PROJECT REPORT ON ENCRYPTION AND DECRYPTION

TABLE OF CONTENTS

Sr. No.	Title	Page No.
1.	**Abstract**	3
2.	**Objective and Scope**	3
	2.1 Objective	3
	2.2 Scope	4
3.	**Theoretical Background**	5
	3.1 Existing System	5
	3.2 Proposed System	5
	3.3 Features	5
	3.4 RSA Algorithm	6
4.	**Feasibility Study**	7
	4.1 Economic Feasibility	7
	4.2 Technical Feasibility	8
	4.3 Operational Feasibility	9
5.	**Problem Definition**	9
	5.1 Project Mission	9
	5.2 Target	9
	5.3 Target Users	9
	5.4 Scope and Key Elements	10
6.	**System Design**	11
	6.1 System Design	11
	6.2 Data Flow Diagram	11
	6.3 Use Case Diagram	11
7.	**Source Code**	12
8.	**Screens**	20
9.	**Conclusion**	21
10.	**References**	22

1.ABSTRACT

The title of our project is “ENCRYPTION AND DECRYPTION”. This project encrypts and decrypts the textual files by using RSA algorithm. Our aim is to develop the software named ENCRYPTION AND DECRYPTION that encrypts and decrypts the textual files by using RSA algorithm. Encryption and Decryption is a strong text and file encryption software for personal and professional security. It protects privacy of our email messages, documents and sensitive files by encrypting them using RSA algorithm to provide high protection against unauthorized data access.

Every day hundreds and thousands of people interact electronically, whether it is through emails, e-commerce, etc. through internet. Sending sensitive messages over the Internet is very dangerous. If you need to send sensitive messages over the Internet, you should send it in the encrypted form. Encryption and Decryption allows you easily encrypt and decrypt your messages. If you need to send sensitive information via email, simply paste the encrypted text into your email and all the recipients has to do is to decrypt the text.

2.OBJECTIVE & SCOPE

2.1 Objective

The main objective of our project is to encrypt/decrypt the textual files for personal and professional security. Encryption and Decryption protects privacy of our email messages, documents and sensitive files by encrypting them using RSA algorithm to provide high protection against unauthorized data access.

Every day hundreds and thousands of people interact electronically, whether it is through emails, e-commerce, etc. through internet. The Internet is comprised of millions of

interconnected communication and transfer of information around the world. People use emails to correspond with one another. The www is used for online business, data distribution, marketing, research, learning and a myriad of other activities.

Sending sensitive messages over the Internet is very dangerous as all emails are transmitted in an unsecured form and anybody - ISP, your boss, etc. can read your emails.

If you want to send sensitive information via email, simply paste the encrypted text into your email or attach the file, all the recipient has to do is to decrypt your text or file. Encryption and Decryption works with text information and files. Just select what you want to encrypt, and Encryption and Decryption software helps you keep documents, private information and files in a confidential way.

2.2 Scope

The scope of our project is presently specific. Both the sender and the receiver must have this software installed on their systems to encrypt/decrypt and compress/decompress the files transmitted between them. This includes all the users who want to interact electronically, whether it is through emails, e-commerce, etc. Through internet in order to keep their private information confidential.

- Each step is clearly stated and user will not face any ambiguity in using the software.
- The software provides clarity in its functionality even to naïve users.
- No complexity is involved.

3.THEORETICAL BACKGROUND

3.1 The Existing System

- As observed the current encryption/decryption software's doing the encryption and decryption task are all very complicated in their functionality.
- The method of encryption/decryption and key generation of current system for a new user to understand is complex in nature.

3.2 The Proposed System

The proposed system is quite simple to use. It is not complex in its functionalities. It is easy for a naïve user to use it.

If you want to send sensitive information via email, simply paste the encrypted text into your email or attach the encrypted file, all the recipient has to do is to decrypt your text or file. Encryption and Decryption works with text information and files. Just select what you want to encrypt, and Encryption and Decryption software helps you keep documents, private information and files in a confidential way.

3.3 Important features of Encryption and Decryption

- The system is highly user friendly.
- It uses two different keys (a key pair) for encryption and decryption. These algorithms are called "public-key" because the encryption key can be made public. Anyone can use the public key to encrypt a message, but only the owner of the corresponding private key can decrypt it.

- A message can be encrypted with a private key and decrypted with the corresponding public key. If Alice (or anyone else) can decrypt a message with Bob's public key she knows that the message must have come from Bob because no one else has Bob's private key.
- The system provides security and convenience as private keys never need to be transmitted or revealed to anyone.
- The system provides the integrity of data or information.
- The software provides clarity in its functionality even to naïve users.

3.4THE ALGORITHM

The RSA algorithm involves three steps, key generation, encryption and decryption.

Key Generation

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated in the following way:

1. Choose two distinct large random prime numbers p and q
2. Compute n=pq

   →n is used as the modulus for both the public and private keys
3. Compute the totient: $\varphi(n)=(p-1)(q-1)$.
4. Choose an integer e such that $1<e<\varphi(n)$ and e and $\varphi(n)$ share no factors other than 1 (i.e., e and $\varphi(n)$ are co-prime)

→e is released as the public key exponent

5. Compute d to satisfy the congruence relation $de \equiv 1 (\text{mod } \varphi(n)$; i.e., $de = 1 + k\,\varphi(n)$ for some integer k.

   →d is kept as the private key exponent

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d which must be kept secret.

Encryption

Sender A does the following: -

1. Obtains the recipient B's public key (n, e).
2. Represents the plaintext message as a positive integer M
3. Computes the ciphertext $C = M^e \text{ mod } n$.
4. Sends the ciphertext C to B.

Decryption

Recipient B does the following: -

1. Uses his private key (n, d) to compute $M = C^d \text{ mod } n$.
2. Extracts the plaintext from the message representative M.

4.FEASIBILITY STUDY

After investigation it is essential to determine whether the project is feasible or not. In feasibility study is tested whether the system to be developed would be able to accomplish its task on the working grounds. Its impact was also found to be not adverse. It was found that

the user's requirements would be met and the resources would be used in an effective manner. In feasibility study the important aspects related to the project were considered like the problem definition and the process for solution. The cost and benefit analysis was also done.

4.1 Feasibility Considerations

To do a feasibility study, the economic, technical and behavioral factors in the system development were considered.

The three key considerations were as follows:

4.1.1 Economic Feasibility

The project developed, Encryption and Decryption was within budget and producing the desired results. The labor or the human ware consisted of the three group members of our project. The output consisted of getting the desired results. Thus with the consideration of the inputs, the outputs were achieved successfully. The project was within limit. The inputs didn't overdo the outputs.

4.1.2 Technical Feasibility

Technical feasibility revolves around the technical support of the project. The main infrastructure of the project included the project labs in the college campus. The systems there were easily able to absorb the new s/w being installed. The project thus was technically feasible. The equipment and the s/w produced no problem. The project's technical requirements were met. The project could be made to work correctly, fulfilling its task, with the existing s/w and personnel.

4.1.3 Operational Feasibility

Operational Feasibility aims to determine the impact of the system on the users. The system developing has an influence on its users. Our system "Encryption and Decryption" was new for them but it was simple enough for any naïve person to understand. The evolution of this new system required no special training for the users. Encryption and Decryption was found to be feasible in this regard. The system developed would be user friendly and no complexities would be involved in its functionalities.

5.PROBLEM DEFINITION

5.1 Project Mission

The aim of our project is to develop software named ENCRYPTION AND DECRYPTION. The project encrypts and decrypts the textual files using RSA algorithm to maintain the security and integrity of data and information and to provide high protection against unauthorized data access.

5.2 Target

Our target is the common man who wants to interact electronically, whether it is through emails, e-commerce, etc. Through internet. Sending sensitive messages over the Internet is very dangerous. So, our project helps him to interact in a safe and secure manner in order to keep their private information confidential.

5.3 Target Users

The main target users of our project are the people who transmit confidential information via emails or through internet.

5.4 Scope and Key Elements

The scope of our project is presently specific. Both the sender and the receiver must have this software installed on their systems to encrypt/decrypt and compress/decompress the files transmitted between them. This includes all the users who want to interact electronically, whether it is through emails, e-commerce, etc. Through internet in order to keep their private information confidential. The key elements of our website include the objectives, plus the following:

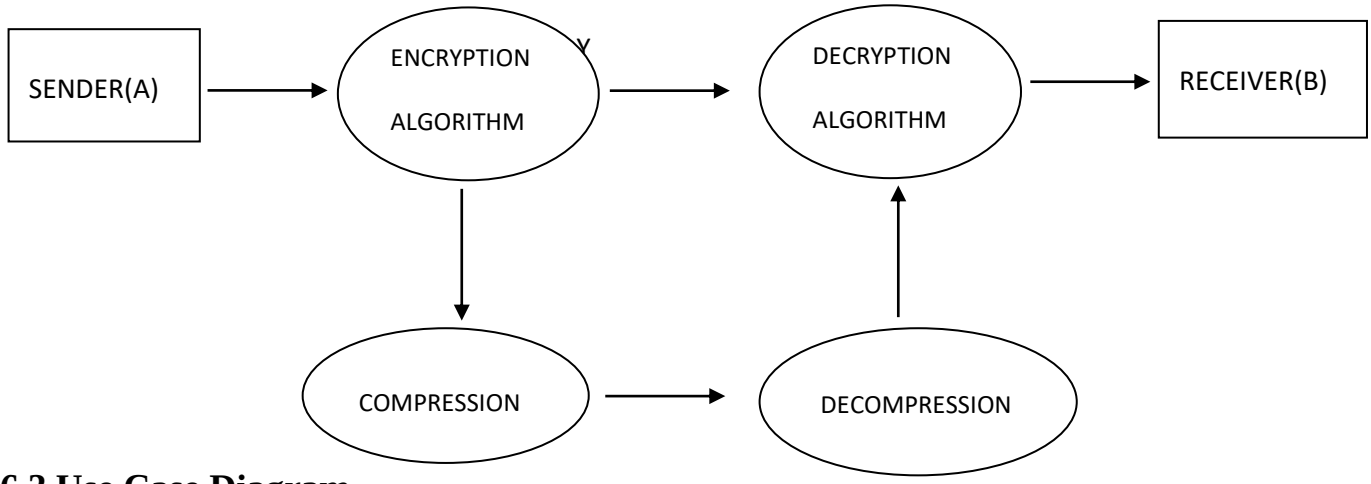
- The system provides the security and integrity of data or information.
- It will provide a more clear and non-ambiguous description of the functions.
- The system is highly user friendly.
- It uses two different keys (a key pair) for encryption and decryption. These algorithms are called "public-key" because the encryption key can be made public. Anyone can use the public key to encrypt a message, but only the owner of the corresponding private key can decrypt it.
- The software provides clarity in its functionality even to naïve users.

6. SYSTEM DESIGN

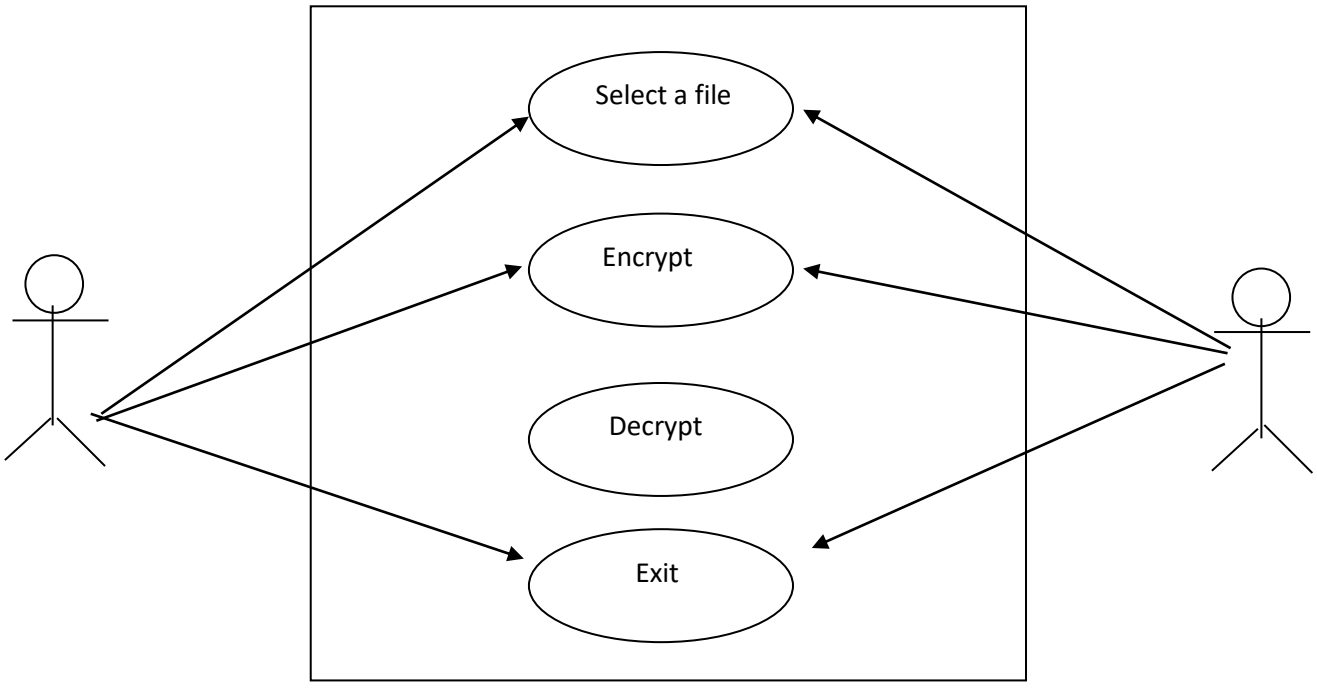
6.1 System Design

Software design sits at the technical kernel of software engineering process and is applied regardless of the development paradigm and the area of application. Once the system requirements have been analyzed and specified, system design is the first of the three technical activities- design, code and test that are required to build and verify s/w. emphasis is on translating the s/w requirements into design specification. It involves preparing I/P- O/P specifications, making security and control specification, and preparing a logical and physical design work through.

6.2 Data Flow Diagram



6.3 Use Case Diagram



7.SOURCE CODE

frmMain.java

```
package ecryptiondecryption;

import java.awt.Frame;
import java.awt.Toolkit;
import java.awt.datatransfer.StringSelection;
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

/**
 *
 * @author MH Habib
 */
public class EDcrypt extends javax.swing.JFrame {

    /**
     * Creates new form EDcrypt
     */
    private static SecretKeySpec secretKey;
    private static byte[] key;
    public static void setKey(String myKey)
    {
        MessageDigest sha = null;
        try {
            key = myKey.getBytes("UTF-8");
            sha = MessageDigest.getInstance("SHA-1");
            key = sha.digest(key);
            key = Arrays.copyOf(key, 16);
            secretKey = new SecretKeySpec(key, "AES");
        }
        catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
    public EDcrypt() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
```

```
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    text1 = new javax.swing.JTextArea();
    jScrollPane2 = new javax.swing.JScrollPane();
    text2 = new javax.swing.JTextArea();
    jScrollPane3 = new javax.swing.JScrollPane();
    text3 = new javax.swing.JTextArea();
    jScrollPane4 = new javax.swing.JScrollPane();
    text4 = new javax.swing.JTextArea();
    msg1 = new javax.swing.JTextField();
    msg2 = new javax.swing.JTextField();
    encrypt = new javax.swing.JButton();
    decrypt = new javax.swing.JButton();
    copyencrypt = new javax.swing.JButton();
    copydecrypt = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    message1 = new javax.swing.JLabel();
    message2 = new javax.swing.JLabel();
    mainsection = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Encryption and Decryption ");
    setAlwaysOnTop(true);
    setUndecorated(true);
    setResizable(false);
    getContentPane().setLayout(null);

    text1.setBackground(new java.awt.Color(204, 255, 255));
    text1.setColumns(20);
    text1.setFont(new java.awt.Font("Dialog", 0, 14)); // NOI18N
    text1.setRows(5);
    text1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(102, 102, 102), 2));
    jScrollPane1.setViewportView(text1);

    getContentPane().add(jScrollPane1);
    jScrollPane1.setBounds(80, 80, 300, 120);

    text2.setBackground(new java.awt.Color(204, 255, 255));
    text2.setColumns(20);
    text2.setFont(new java.awt.Font("Dialog", 0, 14)); // NOI18N
    text2.setRows(5);
    text2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(102, 102, 102), 2));
    jScrollPane2.setViewportView(text2);
```

```
        getContentPane().add(jScrollPane2);
        jScrollPane2.setBounds(80, 260, 300, 140);

        text3.setBackground(new java.awt.Color(204, 255, 255));
        text3.setColumns(20);
        text3.setFont(new java.awt.Font("Dialog", 0, 14)); // NOI18N
        text3.setRows(5);
        text3.setToolTipText("");
        text3.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(102, 102, 102), 2));
        jScrollPane3.setViewportView(text3);

        getContentPane().add(jScrollPane3);
        jScrollPane3.setBounds(470, 80, 320, 120);

        text4.setBackground(new java.awt.Color(204, 255, 255));
        text4.setColumns(20);
        text4.setFont(new java.awt.Font("Dialog", 0, 14)); // NOI18N
        text4.setRows(5);
        text4.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(102, 102, 102), 2));
        jScrollPane4.setViewportView(text4);

        getContentPane().add(jScrollPane4);
        jScrollPane4.setBounds(470, 260, 320, 140);

        msg1.setBackground(new java.awt.Color(204, 204, 204));
        msg1.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        msg1.setBorder(new javax.swing.border.MatteBorder(null));
        getContentPane().add(msg1);
        msg1.setBounds(200, 220, 180, 30);

        msg2.setBackground(new java.awt.Color(204, 204, 204));
        msg2.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        msg2.setBorder(new javax.swing.border.MatteBorder(null));
        getContentPane().add(msg2);
        msg2.setBounds(595, 220, 190, 30);

        encrypt.setBackground(new java.awt.Color(0, 0, 0));
        encrypt.setFont(new java.awt.Font("Ebrima", 1, 14)); // NOI18N
        encrypt.setForeground(new java.awt.Color(255, 255, 255));
        encrypt.setText("Encrypt");

encrypt.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED
));
        encrypt.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        encrypt.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                encryptActionPerformed(evt);
            }
        });
        getContentPane().add(encrypt);
        encrypt.setBounds(80, 420, 90, 30);

        decrypt.setBackground(new java.awt.Color(0, 0, 0));
```

```
        decrypt.setFont(new java.awt.Font("Ebrima", 1, 14)); // NOI18N
        decrypt.setForeground(new java.awt.Color(255, 255, 255));
        decrypt.setText("Decrypt");

decrypt.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED
));
        decrypt.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        decrypt.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                decryptActionPerformed(evt);
            }
        });
        getContentPane().add(decrypt);
        decrypt.setBounds(470, 420, 90, 30);

        copyencrypt.setBackground(new java.awt.Color(102, 0, 0));
        copyencrypt.setFont(new java.awt.Font("Ebrima", 1, 14)); // NOI18N
        copyencrypt.setForeground(new java.awt.Color(255, 255, 255));
        copyencrypt.setText("Copy Encryption");

copyencrypt.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAI
SED));
        copyencrypt.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        copyencrypt.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                copyencryptActionPerformed(evt);
            }
        });
        getContentPane().add(copyencrypt);
        copyencrypt.setBounds(240, 420, 140, 30);

        copydecrypt.setBackground(new java.awt.Color(102, 0, 0));
        copydecrypt.setFont(new java.awt.Font("Ebrima", 1, 14)); // NOI18N
        copydecrypt.setForeground(new java.awt.Color(255, 255, 255));
        copydecrypt.setText("Copy Decryption");

copydecrypt.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAI
SED));
        copydecrypt.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        copydecrypt.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                copydecryptActionPerformed(evt);
            }
        });
        getContentPane().add(copydecrypt);
        copydecrypt.setBounds(640, 420, 150, 30);

        jLabel2.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        jLabel2.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mousePressed(java.awt.event.MouseEvent evt) {
                jLabel2MousePressed(evt);
            }
        });
```

```
        getContentPane().add(jLabel2);
        jLabel2.setBounds(810, 5, 30, 20);

        jLabel3.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        jLabel3.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mousePressed(java.awt.event.MouseEvent evt) {
                jLabel3MousePressed(evt);
            }
        });
        getContentPane().add(jLabel3);
        jLabel3.setBounds(780, 5, 30, 20);

        jLabel1.setFont(new java.awt.Font("Ebrima", 1, 15)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(204, 51, 0));
        jLabel1.setText("Encryption Key :");
        getContentPane().add(jLabel1);
        jLabel1.setBounds(80, 220, 120, 30);

        jLabel4.setFont(new java.awt.Font("Ebrima", 1, 15)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(204, 51, 0));
        jLabel4.setText("Decryption Key :");
        getContentPane().add(jLabel4);
        jLabel4.setBounds(470, 220, 120, 30);

        jLabel5.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(0, 0, 51));
        jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel5.setText("Decryption");
        getContentPane().add(jLabel5);
        jLabel5.setBounds(470, 50, 300, 30);

        jLabel6.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
        jLabel6.setForeground(new java.awt.Color(0, 0, 51));
        jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel6.setText("Encryption");
        getContentPane().add(jLabel6);
        jLabel6.setBounds(80, 50, 300, 30);

        message1.setForeground(new java.awt.Color(204, 0, 0));
        message1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        getContentPane().add(message1);
        message1.setBounds(80, 450, 300, 20);

        message2.setForeground(new java.awt.Color(204, 0, 0));
        message2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        getContentPane().add(message2);
        message2.setBounds(470, 450, 320, 20);

        mainsection.setBackground(new java.awt.Color(204, 204, 204));
        mainsection.setForeground(new java.awt.Color(204, 0, 0));
        mainsection.setIcon(new javax.swing.ImageIcon(getClass().getResource("/image/edcrypt.png"))); //
NOI18N
        mainsection.setAutoscrolls(true);
```

```
        mainsection.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        getContentPane().add(mainsection);
        mainsection.setBounds(0, 0, 850, 500);

        setSize(new java.awt.Dimension(850, 499));
        setLocationRelativeTo(null);
    }// </editor-fold>

    private void jLabel2MousePressed(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        System.exit(0);

    }

    private void jLabel3MousePressed(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        this.setState(Frame.ICONIFIED);
    }

    private void encryptActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String strToEncrypt;
        String secret;
        try
        {
            strToEncrypt=text1.getText();
            secret=msg1.getText();
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            text2.setText(Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-
8"))));

        }
        catch (Exception e)
        {
            text2.setText("Please fill up the right secret key");

        }
    }

    private void copyencryptActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Toolkit.getDefaultToolkit().getSystemClipboard().setContents(new
StringSelection(text2.getText()),null);
        message1.setText("Your encryption result is copied!");
        message2.setText("");
    }

    private void decryptActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String secret;
        String strToDecrypt;
```

```
        try
        {
            secret=msg2.getText();
            strToDecrypt=text3.getText();
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            text4.setText(new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt))));
            //return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
        }
        catch (Exception e)
        {
            text4.setText("Please fill up the right secret key");
        }
    }

    private void copydecryptActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Toolkit.getDefaultToolkit().getSystemClipboard().setContents(new
StringSelection(text4.getText()),null);
        message2.setText("Your decryption result is copied!");
        message1.setText("");
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(EDcrypt.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(EDcrypt.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(EDcrypt.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(EDcrypt.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
```

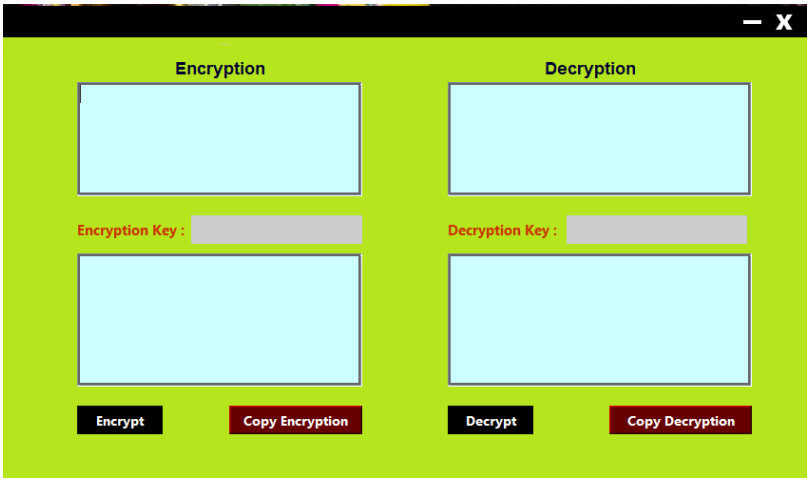
```
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new EDcrypt().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton copydecrypt;
    private javax.swing.JButton copyencrypt;
    private javax.swing.JButton decrypt;
    private javax.swing.JButton encrypt;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JLabel mainsection;
    private javax.swing.JLabel message1;
    private javax.swing.JLabel message2;
    private javax.swing.JTextField msg1;
    private javax.swing.JTextField msg2;
    private javax.swing.JTextArea text1;
    private javax.swing.JTextArea text2;
    private javax.swing.JTextArea text3;
    private javax.swing.JTextArea text4;
    // End of variables declaration
}
```

8. SCREENS

HOME PAGE



ENCRYPTION

DECRYPTION

9.CONCLUSION

After implementing the system in the market its advantages were incomparable to the present contemporary systems available in the market. The most admirable feature founded was its simplicity in terms of application to the user but its highly beneficial outputs can't be ignored. The users will be highly benefited after using the system.

There is always a room for improvement in any software, however efficient the system may be. The system is flexible enough for future modifications. The system has been factored into different modules to make the system adapt to further changes. Every effort has been made to cover all the user requirements and make it user friendly. Appropriate messages and tool tips have been provided wherever necessary.

Goals achieved: Following are the goals achieved in designing and implementing the system.

User Friendliness: The system is user friendly as any naïve user can use it and easily understand its functionality.

Security: Security is enabled as certain features and functionalities are restricted to the registered users only and also at other levels security is implemented.

Fast: The system aims to respond to the user as fast as possible.

Future Plans: The system is adaptable to the future changes if any are deployed.

10. REFERENCES

- www.google.com
- http://en.wikipedia.org/wiki/RSA_algorithm
- "Data Communications and Networking", 4th Edition, by Behrouz A. Forouzan, Tata McGraw Hill
- Roger S.Pressman," Software Engineering A Practitioners Approach", McGraw Hill, 1992, p.p.207-237.
- E Balagurusamy," Programming with JAVA", McGraw Hill,2003.
- William Stalings," Cryptography and Network Security".
- RSA Key Generator for default keys used: http://crypto.cs.mcgill.ca/~crepeau/RSA/generator_frame.html
- RSA Patent info: www.rsasecurity.com/rsalabs/node