INFORMATION SECURITY ANALYSIS AND AUDIT PROJECT

COURSE CODE:CSE3501
SLOT:F2,L49+L50
SEMESTER:FALL SEMESTER 2022-2023
GROUP MEMBERS:

SWATHI BAIJU 20BCE0749
H.THIRSHA SREE 20BCE2518
AISHWARYA GANESH SHANKAR 20BCE2629

SUBMITTED TO:Prof. RAJARAJAN G

FINAL PROJECT REVIEW

IMAGE ENCRYPTION AND DECRYPTION USING AES,DES AND 3DES

INDEX:

1. **Introduction**
2. **Objectives**
3. **Literature studies**
4. **Existing system/model**
5. **Proposed Model**
6. **Outcomes of the proposed model**
7. **Merits and demerits of the model**
8. **Code**
9. **Screenshot of the output**
10. **Future work**
11. **Conclusion**
12. **References**

INTRODUCTION:

Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication. There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public key (or asymmetric) cryptography, and hash functions. The rapid continuous increase in exchange of multimedia data over protected and unprotected
networks such as the worldwide available internet and local networks such as shared networks and local area networks etc has encouraged activities such as unauthorised access, illegal usage, disruption, alteration of transmitted and stored data. This widely spread use of digital media over the internet such as on social media, won cloud storage systems etc and over other communication medium such as satellite communication systems have increased as applications and need for systems to meet current and future demands evolved over the years. Security concerns with regards to such data transmission and storage has been a major concern of both the transmitters and receivers and hence the security of critical cyber and physical infrastructures as well as their underlying computing and communication architectures and systems becomes a very crucial priority of every institution.
Cryptography is the fundamental platform in which modern information security, which involves the use of advanced mathematical approaches in solving hard cryptographic issues, has gained its grounds in the digital world. This has evolved from classical symmetric, in which shifting keys are normally used as well as substitution methods, ciphers to modern public key exchange cryptosystems, which aims to make cryptanalysis a difficult approach to deciphering ciphers, eg. RSA, ElGamal, elliptic curve, Diffie-Hellman key exchange, and they are used in digital signature algorithms and now cutting edge works such as the
quantum cryptography.

OBJECTIVES:

In today's world almost all digital services like internet communication, medical and military imaging systems, multimedia system needs a high level security. There is a need for security level in order to safely store and transmit digital images containing critical information. This is because of the faster growth in multimedia technology, internet and cell phones. Therefore there is a need for image encryption and decryption techniques in order to hide images from such attacks. Such technique helps to avoid intrusion attacks.

LITERATURE STUDIES:

1. Image Encryption Based on the Modified Triple- DES Cryptosystem

ABSTRACT :
When encryption grey-scale images are stored using a lossless format (e.g. not JPEG), and a fixed key for a symmetric cryptosystem is used, edges may reveal due areas of the same colour in the image, thus giving some information about the original image. This is

particularly problematic when the image contains only two colours, black and white being the worst case scenario. The former problem may be solved using an asymmetric system, but the time employed in its operation is a relevant detrimental factor. However, using the JV Theorem along with Triple- DES | with the modification that a variable permutation is used on the input string for the third round of the encryption process first cycle— the output is a grey- scale image whose tones present a uniform distribution, according to a chi-square goodness-of-fit test. Also, it is shown that the use of variable permutation strengthens the Triple-DES cryptosystem. On the other hand, since the images are being encrypted using a symmetric cypher algorithm, the processing time is less than that of any known asymmetric algorithm. Furthermore, a criterion for computing the amount of permutations is provided.

2. IMAGE ENCRYPTION AND DECRYPTION USING AES ALGORITHM

ABSTRACT : Cryptography is art of converting one form of data into another form and in turn make the data unreadable for the end users. This cryptography has two methods like encryption and decryption. The process of converting plain text into cipher text is known as encryption and the process of converting cipher text into plain text is known as decryption. In this application we try to use AES algorithm for encryption or decryption of either text or image data

3.Performance evaluation of various symmetric encryption algorithms

ABSTRACT-With rise in the use of internet in various fields like banking, military,government and various other sectors,the security and privacy of the data has been the main concern. Today, most of the data handling and electronic transactions are done on the internet. When the information is transferred from the sender to the receiver over the internet,it may be eavesdropped by an intruder and thus is a continuous threat to the secrecy or confidentiality of the data. The popular technique that protects the confidentiality of the data is cryptography which converts the plain text or images into unreadable form(cipher form) and then receiver applies reverse mechanism to decrypt the unreadable form of data to readable form. This mechanism is called as encryption-decryption process. Thus to secure the data over the internet, it is important to find out which algorithm performs better than the other algorithms. In this paper, the different symmetric encryption algorithms like DES, 3DES, AES have been analysed with respect to different parameters and data types (like text files, image).

4. A Review on DES, AES and Blowfish for Image Encryption & Decryption

ABSTRACT-Cryptography & Network Security is a concept to protect network and data transmission over wireless network. Data Security is the main aspect of secure data transmission over unreliable network. Data Security is a challenging issue of data communications today that touches many areas including secure communication channel, strong data encryption technique and trusted third party to maintain the database. The conventional methods of encryption can only maintain the data security . Cryptography is a key technology in electronic key systems. It is used to keep data secret, digitally sign documents, access control and so forth.The terms used in cryptography are plain image,cipher (encrypted image), encryption, decryption. A process of converting Plain image into Cipher (encrypted image) is called as Encryption.Cryptography uses the encryption technique to send confidential messages through an insecure channel.Encryption takes place at the sender side. A reverse process of encryption is called as Decryption. It is a process of converting Cipher (encrypted image) into Original image.Cryptography uses the decryption technique at the receiver side to obtain the original message from non-readable message.

5. Digital Image Encryption Based On Advanced Encryption Standard(AES) Algorithm

ABSTRACT-With the rapid development of network and communication technology, digital image communication has become an important way of information transmission.Therefore, much more attention has been paid to the development of the digital image encryption technology.In this paper ,we propose a digital image encryption technology based on AES algorithm,and the algorithm implementation in MATLAB.Then,we perform digital image processing,obtain the date that can use the AES encryption algorithm,combine both approaches.Then,the digital image can be encrypted,and the algorithm is realised in MATLAB simulation.Through the comparison of the histogram analysis and the analysis of the key,the result has showed that the method can better realise the effect of encryption and decryption.

<u>EXISTING MODEL:</u>

Some of the existing image encryption techniques are -

Wavelet Transform -This method basis is the fact that one can compress the high-frequency part of an image retaining the low-
frequency part. Chaotic encryption is done for low-frequency wavelet coefficients. XOR operation is applied so that information in high-frequency wavelet coefficients is hidden. The current wavelet analysis uses Mallet algorithm wherein the low-frequency component is decomposed continuously. They also use Logistic chaotic map for encryption. This method is robust for noise attack.

Arnold Transform - In this method, the receiver needs to select and send a reference image in addition to the original image which needs to be secure. The final encrypted image will look similar to the reference image. This reference image is double the size of the image needed to be encrypted. Discrete wavelet transform can also be used to encrypt the image. The scheme proposed by V M Manikandan and V Masilamani is used in such a way that the pixel values of the image before encryption are in the range of (0-255). Arnold transform has a special periodic property which ensures that after j (j < maximum of length, breadth of the image) iterations, the scrambled matrix will be transformed into the original one. The usefulness of this method is that the encrypted image will look like a natural image without producing the conventional noise-like image. And the receiver gets his/herimage from the actual looking image.

DNA Sequence Operation - This method is proposed by Xiuli Chai, Yran Chen, Lucie Vroyde. A DNA sequence consists of four nucleic acid bases, i.e. A (Adenine), C (Cytosine), G (Guanine) and T (Thymine). A, T and G, C are complementary. As zero and one are complementary in a binary system, 00 and 11, 10 and 01 are complementary. Eight of the 24 type encoding rules satisfy Watson and Crick complementary rule. This method deals with DNA encoding to encode the image. SHA-256 hash of the plane image is used to generate the external secret key. Permutation of pixels is executed followed by DNA level diffusion. This method is resistant to all kinds of brute force attacks, entropy, and differential attacks.

Rubik Cube Transform- This method proposed by Raniprima, Hidayat, Nur Andini data hiding is done by encryption with Rubik Cube Transform followed by Stenography. Two keys are randomly generated. Rows and columns of the image are circularly shifted based on the keys generated. The encrypted image is then embedded into a cover image. This method is resistant to brute-force attack and the histograms of the original and encrypted images bear no resemblance.

PROPOSED MODEL:

Image encryption decryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, and military communication. Since these images may carry highly confidential information, these images entail extreme protection when users amass somewhere over an unreliable repository. As such we are going to show how the images are secured and protected using encryption and decryption using symmetric key methods by encrypting the image with a symmetric key and decrypting the encrypted image to decrypted image by using the key which is confidential.

Three algorithms are used for image encryption and decryption ,

1.**ADVANCED ENCRYPTION STANDARD (AES) ALGORITHM**

The AES algorithm is a symmetrical block cipher algorithm that takes input image in blocks of 128 bits and converts them to ciphered image using keys of 128, 192, and 256 bits.

2.**DATA ENCRYPTION STANDARD (DES) ALGORITHM**

The DES Algorithm is a block cipher. It uses symmetric keys to convert 64-bit image blocks into 48-bit ciphered image blocks.
DES encryption algorithm uses symmetric keys, which means that the same key is used for encrypting and decrypting the data.

3. **TRIPLE DES ALGORITHM**

Triple DES is a block cipher that applies the DES algorithm thrice. It usually uses three different keys—k1, k2, and k3. The first key, k1, is used to encrypt, the second key, k2, is used to decrypt, and the third key, k3, is used to encrypt again.
The triple DES also has a variant that uses only two keys, where k1 and k3 are the same.

OUTCOMES OF THE PROPOSED WORK:

We propose in the Triple Des model image encryption and decryption of size 10 MB .Both encryption and decryption are done with same cipher key and encrypted image will be in text format.The text format is then decrypted back into the original image using the secret key.In AES and DES model the image is encrypted into an invalid format using a symmetric key of 64 bits and 128 bits in DES and AES respectively and then it is decrypted back into the original image by using the symmetric key given in the code.

MERITS AND DEMERITS OF THE MODEL:

MERITS:

1. Website is easy to implement (and accelerate) in both hardware and software.

2. Image can be encrypted in both .png and .jpeg

3. Model is believed to be secured up.

DEMERITS:

1.Takes time to encrypt and decrypt when image is large.

2. It is a 64 bit and 128 bit cipher key.So,the chiper text length is limited.

3.For AES and DES model the password must be given in the code.

CODE:

1)TRIPLE DES:

Dasboard.html:

```
<!DOCTYPE html>

<html>

<head>

  <title>Welcome</title>

  <link rel="stylesheet" href="./css/style_dashboard.css">

</head>

<body>

<div class="container" onclick="onclick">

    <div class="top"></div>

    <div class="bottom"></div>

    <div class="center">

      <h2 style="font-size: 30px; text-align: center;">Triple DES</h2>

      <button onclick="location.href='./encrypt.html'">Encrypt</button>

      <button onclick="location.href='./decrypt.html'">Decrypt</button>

      <h2>&nbsp;</h2>
    </div>
  </div>
</body>
</html>
```

Decrypt.html:

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <script src="./js/crypto-js.js"></script>
  <meta charset="UTF-8">
  <title>Decrypt</title>
  <link rel="stylesheet" href="./css/style.css">
</head>
<body>
  <h1 id="title">Decrypt</h1>
  <div class="zone">
    <div id="dropZ">
      <i class="fa fa-cloud-upload"></i>
      <div>Drag and drop your file here</div>
      <span>OR</span>
      <div class="selectFile">
        <label for="file">Select file</label>
        <input type="file" name="files[]" id="file" onchange="previewFile()">
      </div>
      <p>File size limit : 10 MB</p>
    </div>
  </div>

  <div class="zone-image">
    <textarea id="content" class="result" rows="20" cols="50"></textarea>
    <label>
      <input type="text" id="secret" placeholder="Enter your secret key">
      <button onclick="decryptFromDES()">Decrypt</button>
    </label>
  </div>

  <script>
    function previewFile() {
        const content = document.querySelector('.result');
        const [file] = document.querySelector('input[type=file]').files;
        const reader = new FileReader();
        reader.addEventListener("load", () => {
            content.innerText = reader.result;
        }, false);

        if (file) {
            reader.readAsText(file);
        }
    }
```

```
   function decryptFromDES() {

     var secretKey = document.getElementById('secret').value;

     if(secretKey != ''){

       var fileName = "decrypted_image";

       var encryptedText = document.getElementById('content').value;

       try {

         var decryptedText = CryptoJS.TripleDES.decrypt(encryptedText,
secretKey).toString(CryptoJS.enc.Utf8);

         var element = document.createElement('a');

         element.setAttribute('href', decryptedText);

         element.setAttribute('download', fileName);

         element.style.display = 'none';

           document.body.appendChild(element);

         element.click();

         document.body.removeChild(element);

       }

       catch(err) {

         alert("There is an error in your document or your secret key is wrong!");

       }

     }

     else {

       alert("Please enter your secret key!")

     }

   }

  </script>
```

```
</body>

</html>
```

Encrypt.html:

```
<!DOCTYPE html>

<html lang="en" >

<head>

  <script src="./js/crypto-js.js"></script>

  <script src="./js/FileSaver.js"></script>

  <meta charset="UTF-8">

  <title>Encrypt</title>


  <link rel="stylesheet" href="./css/style.css">


</head>

<body>

  <h1 id="title">Encrypt</h1>

  <div class="zone">

    <div id="dropZ">

      <i class="fa fa-cloud-upload"></i>

      <div>Drag and drop your file here</div>

      <span>OR</span>

      <div class="selectFile">

        <label for="file">Select file</label>

        <input type="file" accept="image/*" name="files[]" id="file"
onchange="loadFile(event)">
```

```
    </div>
    <p>File size limit : 10 MB</p>
  </div>
</div>

<div class="zone-image">
  <img id="output" width="400" style="margin-top: 30px; position: relative;"/>
  <label>
    <input type="text" id="secret" placeholder="Enter your secret key">
    <button onclick="encryptToDES()">Encrypt</button>
  </label>
</div>

<script>
  var data;
  var loadFile = function(event) {
                    var image = document.getElementById('output');
                    image.src = URL.createObjectURL(event.target.files[0]);
  };
  var img = document.getElementById('output');
  img.crossOrigin = 'Anonymous';
  img.onload = function(){
      var canvas = document.createElement('canvas');
      var ctx = canvas.getContext('2d');
      canvas.height = this.naturalHeight;
```

```
        canvas.width = this.naturalWidth;

        ctx.drawImage(this,0,0);

        data = canvas.toDataURL('image/jpeg');

    };

    function encryptToDES() {

      var secretKey = document.getElementById('secret').value;

      if(secretKey != ''){

        var fileName = "encrypted_image";

        var plainText = data;

        var encryptedText = CryptoJS.TripleDES.encrypt(plainText, secretKey);

        saveAs(new File([encryptedText], fileName, {type: "text/plain;charset=utf-8"}));

      }

      else {

        alert("Please enter your secret key!");

      }

    }

  </script>

</body>

</html>
```

<u>2)DES:</u>

```
package com.mycompany.imgencdec;


import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;
import java.security.Key;

import javax.crypto.Cipher;

import javax.crypto.CipherOutputStream;

import javax.crypto.spec.SecretKeySpec;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;


/*

 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license

 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this
template

 */


/**

 *

 * @author Aishwarya Ganesh

 */

public class ImageEncDec extends javax.swing.JFrame {


    /**

     * Creates new form ImageEncDec

     */

    public ImageEncDec() {

        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() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        file_path = new javax.swing.JTextField();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("Image Encryption and Decryption using DES and AES-ISAA
Project");

        jButton1.setText("Choose");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jButton2.setText("Encrypt");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jButton3.setText("Decrypt");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(134, 134, 134)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addComponent(file_path)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
                    .addComponent(jButton1)
                    .addGap(77, 77, 77))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 316,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(64, Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(19, 19, 19)
                    .addComponent(jButton2)
                    .addGap(54, 54, 54)
                    .addComponent(jButton3)
                    .addGap(0, 0, Short.MAX_VALUE))))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(30, 30, 30)
                .addComponent(jLabel1)
                .addGap(46, 46, 46)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton1)
                    .addComponent(file_path, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton2)
                .addComponent(jButton3))
            .addContainerGap(183, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     JFileChooser chooser=new JFileChooser();
    chooser.showOpenDialog(null);
    File f=chooser.getSelectedFile();
    file_path.setText(f.getAbsolutePath());
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
       FileInputStream file=new FileInputStream(file_path.getText());
       FileOutputStream outStream=new FileOutputStream("Encrypt1.jpg");
       byte[] k="Abcdefgh".getBytes();
       SecretKeySpec key=new SecretKeySpec(k, "DES");
       Cipher enc=Cipher.getInstance("DES");
       enc.init(Cipher.ENCRYPT_MODE, (Key) key);
       CipherOutputStream cos=new CipherOutputStream(outStream,enc);
```

```
        byte[] buf=new byte[1024];
        int read;
        while((read=file.read(buf))!=-1){
            cos.write(buf,0,read);


        }
        file.close();
        outStream.flush();
        cos.close();
        JOptionPane.showMessageDialog(null,"The image is encrypted");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        FileInputStream file=new FileInputStream(file_path.getText());
        FileOutputStream outStream=new FileOutputStream("Decrypt1.jpg");
        byte[] k="Abcdefgh".getBytes();
        SecretKeySpec key=new SecretKeySpec(k, "DES");
        Cipher enc=Cipher.getInstance("DES");
        enc.init(Cipher.DECRYPT_MODE, (Key) key);
        CipherOutputStream cos=new CipherOutputStream(outStream,enc);
```

```
            byte[] buf=new byte[1024];

            int read;

            while((read=file.read(buf))!=-1){

                cos.write(buf,0,read);


            }

            file.close();

            outStream.flush();

            cos.close();

            JOptionPane.showMessageDialog(null,"The image is decrypted");

        }

        catch(Exception e){

            JOptionPane.showMessageDialog(null, e);

        }


    }


    /**

     * @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(ImageEncDec.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ImageEncDec.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ImageEncDec.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ImageEncDec.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        }
        //</editor-fold>

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

    // Variables declaration - do not modify
    private javax.swing.JTextField file_path;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

<u>3)AES:</u>

```
package com.mycompany.igencdec1;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.security.Key;

import javax.crypto.Cipher;

import javax.crypto.CipherOutputStream;

import javax.crypto.spec.SecretKeySpec;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;


/**
 *
 * @author Aishwarya Ganesh
 */
public class ImageEncDecAES extends javax.swing.JFrame {


    /**
     * Creates new form ImageEncDecAES
     */
    public ImageEncDecAES() {
        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() {

    file_path = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jButton1.setText("Choose");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Encrypt");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jButton3.setText("Decrypt");
```

```
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        jLabel1.setText("Image Encryption and Decryption AES-ISAA");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(77, 77, 77)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 267,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jButton3))
                    .addComponent(file_path, javax.swing.GroupLayout.PREFERRED_SIZE,
174, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                  .addComponent(jButton1)))
              .addContainerGap(56, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(35, 35, 35)
                .addComponent(jLabel1)
                .addGap(44, 44, 44)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(file_path, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1))
                .addGap(39, 39, 39)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton2)
                    .addComponent(jButton3))
                .addContainerGap(122, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>
```

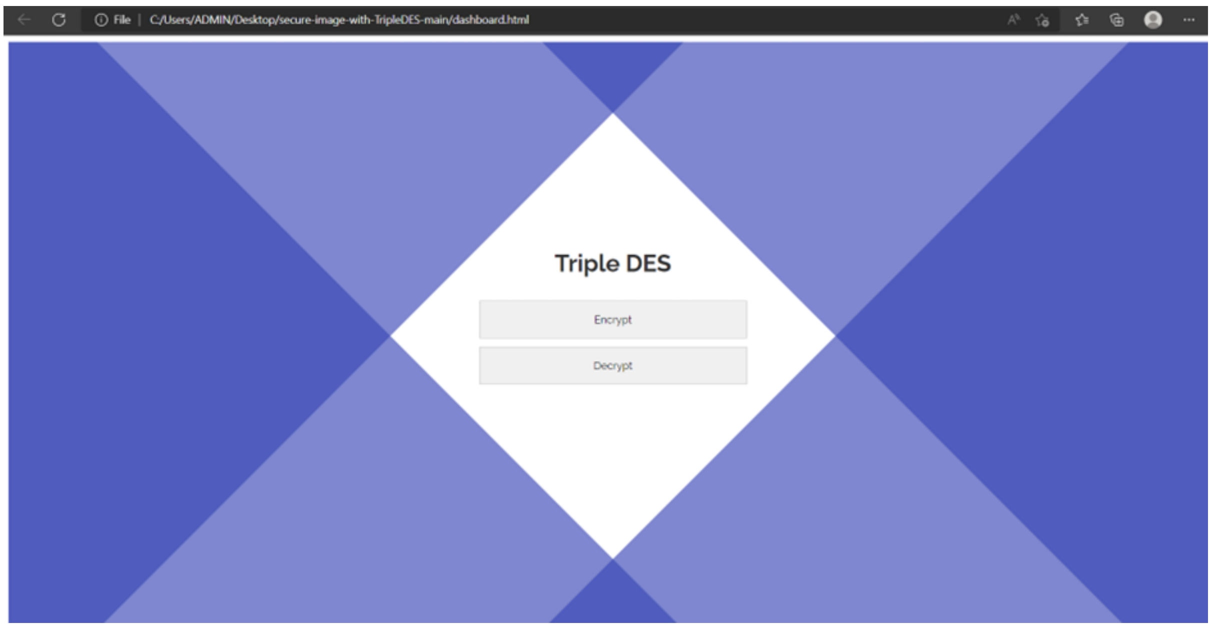
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser chooser=new JFileChooser();
    chooser.showOpenDialog(null);
    File f=chooser.getSelectedFile();
    file_path.setText(f.getAbsolutePath());
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     try{
       FileInputStream file=new FileInputStream(file_path.getText());
       FileOutputStream outStream=new FileOutputStream("Encrypt2.jpg");
       byte[] k="Abcdefgh12345678".getBytes();
       SecretKeySpec key=new SecretKeySpec(k, "AES");
       Cipher enc=Cipher.getInstance("AES");
       enc.init(Cipher.ENCRYPT_MODE, (Key) key);
       CipherOutputStream cos=new CipherOutputStream(outStream,enc);
       byte[] buf=new byte[1024];
       int read;
       while((read=file.read(buf))!=-1){
           cos.write(buf,0,read);

       }
       file.close();
```

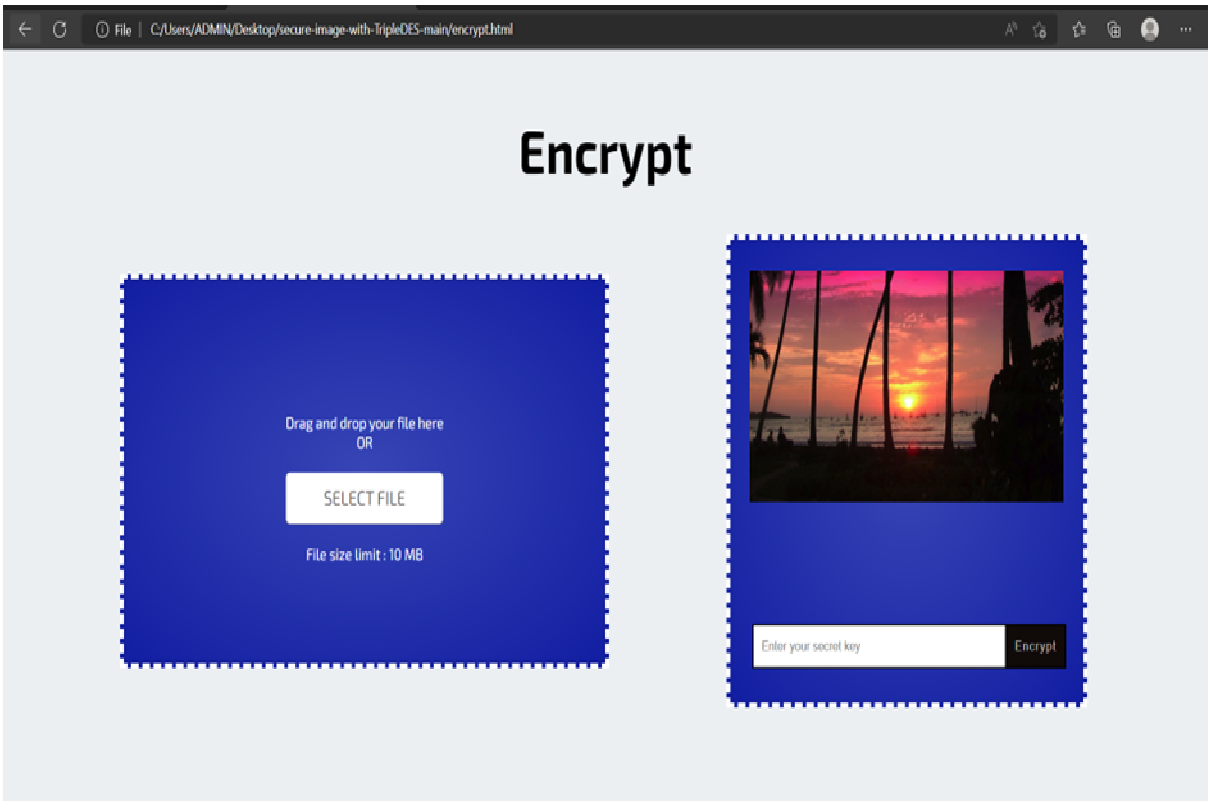
```
            outStream.flush();

            cos.close();

            JOptionPane.showMessageDialog(null,"The image is encrypted");

        }

        catch(Exception e){

            JOptionPane.showMessageDialog(null, e);

        }

    }


    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        try{

            FileInputStream file=new FileInputStream(file_path.getText());

            FileOutputStream outStream=new FileOutputStream("Decrypt2.jpg");

            byte[] k="Abcdefgh12345678".getBytes();

            SecretKeySpec key=new SecretKeySpec(k, "AES");

            Cipher enc=Cipher.getInstance("AES");

            enc.init(Cipher.DECRYPT_MODE, (Key) key);

            CipherOutputStream cos=new CipherOutputStream(outStream,enc);

            byte[] buf=new byte[1024];

            int read;

            while((read=file.read(buf))!=-1){

                cos.write(buf,0,read);


            }

            file.close();
```

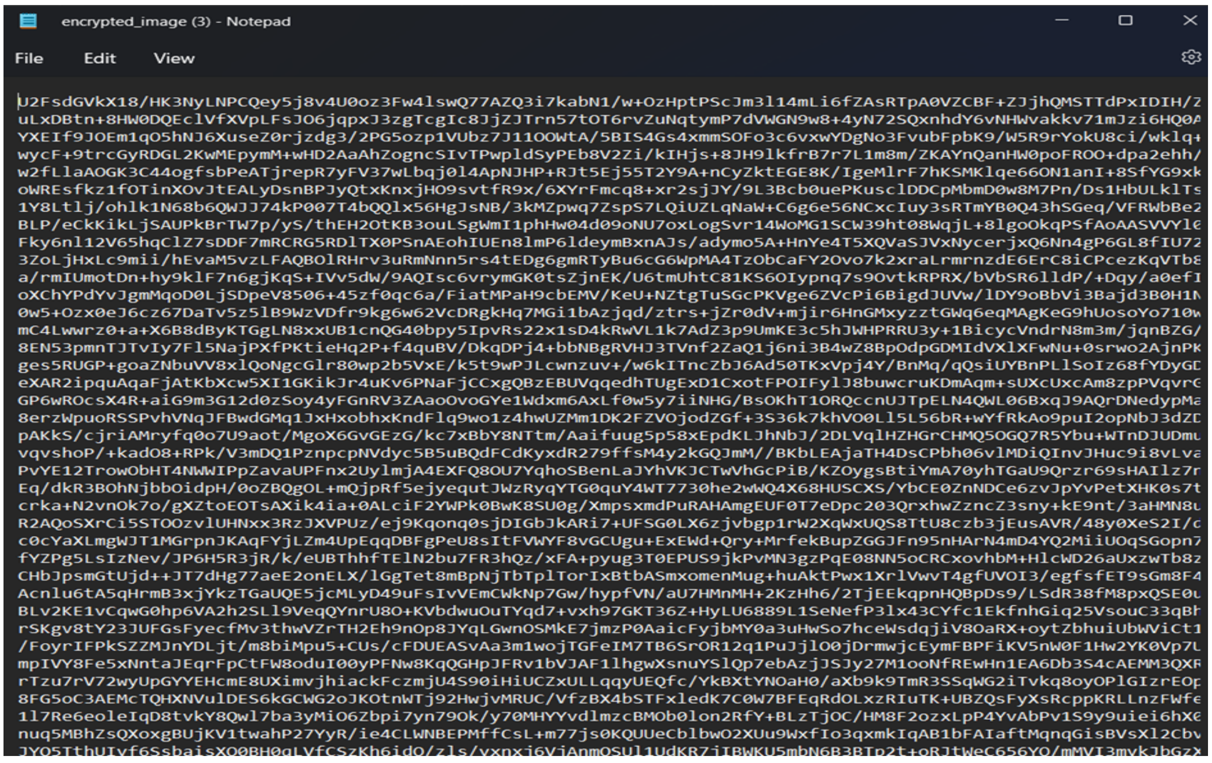
```
          outStream.flush();

          cos.close();

          JOptionPane.showMessageDialog(null,"The image is decrypted");

        }

        catch(Exception e){

          JOptionPane.showMessageDialog(null, e);

        }


    }


    /**

     * @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(ImageEncDecAES.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ImageEncDecAES.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ImageEncDecAES.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ImageEncDecAES.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 ImageEncDecAES().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
```

```
    private javax.swing.JTextField file_path;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

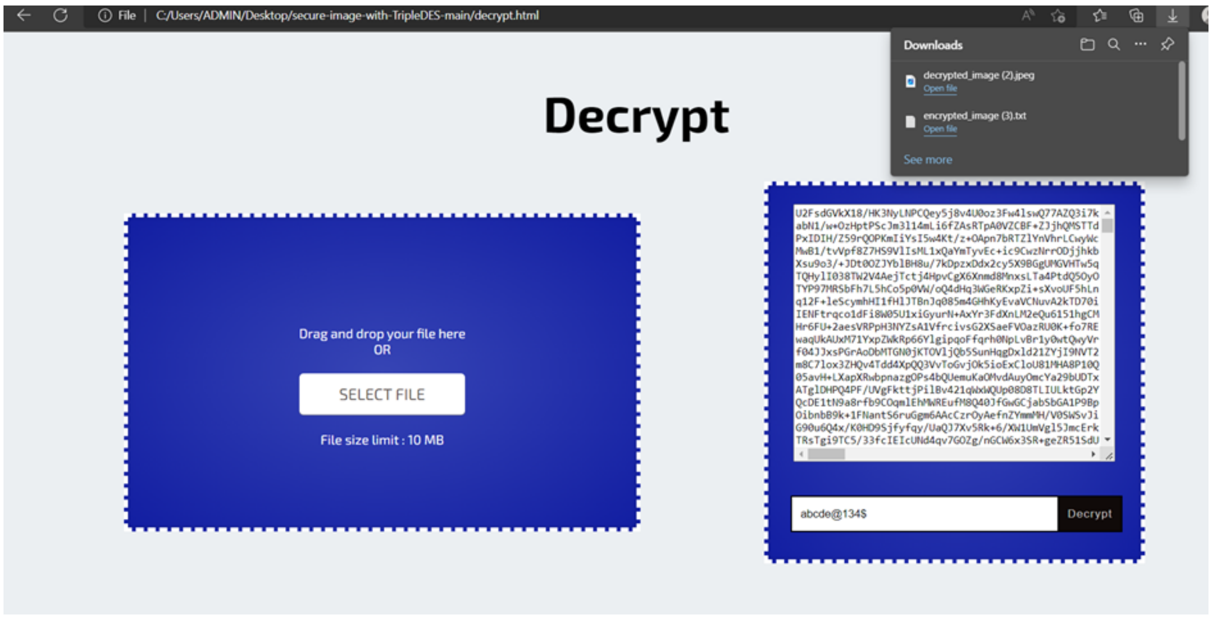
Screenshot of the output:


File | C:/Users/ADMIN/Desktop/secure-image-with-TripleDES-main/dashboard.html
Triple DES
Encrypt
Decrypt


File | C:/Users/ADMIN/Desktop/secure-image-with-TripleDES-main/encrypt.html
Encrypt
Drag and drop your file here
OR
SELECT FILE
File size limit : 10 MB
Enter your secret key
Encrypt

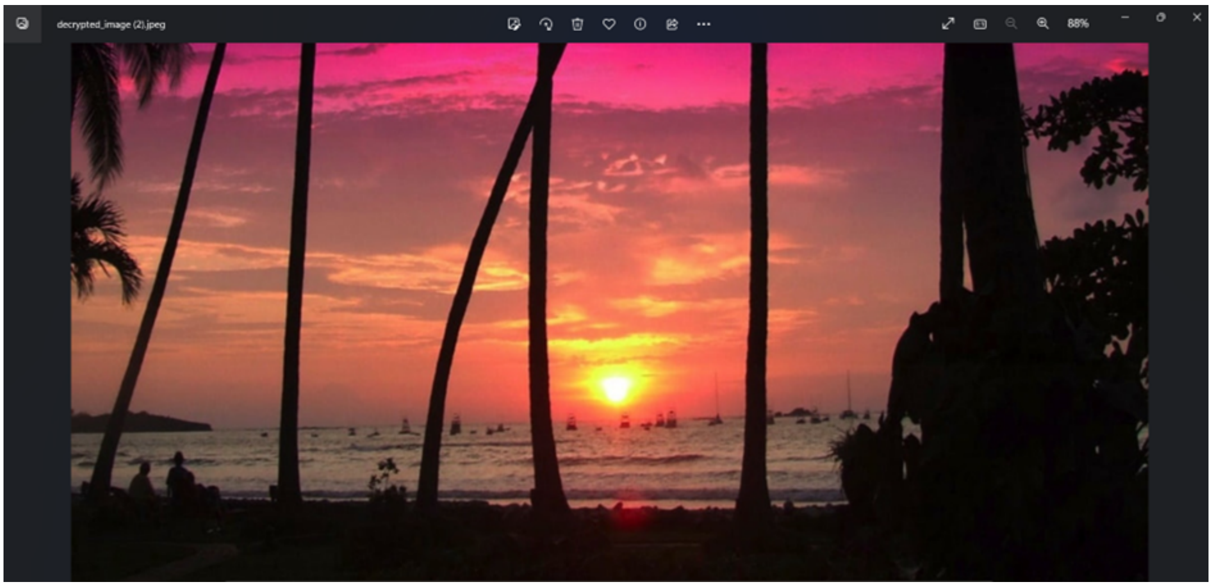


```
U2FsdGVkX18/HK3NyLNPCQey5j8v4U0oz3Fw4lswQ77AZQ3i7kabN1/w+OzHptPScJm3l14mLi6fZAsRTpA0VZCBF+ZJjhQMSTTdPxIDIH/Z
uLxDBtn+8HW0DQEclVfXVpLFsJO6jqpxJ3zgTcgIc8JjZJTrn57tOT6rvZuNqtymP7dVWGN9w8+4yN72SQxnhdY6vNHWvakkv71mJzi6HQ0A
YXEIf9JOEm1qO5hNJ6XuseZ0rjzdg3/2PG5ozp1VUbz7J11OOWtA/5BIS4Gs4xmmSOFo3c6vxwYDgNo3FvubFpbK9/W5R9rYokU8ci/wklq+
wycF+9trcGyRDGL2KwMEpymM+wHD2AaAhZogncSIvTPwpldSyPEb8V2Zi/kIHjs+8JH9lkfrB7r7L1m8m/ZKAYnQanHW0poFROO+dpa2ehh/
w2fLlaAOGK3C44ogfsbPeATjrepR7yFV37wLbqj0l4ApNJHP+RJt5Ej55T2Y9A+nCyZktEGE8K/IgeMlrF7hKSMKlqe66ON1anI+8SfYG9xk
oWREsfkz1fOTinXOvJtEALyDsnBPJyQtxKnxjHO9svtfR9x/6XYrFmcq8+xr2sjJY/9L3Bcb0uePKusclDDCpMbmD0w8M7Pn/Ds1HbULklTs
1Y8Ltlj/ohlk1N68b6QWJJ74kP007T4bQQlx56HgJsNB/3kMZpwq7ZspS7LQiUZLqNaW+C6g6e56NCxcIuy3sRTmYB0Q43hSGeq/VFRWbBe2
BLP/eCkKikLjSAUPkBrTW7p/yS/thEH2OtKB3ouLSgWmI1phHw04d09oNU7oxLogSvr14WoMG1SCW39ht08WqjL+8lgoOkqPSfAoAASVVYl
Fky6nl12V65hqClZ7sDDF7mRCRG5RDlTX0PSnAEohIUEn8lmP6ldeymBxnAJs/adymo5A+HnYe4T5XQVaSJVxNycerjxQ6Nn4gP6GL8fIU72
3ZoLjHxLc9mii/hEvaM5vzLFAQBOlRHrv3uRmNnn5rs4tEDg6gmRTyBu6cG6WpMA4TzObCaFY2Ovo7k2xraLrmrnzdE6ErC8iCPcezKqVTb8
a/rmIUmotDn+hy9klF7n6gjKqS+IVv5dW/9AQIsc6vrymGK0tsZjnEK/U6tmUhtC81KS6OIypnq7s9OvtkRPRX/bVbSR6lldP/+Dqy/a0efI
oXChYPdYvJgmMqoD0LjSDpeV8506+45zf0qc6a/FiatMPaH9cbEMV/KeU+NZtgTuSGcPKVge6ZVcPi6BigdJUVw/lDY9oBbVi3Bajd3B0H1N
0w5+Ozx0eJ6cz67DaTv5z5lB9WzVDfr9kg6w62VcDRgkHq7MGi1bAzjqd/ztrs+jZr0dV+mjir6HnGMxyzztGWq6eqMAgKeG9hUosoYo710w
mC4Lwwrz0+a+X6B8dByKTGgLN8xxUB1cnQG40bpy5IpvRs22x1sD4kRwVL1k7AdZ3p9UmKE3c5hJWHPRRU3y+1BicycVndrN8m3m/jqnBZG/
8EN53pmnTJTvIy7Fl5NajPXfPKtieHq2P+f4quBV/DkqDPj4+bbNBgRVHJ3TVnf2ZaQ1j6ni3B4wZ8BpOdpGDMIdVXlXFwNu+0srwo2AjnPK
ges5RUGP+goaZNbuVV8xlQoNgcGlr80wp2b5VxE/k5t9wPJLcwnzuv+/w6kITncZbJ6Ad50TKxVpj4Y/BnMq/qQsiUYBnPLlSoIz68fYDyGD
eXAR2ipquAqaFjAtKbXcw5XI1GKikJr4uKv6PNaFjCCxgQBzEBUVqqedhTUgExD1CxotFPOIFylJ8buwcruKDmAqm+sUXcUxcAm8zpPVqvr
GP6wROcsX4R+aiG9m3G12d0zSoy4yFGnRV3ZAaoOvoGYe1Wdxm6AxLf0w5y7iiNHG/BsOKhT1ORQccnUJTpELN4QWL06BxqJ9AQrDNedypMa
8erzWpuoRSSPvhVNqJFBwdGMq1JxHxobhxKndFlq9wo1z4hwUZMm1DK2FZVOjodZGf+3S36k7khVO0Ll5L56bR+wYfRkAo9puI2opNbJ3dZD
pAKkS/cjriAMryfq0o7U9aot/MgoX6GvGEzG/kc7xBbY8NTtm/Aaifuug5p58xEpdKLJhNbJ/2DLVqlHZHGrCHMQ5OGQ7R5Ybu+WTnDJUDmu
vqvshoP/+kadO8+RPk/V3mDQ1PznpcpNVdyc5B5uBQdFCdKyxdR279ffsM4y2kGQJmM//BKbLEAjaTH4DsCPbh06vlMDiQInvJHuc9i8vLva
PvYE12TrowObHT4NWWIPpZavaUPFnx2UylmjA4EXFQ8OU7YqhoSBenLaJYhVKJCTwVhGcPiB/KZOygsBtiYmA70yhTGaU9Qrzr69sHAIlz7r
Eq/dkR3BOhNjbbOidpH/0oZBQgOL+mQjpRf5ejyequtJWzRyqYTG0quY4WT7730he2wWQ4X68HUSCXS/YbCE0ZnNDCe6zvJpYvPetXHK0s7t
crka+N2vnOk7o/gXZtoEOTsAXik4ia+0ALciF2YWPk0BwK8SU0g/XmpsxmdPuRAHAmgEUF0T7eDpc203QrxhwZzncZ3sny+kE9nt/3aHMN8u
R2AQoSXrCi5STOOzvlUHNxx3RzJXVPUz/ej9Kqonq0sjDIGbJkARi7+UFSG0LX6zjvbgp1rW2XqWxUQS8TtU8czb3jEusAVR/48y0XeS2I/c
c0cYaXLmgWJT1MGrpnJKAqFYjLZm4UpEqqDBFgPeU8sItFVWYF8vGCUgu+ExEWd+Qry+MrfekBupZGGJFn95nHArN4mD4YQ2MiiUOqSGopn7
fYZPg5LsIzNev/JP6H5R3jR/k/eUBThhfTElN2bu7FR3hQz/xFA+pyug3T0EPUS9jkPvMN3gzPqE08NN5oCRCxovhbM+HlcWD26aUxzwTb8z
CHbJpsmGtUjd++JT7dHg77aeE2onELX/lGgTet8mBpNjTbTplTorIxBtbASmxomenMug+huAktPwx1XrlVwvT4gfUVOI3/egfsfET9sGm8F4
Acnlu6tA5qHrmB3xjYkzTGaUQE5jcMLyD49uFsIvVEmCWkNp7Gw/hypfVN/aU7HMnMH+2KzHh6/2TjEEkqpnHQBpDs9/LSdR38fM8pxQSE0u
BLv2KE1vCqwG0hp6VA2h2SLl9VeqQYnrU8O+KVbdwuOuTYqd7+vxh97GKT36Z+HyLU6889L1SeNefP3lx43CYfc1EkfnhGiq25VsouC33qBh
rSKgv8tY23JUFGsFyecfMv3thwVZrTH2Eh9nOp8JYqLGwnOSMkE7jmzP0AaicFyjbMY0a3uHwSo7hceWsdqjiV8OaRX+oytZbhuiUbWViCt1
/FoyrIFPkSZZMJnYDLjt/m8biMpu5+CUs/cFDUEASvAa3m1wojTGFeIM7TB6SrOR12q1PuJjlO0jDrmwjcEymFBPFiKV5nW0F1Hw2YK0Vp7L
mpIVY8Fe5xNntaJEqrFpCtFW8oduI00yPFNw8KqQGHpJFRv1bVJAF1lhgwXsnuYSlQp7ebAzjJSJy27M1ooNfREwHn1EA6Db3S4cAEMM3QXR
rTzu7rV72wyUpGYYEHcmE8UXimvjhiackFczmjU4S90iHiUCZxULLqqyUEQfc/YkBXtYNOaH0/aXb9k9TmR3SSqWG2iTvkq8oyOPlGIzrEOp
8FG5oC3AEMcTQHXNVulDES6kGCWG2oJKOtnWTj92HwjvMRUC/VfzBX4bSTFxledK7C0W7BFEqRdOLxzRIuTK+UBZQsFyXsRcppKRLLnzFWfe
1l7Re6eoleIqD8tvkY8Qwl7ba3yMiO6Zbpi7yn79Ok/y70MHYYvdlmzcBMOb0lon2RfY+BLzTjOC/HM8F2ozxLpP4YvAbPv1S9y9uiei6hX
nuq5MBhZsQXoxgBUjKV1twahP27YyR/ie4CLWNBEPMffCsL+m77js0KQUUeCblbwO2XUu9WxfIo3qxmkIqAB1bFAIaftMqnqGisBVsXl2Cbv
JYO5TthUIvf6SsbaisXO0BH0qLVfCSzKh6idO/zls/yxnxj6VjAnmOSUl1UdKR7jIBWKU5mbN6B3BTp2t+oRJtWeC656YO/mMVI3mykJbGzX
```

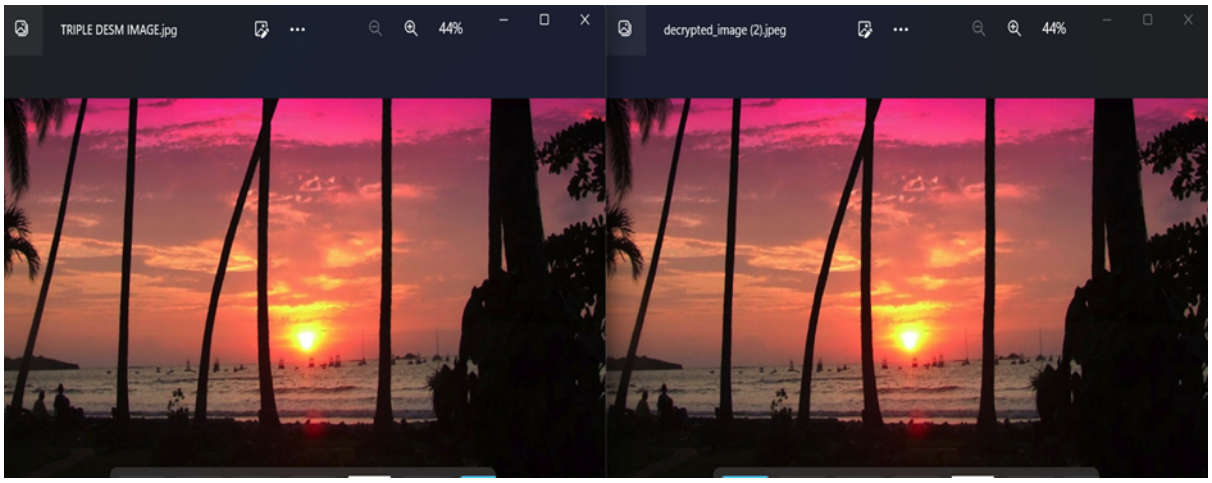


```
U2FsdGVkX18/HK3NyLNPCQey5j8v4U0oz3Fw4lswQ77AZQ3i7k
abN1/w+OzHptPScJm3l14mLi6fZAsRTpA0VZCBF+ZJjhQMSTTd
PxIDIH/Z59rQOPKmIiYsI5w4Kt/z+OApn7bRTZlYnVhrLCwyWc
MwB1/tvVpf8Z7HS9VlIsML1xQaYmTyvEc+ic9CwzNrrODjjhkb
Xsu9o3/+JDt0OZJYblBH8u/7kDpzxDdx2cy5X9BGgUMGVHTw5q
TQHylI038TW2V4AejTctj4HpvCgX6Xnmd8MnxsLTa4PtdQ5OyO
TYP97MRSbFh7L5hCo5p0VW/oQ4dHq3WGeRKxpZi+sXvoUF5hLn
q12F+leScymhHI1fHlJTBnJq085m4GHhKyEvaVCNuvA2kTD70i
IENFtrqco1dFi8W05U1xiGyurN+AxYr3FdXnLM2eQu6151hgCM
Hr6FU+2aesVRPpH3NYZsA1VfrcivsG2XSaeFVOazRU0K+fo7RE
waqUkAUxM71YxpZWkRp66YlgipqoFfqrh0NpLvBr1y0wtQwyVr
f04JJxsPGrAoDbMTGN0jKTOVljQb5SunHqgDxld21ZYjI9NVT2
m8C7lox3ZHQv4Tdd4XpQQ3VvToGvjOk5ioExCloU81MHA8P10Q
05avH+LXapXRwbpnazgOPs4bQUemuKaOMvdAuyOmcYa29bUDTx
ATglDHPQ4PF/UVgFkttjPilBv421qWxWQUp08D8TLIULktGp2Y
QcDE1tN9a8rfb9COqmlEhMWREufM8Q40JfGwGCjabSbGA1P9Bp
OibnbB9k+1FNantS6ruGgm6AAcCzrOyAefnZYmmMH/V0SWSvJi
G90u6Q4x/K0HD9Sjfyfqy/UaQJ7Xv5Rk+6/XW1UmVgl5JmcErk
TRsTgi9TC5/33fcIEIcUNd4qv7GOZg/nGCW6x3SR+geZR51SdU
```

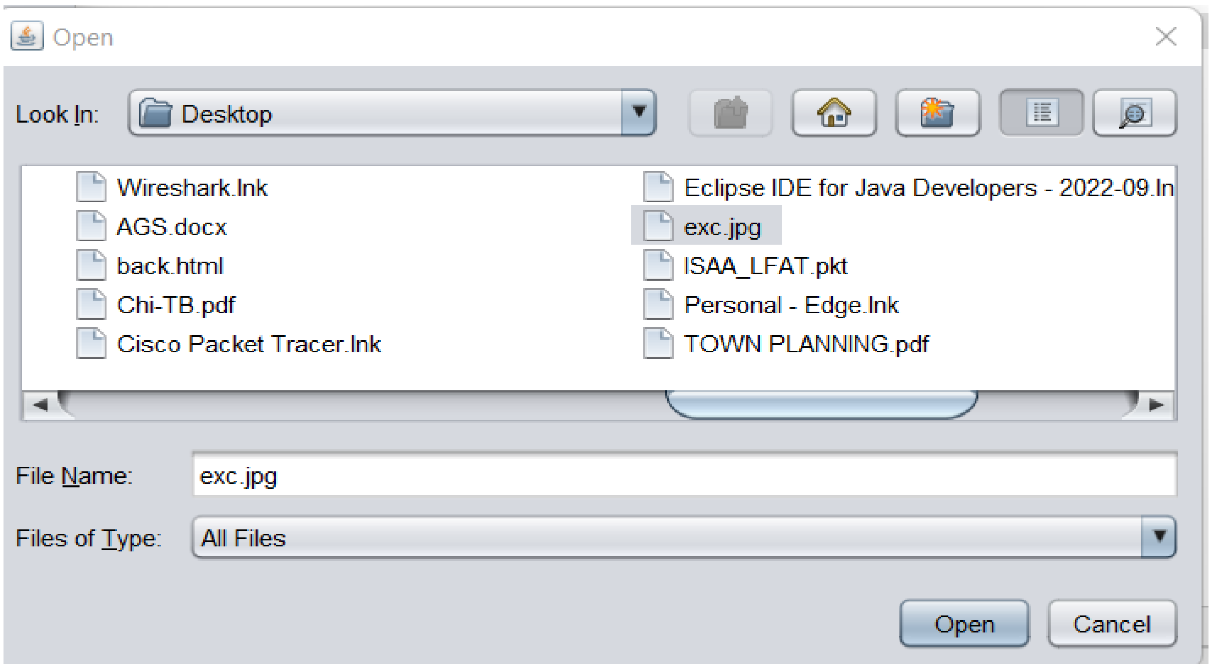
abcde@134$



Original image and Decrypted image:



DES:


Open

Look In: Desktop

Wireshark.lnk
AGS.docx
back.html
Chi-TB.pdf
Cisco Packet Tracer.lnk
Eclipse IDE for Java Developers - 2022-09.ln
exc.jpg
ISAA_LFAT.pkt
Personal - Edge.lnk
TOWN PLANNING.pdf

File Name: exc.jpg

Files of Type: All Files

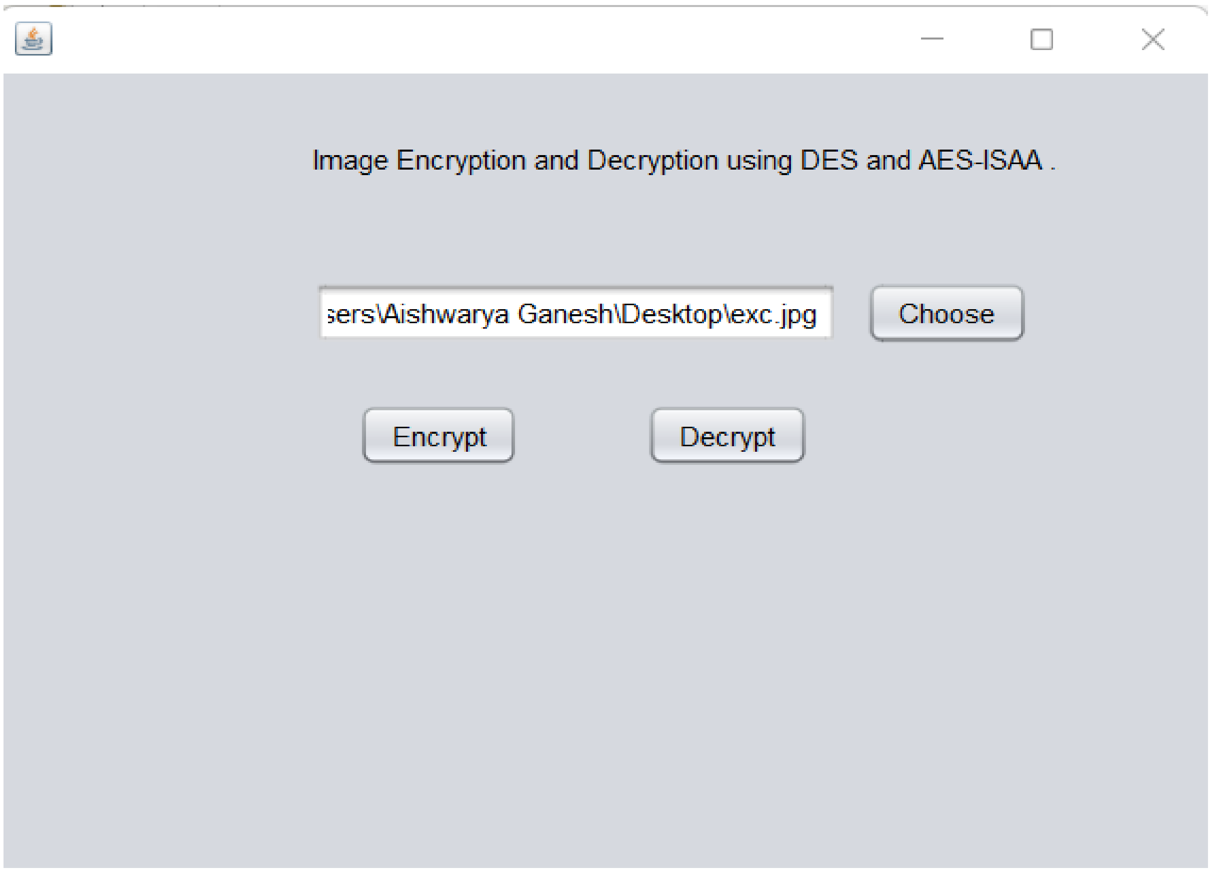


Image Encryption and Decryption using DES and AES-ISAA .

sers\Aishwarya Ganesh\Desktop\exc.jpg

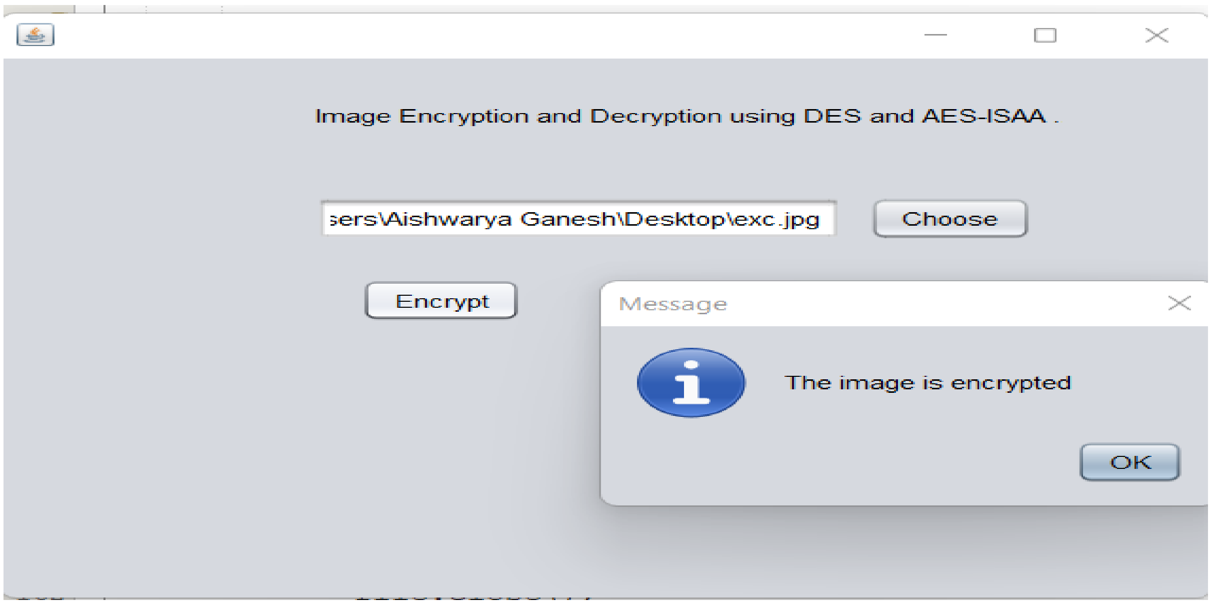
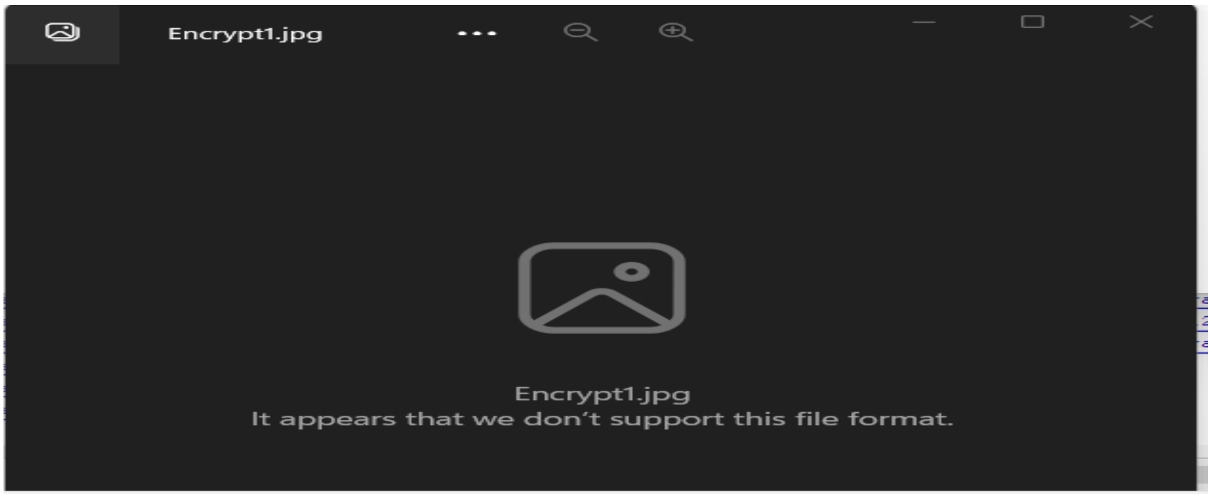


Image Encryption and Decryption using DES and AES-ISAA .

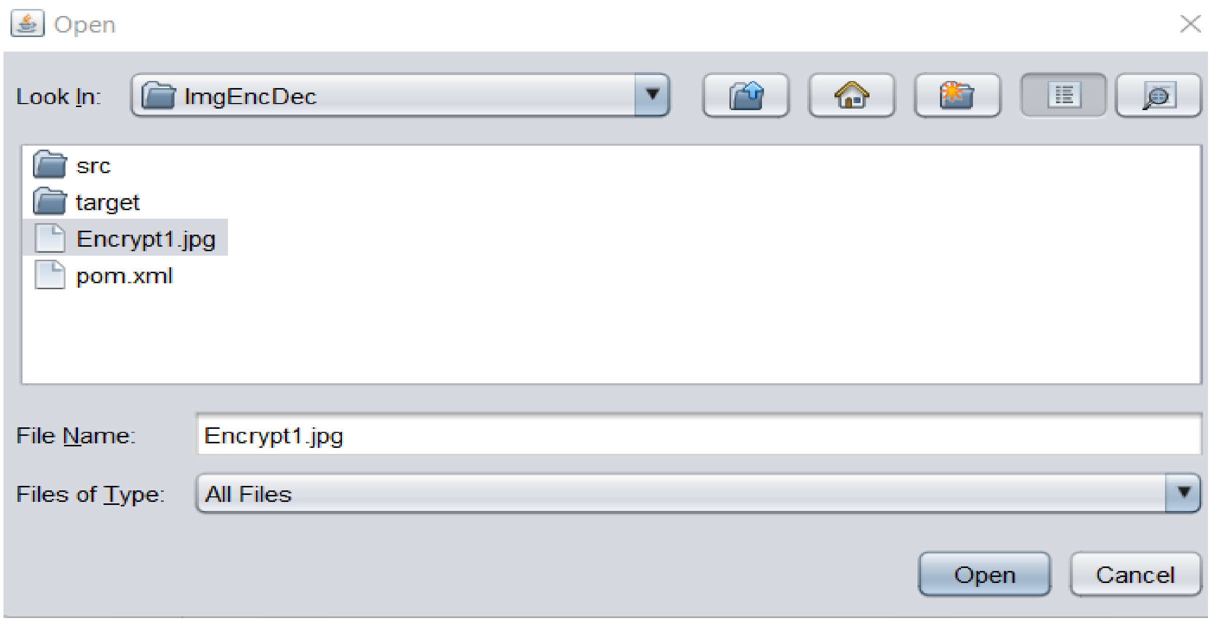
sers\Aishwarya Ganesh\Desktop\exc.jpg

Message

The image is encrypted


Encrypt1.jpg

Encrypt1.jpg
It appears that we don't support this file format.


Open

Look In: ImgEncDec

src
target
Encrypt1.jpg
pom.xml

File Name: Encrypt1.jpg

Files of Type: All Files

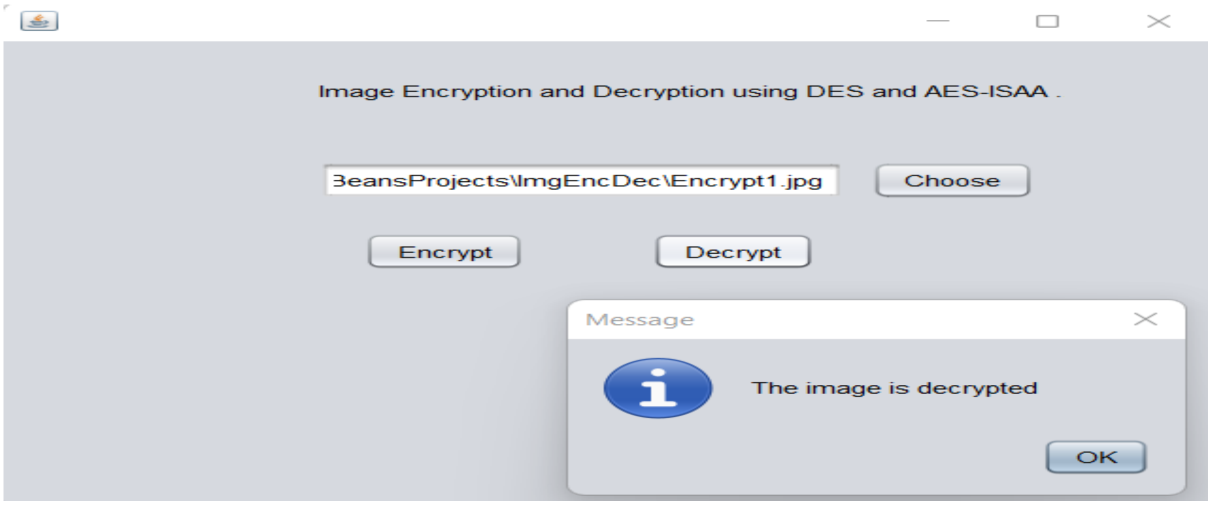
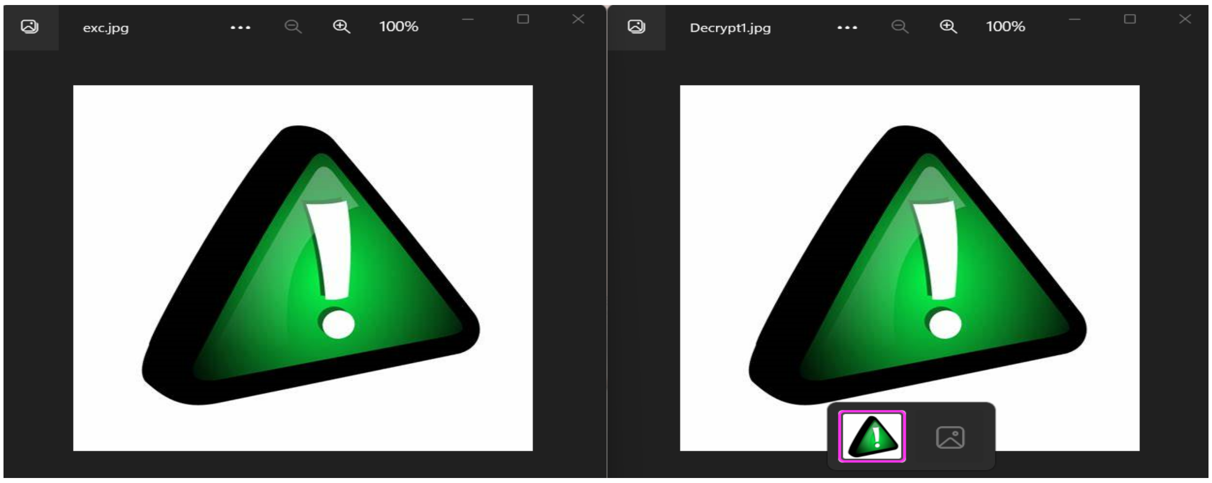


Image Encryption and Decryption using DES and AES-ISAA .

3eansProjects\ImgEncDec\Encrypt1.jpg

The image is decrypted



AES:

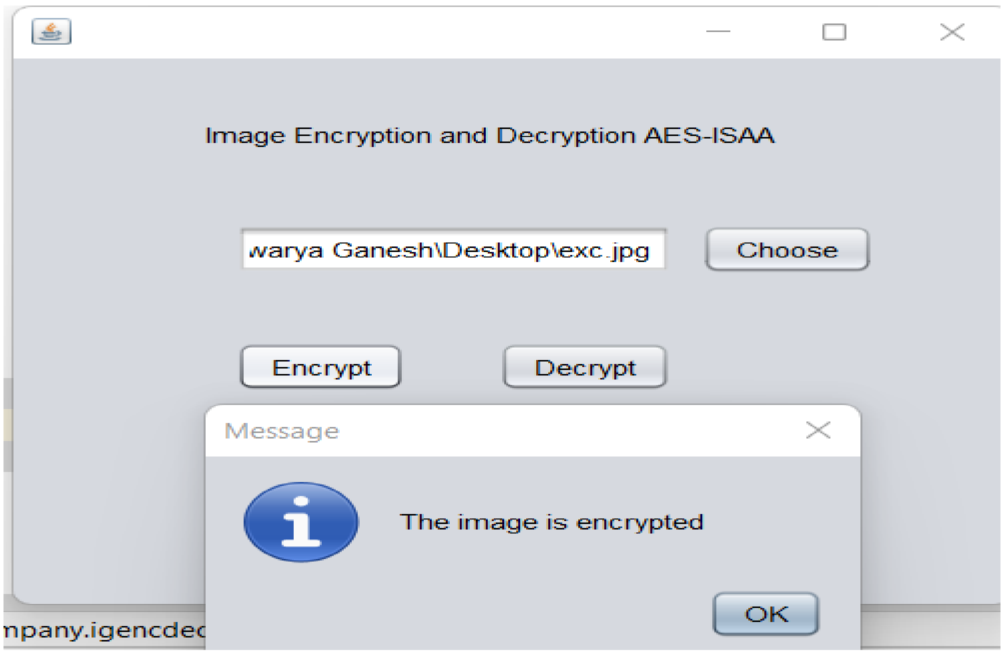
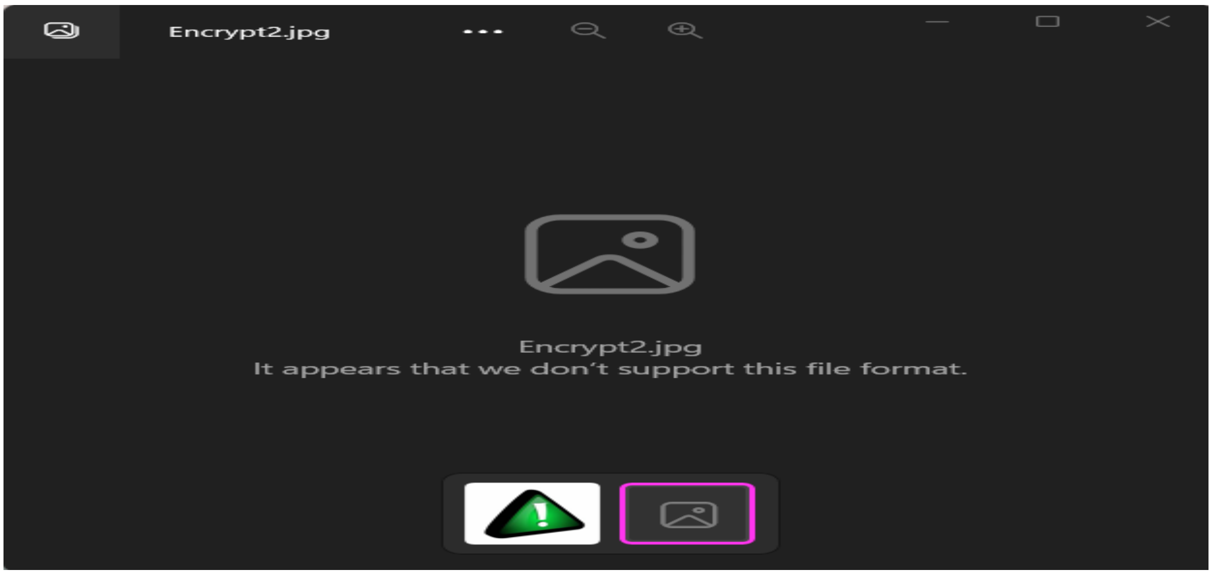


Image Encryption and Decryption AES-ISAA

warya Ganesh\Desktop\exc.jpg

The image is encrypted

npany.igencdec



Encrypt2.jpg
It appears that we don't support this file format.

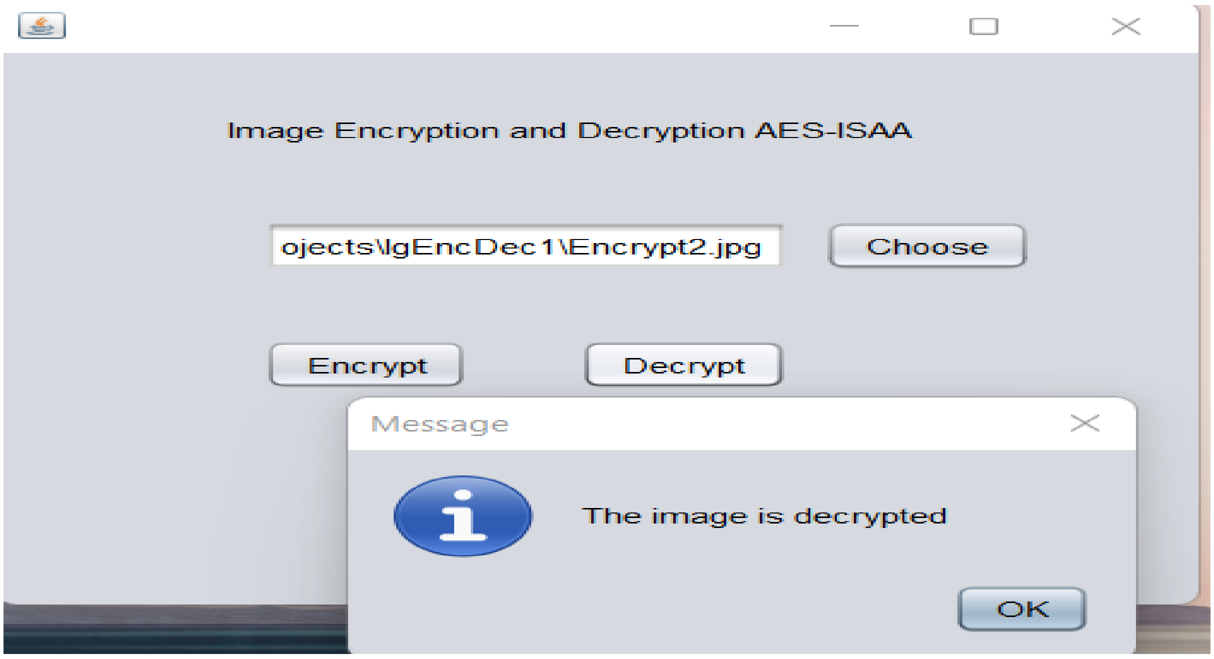
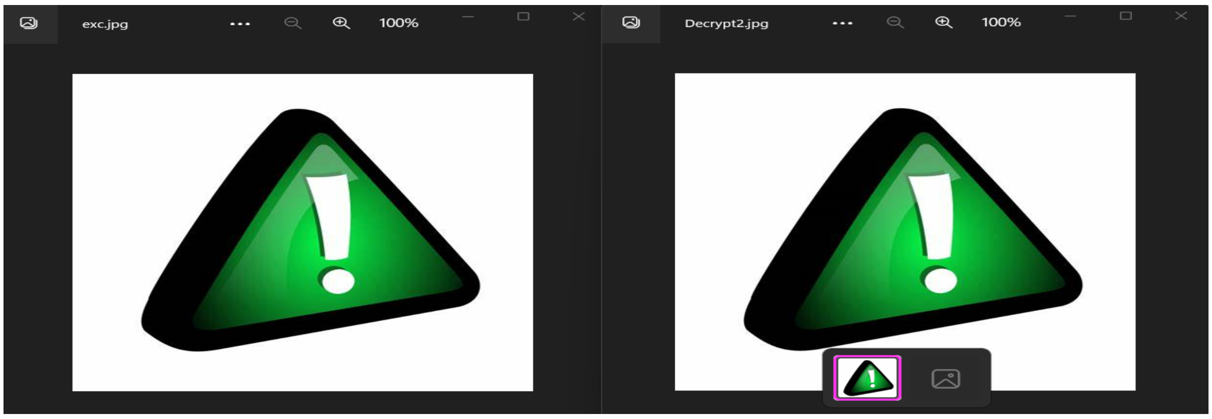


Image Encryption and Decryption AES-ISAA

ojects\IgEncDec1\Encrypt2.jpg

The image is decrypted



FUTURE WORK:

Future work which can be carried out in this field are developing quantum-safe cryptography to counter efforts to crack encrypted data using quantum computers.

CONCLUSION:

Image Encryption and Decryption is used to protect your identity and privacy. If you are ever being watched, you can hide your data securely by using well-implemented encryption and decryption systems.For secure transmission of data in open network, encryption is very important methodology. With encryption we can prevent our data from unauthorized access during transmission. In recent years many image encryption methods have been proposed and used to protect confidential data.

The purpose of file and disk encryption is to protect data stored on a computer or network storage system.Algorithms like AES,DES and 3DES can be used for secure transfer of information.

REFERENCES:

1. https://www.ijitee.org/wp-content/uploads/papers/v8i6s4/F10950486S419.pdf
2. https://www.ijtsrd.com/computer-science/computer-security/49837/survey-on-graphical-password-by-image-segmentation-20212022/eshita-agrawal
3. https://www.ijser.in/archives/v2i11/SjIwMTM0MDM=.pdf
4. https://ieeexplore.ieee.org/abstract/document/7030724/authors#authors
5. https://www.researchgate.net/publication/338778272_Review_of_Image_Encryption_Techniques