular Enigma machine in 1923.
1919	A rotor-based cipher machine was invented, which led to the invention of the popular Enigma machine in 1923.
1929	Lester S. Hill used algebraic operations for encrypting blocks of plaintext.
1937	The Japanese Purple machine was invented, which used telephone stepping relays instead of rotors having totally different permutations at each step rather than related permutations of one rotor in different positions.
1970	The Lucifer cipher was developed, which was later modified to establish the Data Encryption Standard (DES).
1976	The idea of public key cryptography was introduced. Eventually, the RSA algorithm, International Data Encryption Algorithm (IDEA), RC2 and RC4 algorithms, and the Advanced Encryption Standard (AES) were developed.
Beyond 1976	The RSA algorithm, International Data Encryption Algorithm (IDEA), RC2 and RC4 algorithms, and the Advanced Encryption Standard (AES) were developed.

Cryptographic Key

According to the length of the cryptographic key, a certain amount of time and money is needed to break keys of different lengths.

Table 1: Comparison of Time and Money Needed to Break Different Length Keys

Length of Cryptographic Key/Cost	**40-bit**	**56-bit**	**64-bit**	**80-bit**	**128-bit**
$0.1 M	2 seconds	35 hours	1 year	70,000 years	10E19 years
$1.0 M	0.2 seconds	3.5 hours	37 days	7,000 years	10E18 years

Table 1: Comparison of Time and Money Needed to Break Different Length Keys

Length of Cryptographic Key/Cost	**40-bit**	**56-bit**	**64-bit**	**80-bit**	**128-bit**
$100 M	2 milliseconds	2 minutes	9 hours	70 years	10E16 years
$1.0 B	0.2 milliseconds	13 seconds	1 hour	7 years	10E15 years
$100 B	2 microseconds	1 second	32 seconds	24 days	10E13 years

- **NARESH PATEL**

(http://www.nareshpatel.in)

ALGORITHMS

1. **AES -** Advanced Encryption Standard is a symmetric encryption algorithm.
   The algorithm was developed by two Belgian cryptographers Joan Daemen and Vincent Rijmen.
   AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits.
   When you want to encrypt a confidential text into a decryptable format, for example when you need to send sensitive data in e-mail.
   The decryption of the encrypted text it is possible only if you know the right password.
   AES encryption is used by U.S. for securing sensitive but unclassified material, so we can say it is enough secure.

2. **Triple DES** - Triple DES is another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Stealth, you simply type in the entire 192-bit (24 character) key rather than entering each of the three keys individually. The Triple DES DLL then breaks the user-provided key into three subkeys, padding the keys if necessary so they are each 64 bits long. The procedure for encryption is exactly the same as regular DES, but it is repeated three times, hence the name Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

   Triple DES Modes

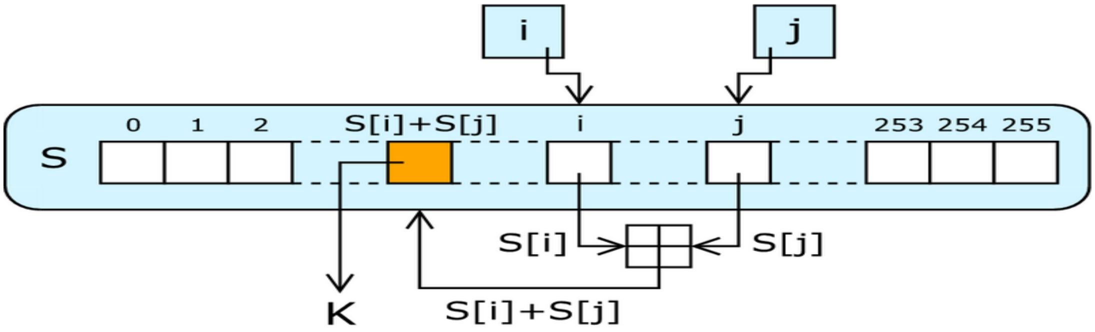
   Triple ECB (Electronic Code Book)

   - This variant of Triple DES works exactly the same way as the ECB mode of DES.
   - This is the most commonly used mode of operation.

   Triple CBC (Cipher Block Chaining)

   - This method is very similar to the standard DES CBC mode.
   - As with Triple ECB, the effective key length is 168 bits and keys are used in the same manner, as described above, but the chaining features of CBC mode are also employed.
   - The first 64-bit key acts as the Initialization Vector to DES.
   - Triple ECB is then executed for a single 64-bit block of plaintext.
   - The resulting ciphertext is then XORed with the next plaintext block to be encrypted, and the procedure is repeated.
   - This method adds an extra layer of security to Triple DES and is therefore more secure than Triple ECB, although it is not used as widely as Triple ECB.

3. **RC4** - RC4 was designed in 1987 by Ron Rivest and is one of the most widely software stream cipher and used in popular protocols, such as SSL (protect Internet traffic), WEP (secure wireless networks) and PDF. It’s considered to be fast and simple in terms of software.
   RC4 generates a pseudo-random stream of bits (a key-stream). As with any stream cipher, these can be used for encryption by combining it with the plaintext using bit-wise exclusive-or. Decryption is performed the same way (since exclusive-or is a symmetric operation).
   To generate the key stream, the cipher makes use of a secret internal state which consists of two parts:
   1. A permutation of all 256 possible bytes (denoted "S" below).
   2. Two 8-bit index-pointers (denoted "i" and "j").
   The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA). Then the stream of bits is generated by a pseudo-random generation algorithm.



   The output byte is selected by looking up the values of S(i) and S(j), adding them together modulo 256, and then looking up the sum in S; S(S(i) + S(j)) is used as a byte of the key-stream, K.

4. **MD5** - Developed in 1991. It is basically MD4 with "safety-belts" and while it is slightly slower than MD4, it is more secure. The algorithm consists of four distinct rounds, which has a slightly different design from that of MD4. Message-digest size, as well as padding requirements, remains the same.

   MD5 algorithm description

   MD5 algorithm takes input message of arbitrary length and generates 128-bit long output hash. MD5 hash algorithm consists of 5

- **Step 1. Append Padding Bits**
- **Step 2. Append Length**
- **Step 3. Initialize MD Buffer**
- **Step 4. Process Message in 16-Word Blocks**
- **Step 5. Output**

MD5 algorithm uses

MD5 is commonly used hash algorithm. It can be found in many implementations (available on some unix-based system as utility *md5*; class *MD5CryptoServiceProvider* in Microsoft's .NET Framework (namespace *System.Security.Cryptography*); example implemetation in Visual C++ or JavaScript , etc). It is used sometimes as file CRC function (Napster...) or one-way cipher in authentication operations (for storing user password hash).

MD5 is also used in conjunction with other cryptographic methods in digital signature applications or in protocols like SSL and others.

CHAPTER 3

PROJECT OBJECTIVE AND SCOPE

The main objective of the project JAVA ENCRYPTOR is to easily use the desktop application for encrypting and decrypting the entered string using the specified key and algorithm.

Certain messages and text require proper encryption so that the text can be sent privately and can be decrypted only using the same key.

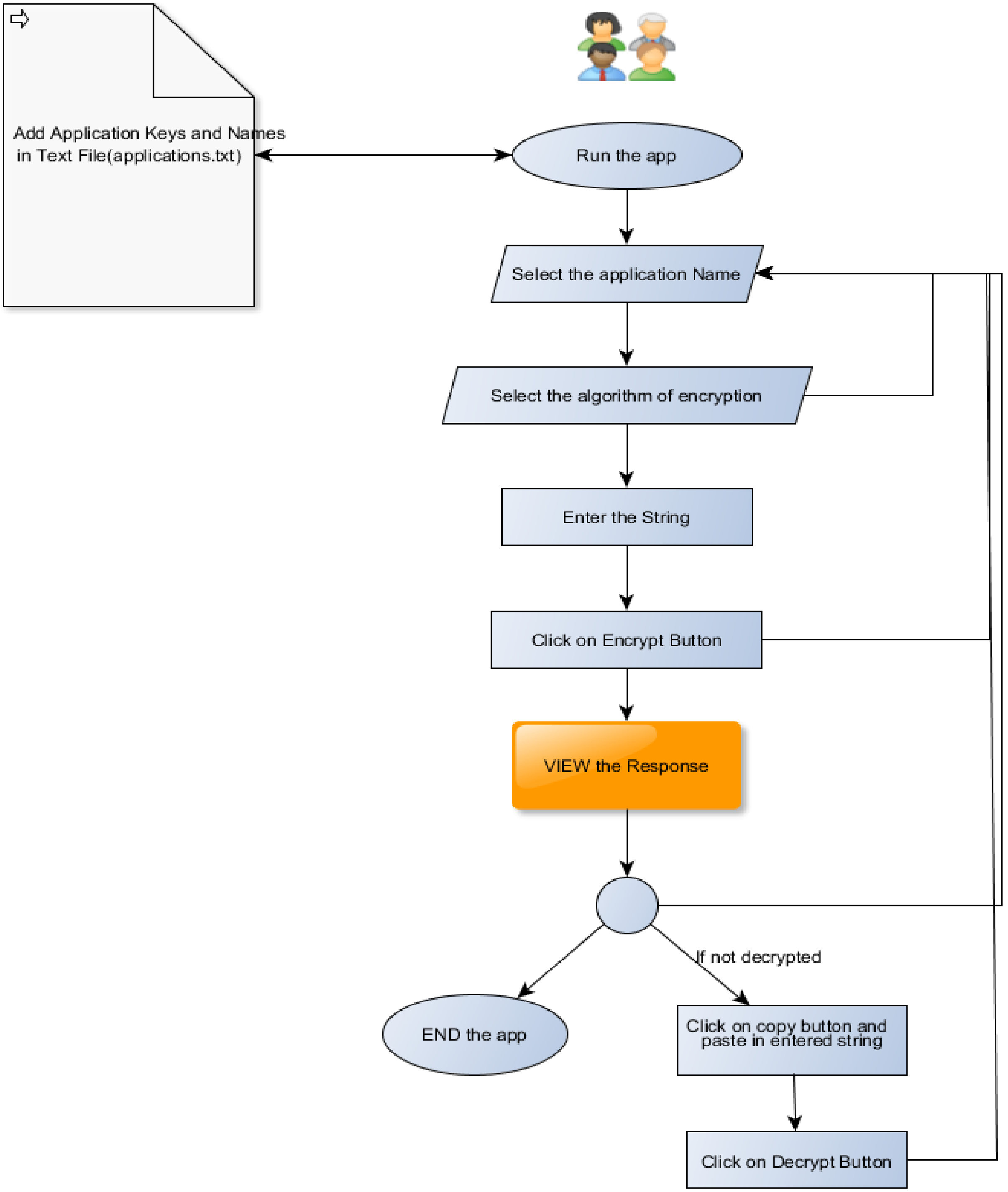
This desktop app made in JAVA using SWING lets you add Application name and its key in a text file separated by space. After that the app is started and lets you choose the application name for the key and the algorithm by which you want to encrypt your message. Then you are supposed to enter the string to be encrypted. After clicking the encrypt button, the encrypted value will be shown in Response Text Field. Similarly you can decrypt the encrypted string.

Features of JAVA ENCRYPTOR

- Application choice
- Algorithm choice
- Encryption of string
- Decryption of encrypted string

CHAPTER 4

JAVA ENCRYPTOR ARCHITECTURE MODEL


Add Application Keys and Names
in Text File(applications.txt)
Run the app
Select the application Name
Select the algorithm of encryption
Enter the String
Click on Encrypt Button
VIEW the Response
If not decrypted
END the app
Click on copy button and paste in entered string
Click on Decrypt Button

CHAPTER 5

DEPLOYMENT OF JAVA ENCRYPTOR

System Requirements

Windows

- Windows 10 (8u51 and above)
- Windows 8.x (Desktop)
- Windows 7 SP1
- Windows Vista SP2
- Windows Server 2008 R2 SP1 (64-bit)
- Windows Server 2012 and 2012 R2 (64-bit)
- RAM: 128 MB
- Disk space: 124 MB for JRE; 2 MB for Java Update
- Processor: Minimum Pentium 2 266 MHz processor
- Browsers: Internet Explorer 9 and above, Firefox

Mac OS X

- Intel-based Mac running Mac OS X 10.8.3+, 10.9+
- Administrator privileges for installation
- 64-bit browser

A 64-bit browser (Safari, Firefox for example) is required to run Oracle Java on Mac OS X.

Linux

- Oracle Linux 5.5+[1]
- Oracle Linux 6.x (32-bit), 6.x (64-bit)[2]
- Oracle Linux 7.x (64-bit)[2] (8u20 and above)
- Red Hat Enterprise Linux 5.5+[1], 6.x (32-bit), 6.x (64-bit)[2]
- Red Hat Enterprise Linux 7.x (64-bit)[2] (8u20 and above)
- Suse Linux Enterprise Server 10 SP2+, 11.x
- Suse Linux Enterprise Server 12.x (64-bit)[2] (8u31 and above)
- Ubuntu Linux 12.04 LTS, 13.x
- Ubuntu Linux 14.x (8u25 and above)
- Ubuntu Linux 15.04 (8u45 and above)
- Ubuntu Linux 15.10 (8u65 and above)
- Browsers: Firefox

Other Pre-requisite for deploying the javaEncryptor application:

Add the application names and their keys in applications.txt file in the project folder to encrypt according to your specified key.

To deploy the javaEncryptor application, you would take the following steps:

1. Open a terminal window. On Microsoft Windows systems, you do this by choosing **Start > Run**, typing **cmd** in the Open field, and clicking OK.
2. Change directories to the *PROJECT_HOME*/dist folder (using the cd command).
3. Type the following line to run the application's main class:

   "java -jar javaEncryptor.jar"
4. Select the application name which will automatically select the key with that name specified in "**applications.txt**".
5. Select the algorithm for encryption among **AES, Triple-DES, RC4, MD5**.
6. Enter the String.
7. Click on Encrypt Button.
8. View the response in Response text field.
9. Copy the Response and paste in Input String text field.
10. Decrypt the String when required.

NOTE: JAVA 7 or above should be installed on the PC where the JAR file of JAVA ENCRYPTOR has to be executed.

CHAPTER 6

SAMPLE CODE

mainFile.java

```
package javaencryptor;
import java.awt.*;
import java.awt.event.*;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.io.*;
import java.util.Scanner;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.BadPaddingException;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.swing.*;
import java.awt.datatransfer.*;
import java.awt.Toolkit;
import java.util.ArrayList;
/**
 * @author umang18oct
 */
public class mainFile {
   private Scanner fileScan;
   private JFrame mainFrame;
   private JLabel headerLabel;
   private JLabel statusLabel;
   private JButton encryptButton;
   private JButton decryptButton;
   private JPanel appPanel;
   private JPanel algoPanel;
   private JPanel textPanel;
   private JPanel responsePanel;
   private JPanel buttonPanel;
   private int[] counter= new int[4];
   private static final int PADDING = 50;
   ArrayList values;
   ArrayList names;
   JComboBox appsName=new JComboBox();

   public mainFile(String fName){
      prepareGUI();
      ArrayList tmp;
      tmp = new ArrayList();
      try{
          InputStream ips=new FileInputStream(fName);
          InputStreamReader ipsr=new InputStreamReader(ips);
          try (BufferedReader br = new BufferedReader(ipsr)) {
              String line;
              while ((line=br.readLine())!=null) {
                  String[] s = line.split(" ");
                  tmp.add(s[0]);
                  tmp.add(s[1]);
                  //System.out.println(tmp);
              }
          }
        }
        catch (Exception e){
        }
        int j=0; int k=0;
        values=new ArrayList();
        names=new ArrayList();
        for(int i=0;i<tmp.toArray().length;i++){
        if(i%2!=0)
            values.add(tmp.get(i));
        else
            names.add(tmp.get(i));
        }
       //System.out.println(names);
       //System.out.println(values);
        appsName.setModel(new DefaultComboBoxModel(names.toArray()));
        counter[0]=0;
        counter[1]=0;
        counter[2]=0;
        counter[3]=0;
    }

   public static void main(String[] args){
      mainFile mainObject = new
mainFile("C://Users/umang18oct/Documents/NetBeansProjects/JavaEncryptor/src/javaencryptor/applicati
ons.txt");
      mainObject.mainApp();
```

```
}
private void prepareGUI(){
  mainFrame = new JFrame("Java Encryptor");
  mainFrame.setSize(1050,500);
  mainFrame.setLayout(new GridLayout(7, 1));
  mainFrame.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent windowEvent){
      System.exit(0);
    }
  });

  headerLabel = new JLabel("", JLabel.CENTER);
  statusLabel = new JLabel("",JLabel.CENTER);

  appPanel = new JPanel();
  appPanel.setLayout(new FlowLayout(FlowLayout.LEADING,PADDING,0));

  algoPanel = new JPanel();
  algoPanel.setLayout(new FlowLayout(FlowLayout.LEADING, PADDING, 0));

  textPanel = new JPanel();
  textPanel.setLayout(new FlowLayout(FlowLayout.LEADING, PADDING, 0));

  responsePanel = new JPanel();
  responsePanel.setLayout(new FlowLayout(FlowLayout.LEADING, PADDING, 0));

  buttonPanel = new JPanel();
  buttonPanel.setLayout(new FlowLayout());
  mainFrame.add(headerLabel);
  mainFrame.add(appPanel);
  mainFrame.add(algoPanel);
  mainFrame.add(textPanel);
  mainFrame.add(responsePanel);
  mainFrame.add(buttonPanel);
  mainFrame.add(statusLabel);
  mainFrame.setVisible(true);
}

private void mainApp(){
  headerLabel.setText("Welcome!");
  JLabel appLabel = new JLabel("",JLabel.CENTER);
  appLabel.setText("Choose Application : ");
  //final JComboBox appCombo;
```

```
//appCombo = new JComboBox((ComboBoxModel) appsName);

appsName.setSelectedIndex(0);
JScrollPane appListScrollPane = new JScrollPane(appsName);

JLabel algoLabel = new JLabel("",JLabel.LEFT);
algoLabel.setText("Choose Algorithm :    ");
final DefaultComboBoxModel algoName = new DefaultComboBoxModel();
algoName.addElement("AES");
algoName.addElement("Triple DES");
algoName.addElement("RC4");
algoName.addElement("MD5");
final JComboBox algoCombo = new JComboBox(algoName);
algoCombo.setSelectedIndex(0);
JScrollPane algoListScrollPane = new JScrollPane(algoCombo);

JLabel textLabel = new JLabel("",JLabel.CENTER);
textLabel.setText("Enter the String :        ");
JTextField textField = new JTextField(16);

JLabel responseLabel = new JLabel("",JLabel.CENTER);
responseLabel.setText("Response :                  ");
JTextField responseField = new JTextField(16);
responseField.setEditable(false);

JButton copyButton = new JButton("COPY");
copyButton.addActionListener(new ActionListener (){
    public void actionPerformed(ActionEvent e){
        String responseData = responseField.getText();
        StringSelection responseString = new StringSelection(responseData);
        Clipboard copy = Toolkit.getDefaultToolkit().getSystemClipboard();
        copy.setContents(responseString, null);
    }
});

encryptButton = new JButton("ENCRYPT");
encryptButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String textData = textField.getText();
        String keyData="";
        if (appsName.getSelectedIndex() != -1) {
            keyData +=values.get(appsName.getSelectedIndex());
        }
        String responseData="";
```

```
            String algoData="";
            if (algoCombo.getSelectedIndex() != -1) {
               algoData +=algoCombo.getItemAt(algoCombo.getSelectedIndex());
            }
            switch (algoData) {
               case "RC4":
                        {
                           try {
                           responseData+=rc4.encrypt(textData,keyData);
                           } catch (Exception ex) {
                             Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                           }
                           counter[0]=1;
                           decryptButton.setEnabled(true);
                           break;
                        }
               case "MD5":
                        {
                           try {
                           responseData+=md5.MD5(textData);
                           } catch (NoSuchAlgorithmException | UnsupportedEncodingException ex) {
                           Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                           }
                           counter[1]=1;
                           decryptButton.setEnabled(false);
                           break;
                        }
               case "AES":
                        {
                           try {
                              responseData+=aes.encrypt(textData,keyData);
                           } catch (NoSuchAlgorithmException | NoSuchPaddingException |
InvalidKeyException | IllegalBlockSizeException | BadPaddingException | IOException ex) {
                              Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                           }
                           counter[2]=1;
                           decryptButton.setEnabled(true);
                           break;
                        }
               case "Triple DES":
                              {
                                 tripleDES obj1 = null;
                                 try {
                                   obj1 = new tripleDES();
```

```
                                  }
                                  catch (Exception ex) {
                                    Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                                  }
                                  responseData+=obj1.encrypt(textData,keyData);
                                  counter[3]=1;
                                  decryptButton.setEnabled(true);
                                  break;
                              }
        }
        //String responseData = ""+obj1.encrypt(textData,keyData);
        responseField.setText(responseData);
    }
});
decryptButton = new JButton("DECRYPT");
decryptButton.setEnabled(false);
decryptButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String textData = textField.getText();
        String keyData="";
        if (appsName.getSelectedIndex() != -1) {
            keyData +=values.get(appsName.getSelectedIndex());
        }
        String responseData="";
        String algoData="";
        if (algoCombo.getSelectedIndex() != -1) {
            algoData +=algoCombo.getItemAt(algoCombo.getSelectedIndex());
        }
        switch (algoData) {
            case "RC4":
                      { if(counter[0]==1){
                            try {
                              responseData+=rc4.decrypt(textData,keyData);
                            } catch (Exception ex) {
                                Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                            }
                        }
                        else{}
                        decryptButton.setEnabled(false);
                        break;
                      }
            case "MD5":
                      {
                          if(counter[1]==1){
                }
                else{}
                decryptButton.setEnabled(false);
                break;
              }
         case "AES":
              {
                if(counter[2]==1){
                  try {
                    responseData+=aes.decrypt(textData,keyData);
                  } catch (InvalidKeyException | IllegalBlockSizeException | BadPaddingException |
UnsupportedEncodingException | NoSuchAlgorithmException | NoSuchPaddingException |
InvalidAlgorithmParameterException ex) {
                    Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                  }
                }
                else{}
                decryptButton.setEnabled(false);
                break;
              }
         case "Triple DES":
                  {
                    if(counter[3]==1){
                      tripleDES obj1 = null;
                      try {
                       obj1 = new tripleDES();
                      }
                      catch (Exception ex) {
                       Logger.getLogger(mainFile.class.getName()).log(Level.SEVERE, null, ex);
                      }
                      responseData+=obj1.decrypt(textData,keyData);
                    }
                    else{}
                    decryptButton.setEnabled(false);
                    break;
                  }
       }
       //String responseData = ""+obj1.decrypt(textData,keyData);
       responseField.setText(responseData);
     }
   });
   statusLabel.setText("Hope you liked it!");
   appPanel.add(Box.createRigidArea(new Dimension(250,0)));
   appPanel.add(appLabel);
```

```
        appPanel.add(appListScrollPane);

        algoPanel.add(Box.createRigidArea(new Dimension(250,0)));
        algoPanel.add(algoLabel);
        algoPanel.add(algoListScrollPane);

        textPanel.add(Box.createRigidArea(new Dimension(250,0)));
        textPanel.add(textLabel);
        textPanel.add(textField);

        responsePanel.add(Box.createRigidArea(new Dimension(250,0)));
        responsePanel.add(responseLabel);
        responsePanel.add(responseField);
        responsePanel.add(copyButton);

        buttonPanel.add(encryptButton);
        buttonPanel.add(Box.createRigidArea(new Dimension(50,0)));
        buttonPanel.add(decryptButton);

        mainFrame.setVisible(true);
    }
}
```

aes.java

```
package javaencryptor;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

/**
 *
 * @author umang18oct
 */
```

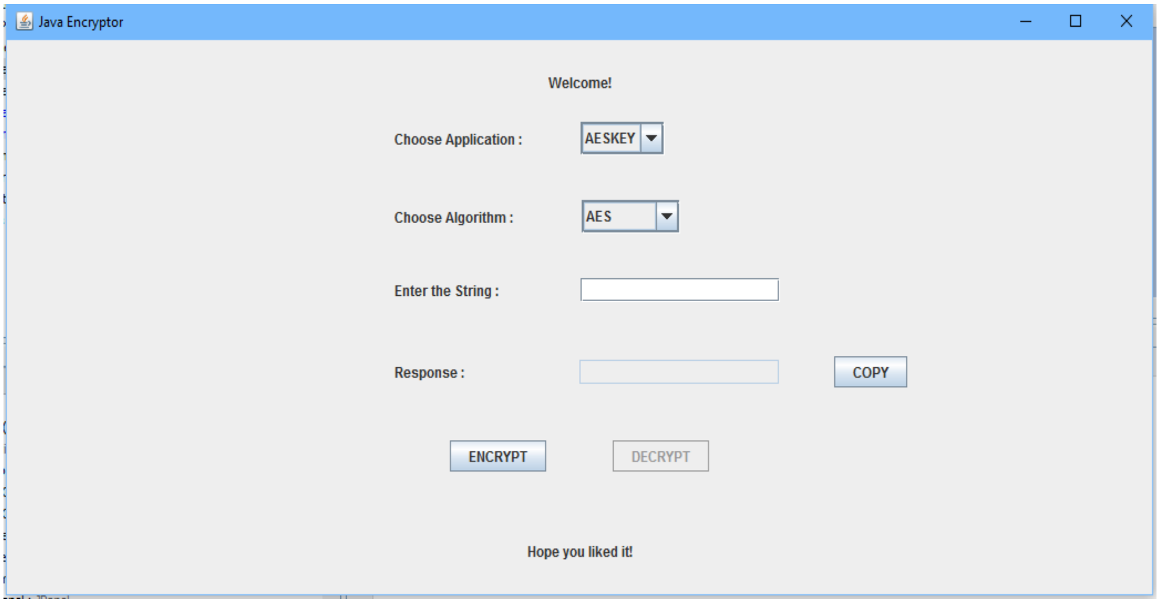
```
public class aes {

    private static final String ALGORITMO = "AES/CBC/PKCS5Padding";
    private static final String CODIFICACION = "UTF-8";

    public static String encrypt(String plaintext, String key)throws NoSuchAlgorithmException,
NoSuchPaddingException,InvalidKeyException, IllegalBlockSizeException,BadPaddingException,
IOException{
            byte[] raw = DatatypeConverter.parseHexBinary(key);
            SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
            Cipher cipher = Cipher.getInstance(ALGORITMO);
            cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
            byte[] cipherText = cipher.doFinal(plaintext.getBytes(CODIFICACION));
            byte[] iv = cipher.getIV();
            ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
            outputStream.write(iv);
            outputStream.write(cipherText);
            byte[] finalData = outputStream.toByteArray();
            String encodedFinalData = DatatypeConverter.printBase64Binary(finalData);
            return encodedFinalData;
    }

    public static String decrypt(String encodedInitialData, String key)throws InvalidKeyException,
IllegalBlockSizeException,BadPaddingException,
UnsupportedEncodingException,NoSuchAlgorithmException,
NoSuchPaddingException,InvalidAlgorithmParameterException{
            byte[] encryptedData = DatatypeConverter.parseBase64Binary(encodedInitialData);
            byte[] raw = DatatypeConverter.parseHexBinary(key);
            SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
            Cipher cipher = Cipher.getInstance(ALGORITMO);
            byte[] iv = Arrays.copyOfRange(encryptedData, 0, 16);
            byte[] cipherText = Arrays.copyOfRange(encryptedData, 16, encryptedData.length);
            IvParameterSpec iv_specs = new IvParameterSpec(iv);
            cipher.init(Cipher.DECRYPT_MODE, skeySpec, iv_specs);
            byte[] plainTextBytes = cipher.doFinal(cipherText);
            String plainText = new String(plainTextBytes);
            return plainText;
    }
}
```

applications.txt

```
/*Dummy Keys */
AESKEY 5468617473206D79204B756E67204675
XYZ 5327009233BCDE1234
```

CHAPTER 7

SAMPLE SCREENSHOTS



Welcome!

Choose Application : AESKEY

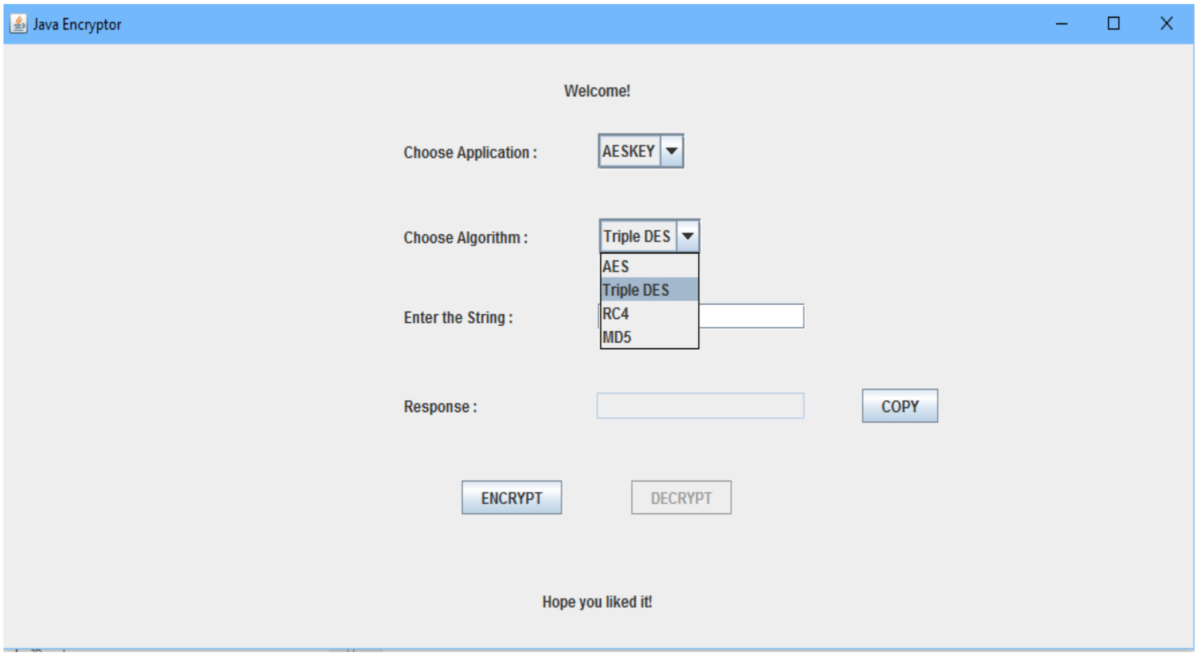
Choose Algorithm : AES

Enter the String :

Response :

Hope you liked it!

Figure 1 Choose Application name Eg: AESKEY



Welcome!

Choose Application : AESKEY

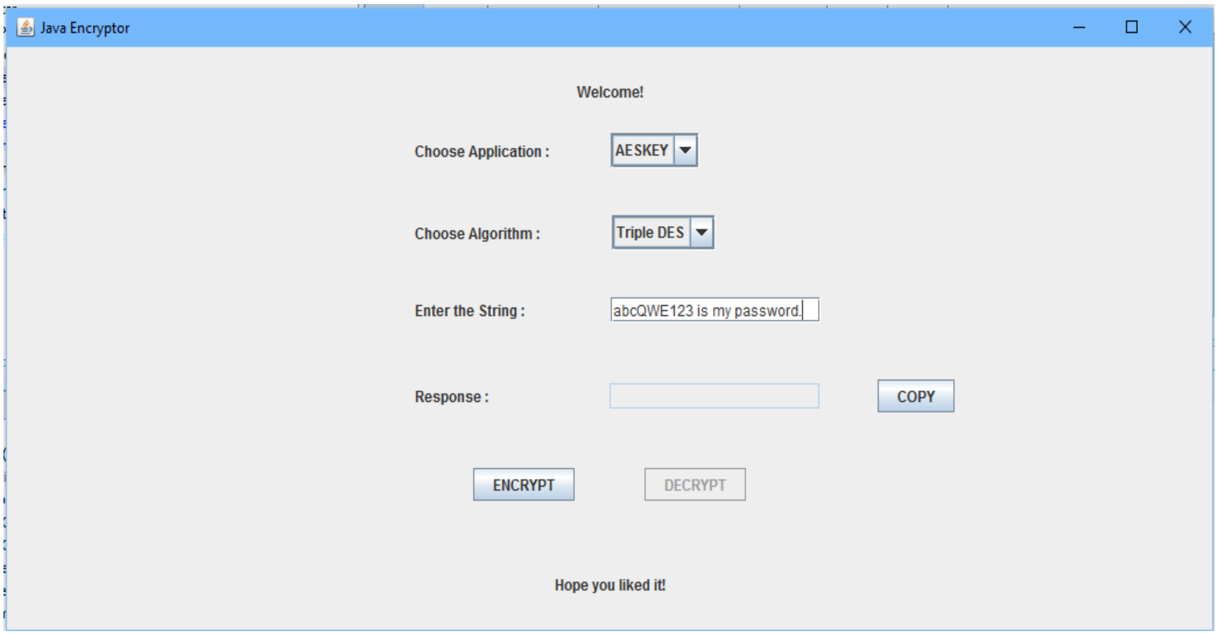
Choose Algorithm : Triple DES

Enter the String :

Response :

Hope you liked it!

Figure 2 Choose Algorithm Eg: Triple-DES



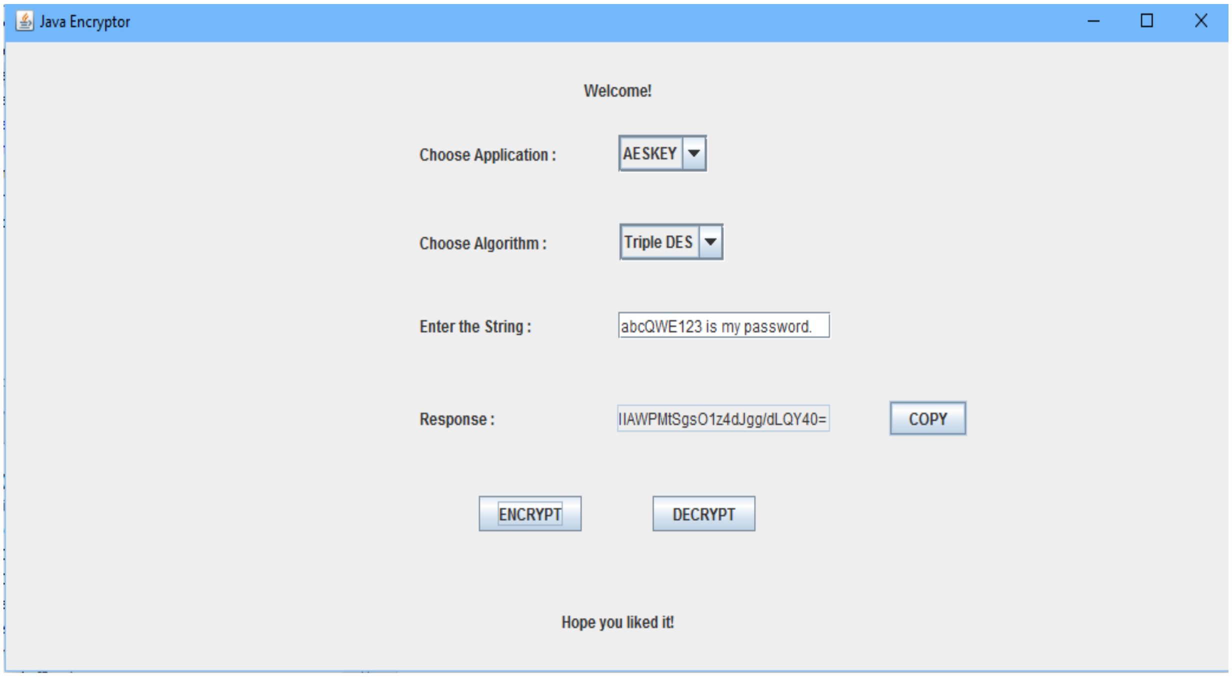
Choose Application : AESKEY

Choose Algorithm : Triple DES

Enter the String : abcQWE123 is my password.

Response :

Figure 3 Enter the String



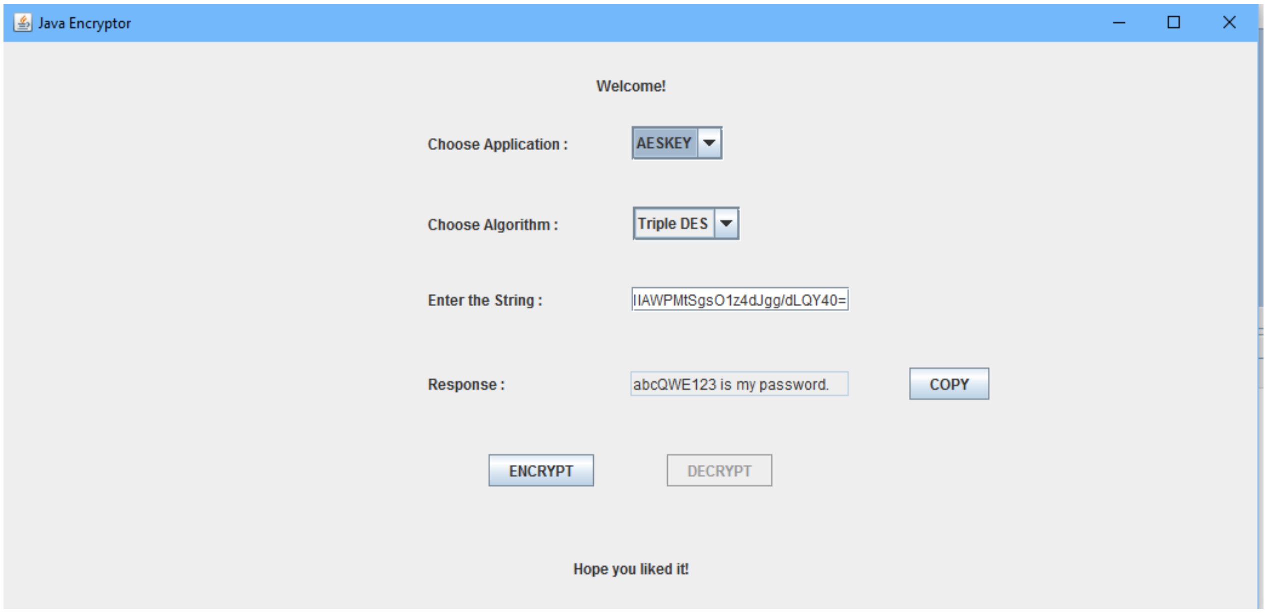
Choose Application : AESKEY

Choose Algorithm : Triple DES

Enter the String : abcQWE123 is my password.

Response : IIAWPMtSgsO1z4dJgg/dLQY40=

Figure 4 Click Encrypt Button and view the Response



Welcome!

Choose Application : AESKEY

Choose Algorithm : Triple DES

Enter the String : IIAWPMtSgsO1z4dJgg/dLQY40=

Response : abcQWE123 is my password.

Hope you liked it!

Figure 5 COPY the encrypted string and paste in Input String text field and click on Decrypt Button and view the decrypted Response

CHAPTER 8

BIBLIOGRAPHY

References:

- **http://www.stackoverflow.com/**
- **http://www.tutorialspoint.com/swing**
- **https://en.wikipedia.org**
- **https://www.nareshpatel.in**
- **http://aesencryption.net/**

GitHub Link for the Project:
https://github.com/umang18oct/javaEncryptor

LinkedIn Link for my Profile:
https://www.linkedin.com/in/umang18oct

Contact Details:

Umang Chaudhary
+91-9718248382
umang18oct@gmail.com

THANK YOU