Sample Code

1. **User view**

package User\_GUI;

import javax.swing.JOptionPane;

import Security.\*;

import Server.\*;

/\*

\* MainForm.java

\*

\* Created on November 3, 2007, 12:09 PM

\*/

import Stegnography.DembedForm;

import Stegnography.EmbedForm;

/\*\*

\*

\* @author Administrator

\*/

public class MainForm extends javax.swing.JFrame {

ServerThread st;

EncryptionForm ef;

DecryptionForm df;

DembedForm debf;

// CompressForm cf;

// DeCompressForm dcf;

EmbedForm ebf;

ReceiverClient rc;

/\*\* Creates new form MainForm \*/

public MainForm() {

initComponents();

st= new ServerThread();

Thread t=new Thread(st);

t.start();

}

/\*\* 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.

\*/

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jMenuBar1 = new javax.swing.JMenuBar();

jMenu1 = new javax.swing.JMenu();

jMenuItem9 = new javax.swing.JMenuItem();

jSeparator1 = new javax.swing.JSeparator();

jMenu3 = new javax.swing.JMenu();

jMenuItem3 = new javax.swing.JMenuItem();

jMenuItem4 = new javax.swing.JMenuItem();

jSeparator2 = new javax.swing.JSeparator();

jMenu4 = new javax.swing.JMenu();

jMenuItem5 = new javax.swing.JMenuItem();

jMenuItem6 = new javax.swing.JMenuItem();

jSeparator3 = new javax.swing.JSeparator();

jMenu5 = new javax.swing.JMenu();

jMenuItem10 = new javax.swing.JMenuItem();

jMenu2 = new javax.swing.JMenu();

jMenuItem8 = new javax.swing.JMenuItem();

jSeparator4 = new javax.swing.JSeparator();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Vedio Stegnography");

setResizable(false);

jMenu1.setText("Options");

jMenuItem9.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK\_X, java.awt.event.InputEvent.ALT\_MASK));

jMenuItem9.setText("Exit");

jMenuItem9.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem9ActionPerformed(evt);

}

});

jMenu1.add(jMenuItem9);

jMenu1.add(jSeparator1);

jMenuBar1.add(jMenu1);

jMenu3.setText("Security");

jMenuItem3.setText("Encrypt");

jMenuItem3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem3ActionPerformed(evt);

}

});

jMenu3.add(jMenuItem3);

jMenuItem4.setText("Decrypt");

jMenuItem4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem4ActionPerformed(evt);

}

});

jMenu3.add(jMenuItem4);

jMenu3.add(jSeparator2);

jMenuBar1.add(jMenu3);

jMenu4.setText("Steg Utiliity");

jMenuItem5.setText("Embed");

jMenuItem5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem5ActionPerformed(evt);

}

});

jMenu4.add(jMenuItem5);

jMenuItem6.setText("Dembed");

jMenuItem6.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem6ActionPerformed(evt);

}

});

jMenu4.add(jMenuItem6);

jMenu4.add(jSeparator3);

jMenuBar1.add(jMenu4);

jMenu5.setText("Send File");

jMenuItem10.setText("Send File");

jMenuItem10.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem10ActionPerformed(evt);

}

});

jMenu5.add(jMenuItem10);

jMenuBar1.add(jMenu5);

jMenu2.setText("Help");

jMenu2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenu2ActionPerformed(evt);

}

});

jMenuItem8.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK\_F1, 0));

jMenuItem8.setText("Help");

jMenuItem8.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jMenuItem8ActionPerformed(evt);

}

});

jMenu2.add(jMenuItem8);

jMenu2.add(jSeparator4);

jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(0, 680, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(0, 387, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem3ActionPerformed

// TODO add your handling code here:

ef=new EncryptionForm();

ef.setBounds(120,120,472, 477);

ef.setVisible(true);

}//GEN-LAST:event\_jMenuItem3ActionPerformed

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem4ActionPerformed

// TODO add your handling code here:

df=new DecryptionForm();

df.setBounds(120,120,472, 477);

df.setVisible(true);

}//GEN-LAST:event\_jMenuItem4ActionPerformed

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem5ActionPerformed

// TODO add your handling code here:

ebf=new EmbedForm();

ebf.setBounds(120,120,472, 384);

ebf.setVisible(true);

}//GEN-LAST:event\_jMenuItem5ActionPerformed

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem6ActionPerformed

// TODO add your handling code here:

debf=new DembedForm();

debf.setBounds(120,120,472, 299);

debf.setVisible(true);

}//GEN-LAST:event\_jMenuItem6ActionPerformed

private void jMenu2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenu2ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jMenu2ActionPerformed

private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem8ActionPerformed

// TODO add your handling code here:

String output="\n\nImportant info on Video Steganography \n\n"+

"1. Encrypt the file (\*.doc).\n" +

"2. Embed the Encrypted File (\*.\*) in to video File (\*.mpeg/\*.dat).\n" +

"3. Send the vedio File the required Host\n\n" +

"4. Must do the reverse process to get the Actual File\n\n"+

" Using this tool the data will be tranfered very easily and securely\n\n";

JOptionPane.showMessageDialog(null,output, "information",JOptionPane.INFORMATION\_MESSAGE);

}//GEN-LAST:event\_jMenuItem8ActionPerformed

private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem9ActionPerformed

// TODO add your handling code here:

System.exit(0);

}//GEN-LAST:event\_jMenuItem9ActionPerformed

private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jMenuItem10ActionPerformed

// TODO add your handling code here:

rc=new ReceiverClient();

}//GEN-LAST:event\_jMenuItem10ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

MainForm mf=new MainForm();

mf.setBounds(100,100,748, 350);

mf.setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JMenu jMenu1;

private javax.swing.JMenu jMenu2;

private javax.swing.JMenu jMenu3;

private javax.swing.JMenu jMenu4;

private javax.swing.JMenu jMenu5;

private javax.swing.JMenuBar jMenuBar1;

private javax.swing.JMenuItem jMenuItem10;

private javax.swing.JMenuItem jMenuItem3;

private javax.swing.JMenuItem jMenuItem4;

private javax.swing.JMenuItem jMenuItem5;

private javax.swing.JMenuItem jMenuItem6;

private javax.swing.JMenuItem jMenuItem8;

private javax.swing.JMenuItem jMenuItem9;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

// End of variables declaration//GEN-END:variables

}

1. Encryption and decryption Process

/\*

\* DBS.java

\*

\* Created on October 5, 2007, 3:40 PM

\*

\* To change this template, choose Tools | Template Manager

\* and open the template in the editor.

\*/

package enc;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* D i g i t a l E n c r y p t i o n S y s t e m \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.border.\*;

class filedialog extends JFrame

{

String name;

filedialog()

{

FileDialog fd=new FileDialog(filedialog.this,"Save as",FileDialog.SAVE);

fd.show();

if (fd.getFile()!=null)

{

name=fd.getDirectory()+fd.getFile();

setTitle(name);

}

}

public String getfile()

{

return name;

}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class DBS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

{

static String name;

public boolean DBST(int op,String x,javax.swing.JTextArea jLabel9)

{

boolean flag=false;

String toBeSaved = "";

String theText;

int choice=op;

name=x;

if(choice == 1)

{

// Get a file using a buffered reader and file input stream

// and stick it in to a byte array

// --------------------------------------------------------

try{

byte[] theFile = getFile(name);

// Ask the user for a key - this is what will encrypt the file

// I can't be arsed with all that buffered reader shit

// its too much hacking. JOptionPane is much easier

// It also adds to user security since having non

// terminal style input prevents some capturing programs

// from stealing the data

// -----------------------------------------------------------

String key = JOptionPane.showInputDialog("Enter your key (the longer the better):");

// This is an update from previous versions

// Decided it would be easier to put

// encryption stuff in to a class

// ----------------------------------------

Encryption encryption = new Encryption(theFile,key);

// encrypt file

// ------------

encryption.encrypt();

// get encrypted file bytes and save it

// ------------------------------------

toBeSaved=saveFile(encryption.getFileBytes());

jLabel9.setText(toBeSaved);

JOptionPane.showMessageDialog(null,"\nYour file has been encrypted and saved\n","message",JOptionPane.INFORMATION\_MESSAGE);

flag=true;}catch(Exception e){flag=false;jLabel9.setText("");e.printStackTrace();}

}

else if(choice == 2)

{

// Basically the reverse of the (choice == 1) clause above

// -------------------------------------------------------

try{

byte[] theFile = getFile(name);

JPasswordField pf = new JPasswordField("Enter the key: ");

String key = JOptionPane.showInputDialog("Enter the key: ");

Encryption encryption = new Encryption(theFile,key);

encryption.decrypt();

toBeSaved=saveFile(encryption.getFileBytes());

jLabel9.setText(toBeSaved);

JOptionPane.showMessageDialog(null,"\nYour file has been decrypted and saved\n","message",JOptionPane.INFORMATION\_MESSAGE);

flag=true;}catch(Exception e){flag=false;jLabel9.setText("");e.printStackTrace();}

}

else if(choice == 4)

{

System.out.println("GOOD DAY!");

System.exit(0);

}

else if(choice == 3)

{

String output="\n\nImportant info on key choice: \n\n"+

"The longer the key, the better. This program\n" +

"implements a key expansion algorithm that given\n" +

"an average length of user-entered key is almost\n" +

"analagous to the one-time pad encryption method\n\n" +

"For example: Use key length of 1: 128bit encryption\n\n"+

" Use key length of 2: 256bit encryption\n\n"+

" Use key length of 8: 1024bit encryption\n\n"+

" etc...\n\n\n";

JOptionPane.showMessageDialog(null,output, "information",JOptionPane.ERROR\_MESSAGE);

}

return flag;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* L o a d i n a f i l e \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public static byte[] getFile(String name)

{

byte[] readFromFile = null;

String txt=name;

try

{

FileInputStream in = new FileInputStream(txt);

readFromFile = new byte[in.available()];

in.read(readFromFile);

in.close();

}

catch(IOException e)

{

System.out.println("\nSorry - file not found!\n");

}

return readFromFile;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* S a v e a f i l e \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public static String saveFile(byte[] toSave)

{

String txt ;

filedialog fd=new filedialog();

txt=fd.getfile();

try

{

FileOutputStream out = new FileOutputStream(txt);

out.write(toSave);

out.close();

}

catch(IOException e)

{

System.out.println("Sorry, but there seems to have been a problem\n" +

"saving your file. Perhaps your hard-drive is full\n" +

"or the write permissions need to be changed\n");

}

return txt;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* C l a s s t h a t h a n d l e s e n c r y p t i o n\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Encryption

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

{

public Encryption(byte[] fileBytes,String key)

{

this.fileBytes = fileBytes;

this.key = key;

// Stick the key in to a character array

// This is so the file bytes can be offset with

// the characters as defined by the overall algortihm

// when either encrypting or decrypting

// --------------------------------------------------

keys = new char[key.length()];

pivot = (int)(fileBytes.length/2);

// These long values are just random bits of data junk

// that are added during the enctypion process

// to add to the overall scrambling capability

//

// -------------------------------------------------------------

delta = 0x9e3779b9;

alpha = 0x7f2637c6;

beta = 0x5d656dc8;

gamma = 0x653654d9;

// Shift the bits slightly (>> and << bitwise operators)

// as determined by characters in the key

// -----------------------------------------------------

sumA = (long)(alpha >> key.charAt(0));

sumB = (long)(beta << key.charAt(1));

sumC = (long)(gamma >> key.charAt(2));

sumD = (long)(delta >> key.charAt(3));

// Sorts out the problem of having an odd or even number of bits

// ----------------------------------------------------------------

if (fileBytes.length%5 > 0)

{

inter = (int)((fileBytes.length-1)/5);

}

else inter = (int)(fileBytes.length/5);

// forLevel2 is used in the level2 method

// --------------------------------------

forLevel2 = key.length();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* s o m e m e t h o d s \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public void setFileBytes(byte[] newBytes)

{

fileBytes = newBytes;

}

public byte[] getFileBytes()

{

return fileBytes;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* D o e s e x a c t l y w h a t \*/

/\* i t s a y s \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public void encrypt()

{

int f = 0;

boolean truth = true;

// Takes user key and makes a biger one

// for added security

// ------------------------------------

key = keyStream();

// probably don't need this bit as it is already defined in

// the constructor, but I thought I'd leave it in just in case

// -----------------------------------------------------------

keys = new char[key.length()];

for(int c = 0;c<key.length();c++)

{

keys[c] = key.charAt(c);

}

System.out.println("\nEncrypting\n");

// the outer for loop ensure that the algorithm

// loops round a lot of time, so that

// the file is encrypted mutliple times

// --------------------------------------------

for(int extra = 0;extra<127;extra++)

{

for(int i = 0;i<fileBytes.length;i = i + keys.length)

{

if (truth == false)

break;

f = 0;

for(int j = i;j<i+keys.length;j++)

{

if(j>=fileBytes.length)

{

truth = false;

break;

}

fileBytes[j] = (byte)((fileBytes[j] ^

(keys[f] - 'A' << sumD)) ^ (keys[f] + sumD));

sumD -= delta;

f++;

}

}

fileBytes = splitNSwap(fileBytes);

setFileBytes(fileBytes);

}

setFileBytes(level2(fileBytes,true));

}

public void decrypt()

{

setFileBytes(level2(fileBytes,false));

int f = 0;

boolean truth = true;

key = keyStream();

keys = new char[key.length()];

for(int c = 0;c<key.length();c++)

{

keys[c] = key.charAt(c);

}

System.out.println("\nDecrypting\n");

for(int extra = 0;extra<127;extra++)

{

fileBytes = getFileBytes();

fileBytes = splitNSwap(fileBytes);

for(int i = 0;i<fileBytes.length;i = i + keys.length)

{

if (truth == false)

break;

f = 0;

for(int j = i;j<i+keys.length;j++)

{

if(j>=fileBytes.length)

{

truth = false;

break;

}

fileBytes[j] = (byte)((fileBytes[j] ^

(keys[f] - 'A' << sumD)) ^ (keys[f] + sumD));

sumD -= delta;

f++;

}

}

setFileBytes(fileBytes);

}

}

// To add to the confusion, this method basically

// takes the byte[] array as encrypted so far

// splits it in half and then swaps two halves around

// i.e. a b c d e f would become d e f a b c

// --------------------------------------------------

public byte[] splitNSwap(byte[] zeBytes)

{

if(zeBytes.length%2==0)

{

pivot = (int)(zeBytes.length/2);

}

else pivot = (int)((zeBytes.length-1)/2);

fileBytez = new byte[zeBytes.length];

for(int reverse = 0;reverse<pivot;reverse++)

{

fileBytez[reverse] = (byte)(zeBytes[reverse+pivot]^fileBytez[reverse]);

}

for (int reverseB = pivot;reverseB<zeBytes.length;reverseB++)

{

fileBytez[reverseB] = (byte)(zeBytes[reverseB - pivot]^fileBytez[reverseB]);

}

setFileBytes(fileBytez);

return fileBytez;

}

// Problem:

// So far, if a long key is used and only some of the key is correct

// then first part of cipher text will still be decrypted

// so:

// (if you want to see the key that it produces uncomment print statement)

// -----------------------------------------------------------------------

public String scrambleKey(String toBeScrambledFurther)

{

pivot = (int)(toBeScrambledFurther.length()/2);

String newKey = "";

String sub1 = "", sub2 = "";

for (int a = 0; a<pivot;a++)

{

sub1 += toBeScrambledFurther.charAt(a+pivot);

}

for (int b = pivot; b<toBeScrambledFurther.length();b++)

{

sub2 += toBeScrambledFurther.charAt(b-pivot);

}

newKey = sub1+sub2;

//System.out.println(newkey);

return newKey;

}

/\*

\* shuffles around the fileBytes in blocks of key.length().

\* at present this algorithm still needs some heavy debugging

\* It is not presently powerful enough for commercial - class

\* encryption. It needs to be fixed....

\* The problem lies with the fact, that some times, when a short text-file

\* is encrypted, some of the 'central' text is still partially scrambled

\*/

// What is this function passed ?

// Answer - basically two parameters - the first is

// an array of bytes that need to be scrambled. The second

// is a boolean - true if the array is being scrambled

// and false if being descrambled etc.....

// -------------------------------------------------------

public byte[] level2(byte[] oldBytes, boolean state)

{

if(state)

System.out.println("Scrambling encrypted data");

else System.out.println("\nDescrambling encrypted data");

int s = forLevel2;

int stop = oldBytes.length%s;

// some of these variables and loops are currently obsolete

byte[] newBytes = new byte[oldBytes.length];

byte[] tempBytes = new byte[oldBytes.length-stop];

byte[] resultBytes = new byte[oldBytes.length];

byte[] remainderBytes = new byte[stop];

int hello = oldBytes.length-stop;

// obsolete loops

for (int old = 0;old<oldBytes.length-stop;old++)

{

tempBytes[old] = oldBytes[old];

}

for (int old = 0;old<stop;old++)

{

remainderBytes[old] = oldBytes[(oldBytes.length-stop+old)];

}

if (state)

{

for (int outer = 0;outer<s;outer++)

{

for (int c = outer;c<hello+outer;c+=s)

{

if(c+s<oldBytes.length)

{

newBytes[c] = (byte)(oldBytes[c+s]-sumA);

newBytes[c+s] = (byte)(oldBytes[c]+sumB);

}

else break;

}

}

}

else if (!state)

{

for (int outer = s-1;outer >=0;outer--)

{

for(int c = (hello-1-outer);c>=0-outer;c-=s)

{

if(c-s>=0)

{

newBytes[c-s] = (byte)(oldBytes[c]-sumB);

newBytes[c] = (byte)(oldBytes[c-s]+sumA);

}

else break;

}

if (outer <= 0)break;else continue;

}

}

// obsolete code

// experimental

for(int rep = 0;rep<newBytes.length;rep++)

{

resultBytes[rep]=newBytes[rep];

}

for(int rep = 0;rep<remainderBytes.length;rep++)

{

resultBytes[rep]=remainderBytes[rep];

}

setFileBytes(newBytes);

return newBytes;

}

/\* The keyStream() method takes the user key

\* and enlarges it to (key.length()\*key.length()+

\* key.length())\*128, in the following algorithmic method.

\* This improves security. I.e. longer keys are

\* much harder to crack

\*/

public String keyStream()

{

System.out.println("\nGenerating key stream\n");

String answer = key;

String thekey = key;

for(int i = 0;i<(thekey.length()\*128);i++)

{

answer = answer + getPart(thekey);

thekey = getPart(thekey);

}

//answer = scrambleKey(answer);

return answer;

}

// KeyStream helper method

// -----------------------

public String getPart(String thekey)

{

char[] keyPart = new char[thekey.length()];

String result = "";

for(int c = 0;c<thekey.length()-1;c++)

{

keyPart[c] = (char)(thekey.charAt(c+1) - 1);

}

keyPart[thekey.length()-1] = thekey.charAt(0);

for(int put = 0;put<keyPart.length;put++)

{

result = result + keyPart[put];

}

return result;

}

private String key;

private char[] keys;

private byte[] fileBytes;

private byte[] fileBytez;

private int pivot;

private int inter;

private long alpha;

private long beta;

private long gamma;

private long delta;

private long sumA;

private long sumB;

private long sumC;

private long sumD;

private byte[] fileBytesB;

private int forLevel2;

}

}

1. Embedding Process

package Stegnography;

import java.io.File;

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

import java.io.FileInputStream;

import java.io.FileOutputStream;

/\*\*

\*

\* @author Administrator

\*/

public class EmbProcess {

String embfilename;

public String emb(String s, String s1)

{

try{

File file = new File(s);

File file1 = new File(s1);

FileInputStream fileinputstream = new FileInputStream(s);

FileOutputStream fileoutputstream = new FileOutputStream("temp");

byte abyte0[] = new byte[8];

int i;

int k;

for(k = 0; (i = fileinputstream.read(abyte0, 0, 8)) > 0; k = i)

fileoutputstream.write(abyte0, 0, i);

fileinputstream.close();

for(int l = 1; l <= 8 - k; l++)

fileoutputstream.write(65);

fileoutputstream.write("DATAFILE".getBytes(), 0, 8);

System.out.println("File name==="+file1.getName());

StringBuffer stringbuffer = new StringBuffer(file1.getName());

stringbuffer.setLength(50);

fileoutputstream.write(stringbuffer.toString().getBytes(), 0, 50);

fileinputstream = new FileInputStream(s1);

int j;

while((j = fileinputstream.read(abyte0, 0, 8)) > 0)

fileoutputstream.write(abyte0, 0, j);

fileinputstream.close();

fileoutputstream.close();

file.delete();

File file2 = new File("temp");

file2.renameTo(file);

embfilename=file.getName();

}

catch(Exception e){

e.printStackTrace();

embfilename="";

}

return embfilename;

}

public String demb(String s)

{

boolean flag;

String demfile = "";

try

{

File file = new File(s);

String outpath=s.substring(0, s.lastIndexOf("\\")+1);

FileInputStream fileinputstream = new FileInputStream(s);

char c = '\b';

byte abyte0[] = new byte[c];

String s1 = "";

int i;

while((i = fileinputstream.read(abyte0, 0, c)) > 0)

{

s1 = new String(abyte0);

if(s1.equals("DATAFILE"))

break;

}

if(!s1.equals("DATAFILE"))

{

flag=false;

fileinputstream.close();

return demfile;

}

abyte0 = new byte[50];

fileinputstream.read(abyte0, 0, 50);

s1 = new String(abyte0);

String s2 = s1.trim();

String fpath = s2.substring(0, s2.lastIndexOf(".") + 1) + "enc";

System.out.println("fpath------"+fpath);

FileOutputStream fileoutputstream = new FileOutputStream(outpath+fpath);

c = '\u5000';

abyte0 = new byte[c];

while((i = fileinputstream.read(abyte0, 0, c)) > 0)

fileoutputstream.write(abyte0, 0, i);

fileinputstream.close();

fileoutputstream.close();

demfile=fpath;

}

catch(Exception exception)

{

demfile="";

exception.printStackTrace();

System.out.println(exception);

}

return demfile;

}

}

1. Deembedded Process

package Stegnography;

import java.awt.FileDialog;

/\*

\* CRCForm.java

\*

\* Created on November 3, 2007, 12:12 PM

\*/

import javax.swing.JFrame;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Administrator

\*/

public class DembedForm extends javax.swing.JFrame {

EmbProcess ep;

/\*\* Creates new form CRCForm \*/

public DembedForm() {

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.

\*/

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel6 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jButton3 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("VideoSteganoGraphy---DeEmbeding Video File");

setLocationByPlatform(true);

setResizable(false);

jLabel1.setFont(new java.awt.Font("Century", 1, 12));

jLabel1.setText("Select Video File");

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jButton1.setFont(new java.awt.Font("Century", 1, 12));

jButton1.setText("Browse");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel4.setFont(new java.awt.Font("Century", 1, 12));

jLabel4.setText("De-Embed File :");

jLabel5.setFont(new java.awt.Font("Century", 1, 12));

jButton2.setFont(new java.awt.Font("Century", 1, 12));

jButton2.setText("De-Embed");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jLabel6.setFont(new java.awt.Font("Century Gothic", 1, 18));

jLabel6.setText("De-Embeding Video File");

jLabel8.setFont(new java.awt.Font("Century", 1, 12));

jLabel8.setText("Video File :");

jLabel9.setFont(new java.awt.Font("Century", 1, 12));

jButton3.setFont(new java.awt.Font("Century", 1, 12));

jButton3.setText("Close");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.addContainerGap()

.add(jLabel1)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.UNRELATED)

.add(jTextField1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 197, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(18, 18, 18)

.add(jButton1))

.add(layout.createSequentialGroup()

.add(97, 97, 97)

.add(jButton2)

.add(46, 46, 46)

.add(jButton3))

.add(layout.createSequentialGroup()

.add(85, 85, 85)

.add(jLabel6, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 280, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.add(layout.createSequentialGroup()

.addContainerGap()

.add(jLabel8)

.add(18, 18, 18)

.add(jLabel5, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 330, Short.MAX\_VALUE))

.add(layout.createSequentialGroup()

.addContainerGap()

.add(jLabel4)

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jLabel9, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, 338, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(23, 23, 23)

.add(jLabel6, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 36, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(18, 18, 18)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jLabel1)

.add(jTextField1, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT\_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jButton1))

.add(35, 35, 35)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jButton2)

.add(jButton3))

.add(39, 39, 39)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jLabel8, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 24, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE)

.add(jLabel5, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 19, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)

.add(jLabel4)

.add(jLabel9, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE, 24, org.jdesktop.layout.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(45, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField1ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

// TODO add your handling code here:

FileDialog filedialog = new FileDialog(new JFrame(), "Select Audio File");

filedialog.setFile("\*.mpeg;");

String s = new String();

filedialog.setVisible(true);

filedialog.getFile();

jLabel5.setText(filedialog.getFile());

jTextField1.setText(filedialog.getDirectory() + filedialog.getFile());

//l6.setText(filedialog.getDirectory());

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

// TODO add your handling code here:

String s1 = jTextField1.getText();

ep=new EmbProcess();

String genfile = ep.demb(s1);

if(genfile!=null)

JOptionPane.showMessageDialog(null," De-Embed Process Completed");

else

JOptionPane.showMessageDialog(null," De-Embed Process Failed");

jLabel9.setText(genfile);

}//GEN-LAST:event\_jButton2ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton3ActionPerformed

// TODO add your handling code here:

this.dispose();

}//GEN-LAST:event\_jButton3ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new DembedForm().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JTextField jTextField1;

// End of variables declaration//GEN-END:variables

}