

SOURCE CODE

Main Windows:

```
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTabbedPane;
import javax.swing.SwingConstants;

import javax.swing.WindowConstants;

/** It Provides Full GUI for User to Perform Hiding and Unhiding
 */
public class StegoStick extends javax.swing.JFrame {

    {

        //Set Look & Feel
        try {

            javax.swing.UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLo
            okAndFeel");

        } catch(Exception e) {

            e.printStackTrace();

        }

    }

}
```

```
}
```

```
private JLabel titleLabel;
```

```
private JPanel unhidingPanel;
```

```
private JPanel hidingPanel;
```

```
private JTabbedPane containerPane;
```

```
// Main Method
```

```
public static void main(String[] args) {
```

```
    StegoStick inst = new StegoStick();
```

```
    inst.setVisible(true);
```

```
}
```

```
public StegoStick() {
```

```
    super("Digital Steganography");
```

```
    setLocation(300,300);
```

```
    setResizable(false);
```

```
    initGUI();
```

```
}
```

```
private void initGUI() {
```

```

try {
    setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    getContentPane().setLayout(null);
    {
        titleLabel = new JLabel();
        getContentPane().add(titleLabel);
        titleLabel.setText("Digital Steganography");
        titleLabel.setFont(new java.awt.Font("Baskerville Old
Face",1,20));

        titleLabel.setHorizontalAlignment(SwingConstants.CENTER);
        titleLabel.setBounds(0, 0, 595, 49);
    }
    {
        containerPane = new JTabbedPane();
        getContentPane().add(containerPane);
        containerPane.setBounds(0, 49, 595, 371);
        containerPane.setTabPlacement(JTabbedPane.LEFT);
        containerPane.setFont(new java.awt.Font("Corbel",0,16));

        {
            hidingPanel = new HideGUI();
            containerPane.addTab("Hiding", null, hidingPanel, null);
            hidingPanel.setPreferredSize(new
java.awt.Dimension(459,371));

            hidingPanel.setOpaque(false);

```

```

        unhidingPanel = new UnhideGUI();
        containerPane.addTab("UnHiding", null, unhidingPanel,
null);

        unhidingPanel.setOpaque(false);
    }

}

pack();
this.setSize(610, 461);
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

Hide Message:

```
import javax.swing.*;
```

```
public class Hide{
```

```
    HidePanel parent;
```

```
    String msg;
```

```
    String secretFileName;
```

```
    String coverFileName;
```

```
    String dstFileName;
```

```

String password;

String eVal;

String nVal;

int encryptTechnique;


Splash splashScreen;


boolean isConverted = false;

    boolean isHideFile = true;

    int coverType=3;

    int BMP = 1, MEDIA= 2, OTHER=3;

    ConvertToBMP c;


/* Native Methods implemented in C++ */

    public static native int hideInBMP(String secretFile, String coverFile, String dstFile,
String Password);

    public static native int hideInMEDIA(String secretFile, String coverFile, String dstFile,
String password);

    public static native int hideInOTHER(String secretFile, String coverFile, String dstFile,
String password);


    public Hide(HidePanel parent){

        this.parent = parent;

    }

    public void hideUsingRSA(String msg, String secretFileName, String coverFileName,
String dstFileName, String eVal, String nVal){

```

```
this.msg = msg;
this.secretFileName = secretFileName;
this.coverFileName = coverFileName;
this.dstFileName = dstFileName;
this.eVal = eVal;
this.nVal = nVal;
parent.encryptTechnique = 3; // RSA

if(!verifyData())
    return;

splashScreen = new Splash(2); // Processing.jpg

// Encrypt the File
Encryptor cryptor = new Encryptor();

if(cryptor.encrypt(this.secretFileName, eVal, nVal) == 0){
    JOptionPane.showMessageDialog(parent,"Invalid Secret File",
                                   "Error", JOptionPane.ERROR_MESSAGE);
    parent.secretField.setText("");
    return;
}

// New Encrypted File
this.secretFileName = cryptor.getOutputFileName();

// create password for Stego
password = nVal;
```

```

        // Now Hide the Data

        hide();
    }

    public void hideUsingPassword(String msg, String secretFileName, String
coverFileName, String dstFileName, String password, int encryptTechnique){

        this.msg = msg;

        this.secretFileName = secretFileName;

        this.coverFileName = coverFileName;

        this.dstFileName = dstFileName;

        this.password = password;

        this.encryptTechnique = encryptTechnique;

        if(!verifyData())

            return;

        splashScreen = new Splash(2); // Processing.jpg

        try{

            Thread.sleep(200);

        }catch(Exception ex){

        }

        // Encrypt the File

        Cryptor cryptor = new Cryptor();

        cryptor.setAlgorithm( encryptTechnique );

```

```

if(cryptor.encrypt(this.secretFileName, password) == 1){
    JOptionPane.showMessageDialog(parent,"Invalid Secret File",
                                   "Error", JOptionPane.ERROR_MESSAGE);

    parent.secretField.setText("");
    return;
}

// New Encrypted File

    this.secretFileName = cryptor.getOutputFileName();


    // Now Hide the Data

    hide();
}


// It verifies given data and covert Images to BMP
private boolean verifyData(){
    c = new ConvertToBMP();

    if(!secretFileName.equals("") && !msg.equals("")){
        JOptionPane.showMessageDialog(parent,"You are trying to hide
both Secret File and Secret Message"+
                                   "
\\nStegoStick is hiding Secret File only",

        "Warning",

        JOptionPane.WARNING_MESSAGE);

```



```

    }

    if(secretFileName.equals("")){
        // isHideFile = false;
        if(msg.equals("")){
            JOptionPane.showMessageDialog(parent,"Secret File Name or
Message not Entered",
                                        "Error", JOptionPane.ERROR_MESSAGE);
            return false;
        }
        // Convert message to file.XXXX
        secretFileName = FileParser.toFile(msg);
    }

    if(coverFileName.equals("")){
        JOptionPane.showMessageDialog(parent,"Cover File Name not Entered",
                                        "Error", JOptionPane.ERROR_MESSAGE);
        return false;
    }

    if(dstFileName.equals("")){
        JOptionPane.showMessageDialog(parent,"Destination File Path not Entered",
                                        "Error", JOptionPane.ERROR_MESSAGE);
        return false;
    }

    if(parent.encryptTechnique != SelectionPanel.RSA){

```

```

        if(password.equals("")){
            JOptionPane.showMessageDialog(parent,"password not Entered",
                                         "Error", JOptionPane.ERROR_MESSAGE);

            return false;
        }
    }
    else{
        if(eVal.equals("")){
            JOptionPane.showMessageDialog(parent,"E Value not Entered",
                                         "Error", JOptionPane.ERROR_MESSAGE);

            return false;
        }
        if(nVal.equals("")){
            JOptionPane.showMessageDialog(parent,"N Value not Entered",
                                         "Error", JOptionPane.ERROR_MESSAGE);

            return false;
        }
    }
}

```

```

        // If image is not BMP image convert to temporary BMP image for
Hiding
        try{
            if((coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".jpg"))
            ||

```

```

        (coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".jpeg")) ||
        (coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".gif"))
    ||

    coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".bmp")){
        if(!c.convertToBMP(coverFileName)){
            JOptionPane.showMessageDialog(parent,"Invalid Cover File",
                                           "Error", JOptionPane.ERROR_MESSAGE);
            parent.coverField.setText("");
            return false;
        }
        isConverted = true;
        coverFileName = new String("C:\\WINDOWS\\Temp\\temp.bmp");
        coverType = BMP;
    }
    else
    if(coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".wav") ||

    coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".avi") ||

    coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".mpg") ||

    coverFileName.substring(coverFileName.indexOf('.')).equalsIgnoreCase(".mpeg") )
        coverType = MEDIA;
    else
        coverType = OTHER;
    }catch(StringIndexOutOfBoundsException ex){

```

```

        JOptionPane.showMessageDialog(parent, "Invalid Cover File","Error",

JOptionPane.ERROR_MESSAGE);

        parent.coverField.setText("");
        return false;
    }

    return true;
}

private void hide(){
    int result = 0;
    String outFileName="";
    //  JOptionPane.showMessageDialog(null,"Hiding "+secretFileName);
    try{
        System.loadLibrary("StegBMP");
        System.loadLibrary("StegMEDIA");
        System.loadLibrary("StegOTHER");
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null,"Required DLLs Not
Found\n"+ex.getCause(),"Error Loading Libraries", JOptionPane.ERROR_MESSAGE);
    }

    if(coverType == BMP)
        // Hide file -- 1

```

```

        result = hideInBMP(secretFileName, coverFileName, dstFileName,
password);

        else if(coverType == MEDIA)

            result = hideInMEDIA(secretFileName, coverFileName, dstFileName,
password);

        else

            result = hideInOTHER(secretFileName, coverFileName, dstFileName,
password);

switch(result){

    case 0 : // Unknown Error during hiding

        JOptionPane.showMessageDialog(parent, "Unable to hide due to
unknown Error",

            "Error", JOptionPane.ERROR_MESSAGE);

        break;

    case 1 : // Invalid secret File Name

        JOptionPane.showMessageDialog(parent, "Invalid Secret File Name",

            "Error", JOptionPane.ERROR_MESSAGE);

        parent.secretField.setText("");

        break;

    case 2 : // Invalid cover File Name

        JOptionPane.showMessageDialog(parent, "Invalid Cover File Name",

            "Error", JOptionPane.ERROR_MESSAGE);

        parent.coverField.setText("");

        break;

    case 3 : // Capacity of cover file is less than secret file

        JOptionPane.showMessageDialog(parent, "Secret File exceeds capacity

```

of Cover File",

"Error", JOptionPane.ERROR_MESSAGE);

parent.clearFields();

break;

case 4 : // success

if(coverType == BMP)

outFileName = "steg.bmp";

else

outFileName

=

"steg"+coverFileName.substring(coverFileName.indexOf('.'));

JOptionPane.showMessageDialog(parent, "Secret File is successfully
hidden into cover file with resultant file "+outFileName ,

"Hiding

Successful",

JOptionPane.INFORMATION_MESSAGE);

parent.clearFields();

break;

case 5 : // Invalid Destination File Name

JOptionPane.showMessageDialog(parent, "Invalid Destination File Path",

"Error", JOptionPane.INFORMATION_MESSAGE);

parent.dstField.setText("");

break;

}

```
        if(isConverted){                                // if other images converted to
temporary BMP image
            c.deleteTemplImage();
        }
        splashScreen.dispose();

    }
}
```