P-1817

DIGITAL RIGHTS PROTECTION FOR MP3 DATA

A PROJECT REPORT

Submitted by

RADHA.K	**71203104027**
SALINI.M	**71203104037**
SHANMUGA PRIYA.M	**71203104043**

in partial fulfillment for the award of the degree
of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE-641 006

P-1817

ANNA UNIVERSITY: CHENNAI 600 025
APRIL 2007

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"Digital Rights Protection for MP3 Data"** is the bonafide work of **"Radha.K, Salini.M, Shanmuga Priya.M"** who carried out the project under my supervision.

SIGNATURE
Prof. S. Thangasamy, Ph.D,
HEAD OF THE DEPARTMENT,
Department of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641 006.

SIGNATURE
Mrs.Amutha Venkatessh, M.E,
SUPERVISOR,
Senior Lecturer,
Department of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641 006.

The candidates with University Register Nos.: **71203104027,71203104037,71203104043** were examined by us in the project viva-voce examination held on 25/4/07.

25.4
INTERNAL EXAMINER

25/4/07
EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled **"DIGITAL RIGHTS PROTECTION FOR MP3 DATA"** is a record of original work done has not been submitted to Anna University or any institutions, for the fulfillment of course study.

The report is in partial fulfillment of the requirements for the award of the degree of Bachelor of Computer Science and Engineering of Anna University, Chennai.

Place: Coimbatore
Date: 21/04/07

Radha

(K. Radha)

M. Salini
(M. Salini)

(M. Shanmuga Priya)

ACKNOWLEDGEMENT

We are grateful to **Dr.Joseph V.Thanikal,** Principal, Kumaraguru College of Technology for having given us an opportunity to embark on this project.

We are obliged to **Dr.S.Thangasamy**, Ph.D., Head of the Department of Computer Science and Engineering for his guidance during the course of the project.

We also extend our thanks to our project coordinator **Ms.Rajini**,Assistant Professor, Department of Computer Science and Engineering for providing useful suggestions and corrections.

We are indebted to our project guide **Mrs.Amutha Venkatessh**, M.E, Senior Lecturer, Department of Computer Science and Engineering for her guidance and support given to us throughout this project.

We thank the teaching and non-teaching staffs of our Department for providing us the technical support in the duration of our project.

Above all, we divulge and offer our distinguished thanks for the great support and comfort given by our **LORD God Almighty** during dire situations.

We also take this opportunity to thank our beloved parents and dear friends for their inexpressible moral support.

ABSTRACT

The project entitled "Digital Rights Protection for MP3 Data" is developed and designed using the Microsoft's language C#. The main intention of this software is to determine the unethical party who distributes unauthorized copies of the multimedia contents.

"Digital rights" protection is a major issue in the e-commerce of the multimedia contents. Watermarking technology proposed is an enabling technology for the rights protection of the multimedia digital contents. In order to reinforce protection and access control this application has been developed. A unique watermark is embedded in each piece of the multimedia content before it is distributed to the customer. So the user who buys the multimedia contents from the content provider will have the watermarked multimedia content. When he tries to make an unauthorized copy of the multimedia content the duplicate copy will also have a unique watermark embedded in each piece of the multimedia contents. So when unauthorized copies of a piece of contents are found, the customer who owns the content will be readily identified by the embedded watermark and also the content provider who sells the content will have the details of the customer like customer's name, transaction dates and serial number. The piracy protection software that is developed has addressed this problem using cryptographic technologies. This software employs a commutative encryption algorithm to protect the privacy of watermarks. Information is doubly locked by two encryption keys kept separately by the customer and the content provider. In this application the customer will get only the watermarked content and the content provider will have no idea how the watermark is formed.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1Specific Subject Matter Of "Digital Rights Protection for MP3 Data "

Moving Pictures Experts Group audio layer 3 (MP3) compression format has gained popularity for distributing digital music over the internet. The MP3 format allows raw digital music to be compressed down to one-tenth or more in size, while still keeping reasonably high fidelity. Further with peer to peer applications, digital contents could be distributed over the internet without control. This promotes illegal acts like unauthorized copying and distributing of digital contents.

It is estimated that the music industry suffered an annual business loss of five billon dollars. This makes major music companies hesitated to sell music in digital formats over the internet. In order to reinforce protection and access control to digital contents, piracy protection software is designed. This software supports a security infrastructure for the tailor-made protected digital contents with digital rights to individual customer.

As significant portion of trades is expected to deal with multimedia contents, digital rights protection becomes an eminent issue in e-commerce.It supports rights information such as owner's identity, transaction dates and number.The watermark signals recover the rights information originally embedded in a piece of digital contents and therefore origins of pirate copies can be traced. Based on this model a novel watermarking software is proposed to address common problem in e-trading.

2. ANALYSIS OF THE PROBLEM

2.1. Problem Definition

Due to significant loss of sales in the recent years the music industry is looking for new distribution approaches to sell music content. One of these approaches is *music on demand*. Here the customer can choose individual audio tracks, which enables him to buy music more selectively than buying complete albums with a fixed selection of musical pieces. Digital watermarking is a technology capable of solving important practical security problems like authentication for copyright protection or manipulation recognition for integrity verification. When copies do not differ, source of illegal copies cannot be identified it is shown in fig 2.1.1

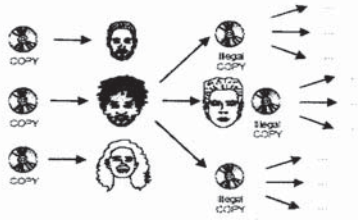


Fig 2.1.1when copies do not differ, Sources of illegal copies cannot be identified.
Figure Description: The above figure describes that the illegal copies cannot be identified when copies do not differ.

2.1.1. Two Methods of Music On Demand

Audio File Download

Here a customer can select and download individual audio tracks from an online shop. Usually he downloads the music in a lossy compression file format, like *mp3* or *wma*. The complete process is carried out online and requires reliable authentication for payment and high download rates for a comfortable access to the music.

CD-on-Demand

This is a hybrid approach where the selection of the musical pieces is done online via a web interface by the customer. Here the customer can select the musical contents according to his desire. After the selection of the tracks is done, a CD-R is burnt and sent to the customer by mail. Here the customer receives the music in full CD quality and on a physical storage object. After the customer receives the CD he may save it in his system. Naturally a delay for mail delivery occurs.

2.1.2. Tracking out the Loopholes

With respect to pirates who want to create illegal copies of the music, both of the above mentioned two methods share one big advantage compared to music distribution via mail order or music stores: The audio files can be watermarked before the customer receives it. This enables the personalization of the copies. When a customer buys a CD in a store, there is no way to trace him back if an illegal copy of this CD appears g. in a peer to peer network. If he buys via music on demand service, e.g. his customer ID could be embedded into the songs by digital audio watermarking. Information is doubly locked by two encryption keys kept separately by the customer and the content provider. In this application the customer will get only the watermarked content and the content provider will have no idea how the watermark is formed.

But embedding a customer ID is not promising in all music on demand scenarios. When the music on demand service is only an alternative to shop sales of the music, untraceable illegal copies will appear from ripped audio CDs. A good chance of stopping illegal copies is only achieved if the music on demand service is the *only* source of the music. With this given, no copies without an embedded customer ID should appear e.g. in peer to peer networks.

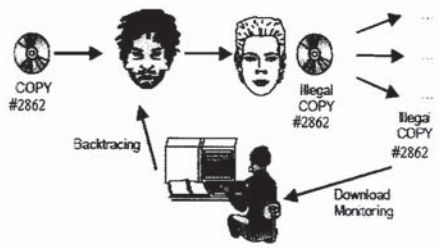


Fig 2.1.2 Individual copies can be traced back to the pirate

Figure Description: This figure describes that the individual copies can be traced back to the pirate by download monitoring.

2.2. System Analysis

The existing system has to be analyzed well for its merits and demerits. Our main aim is to make use of all its merits and to remove the demerits of it. The merits are to be carried out and the demerits are to be overcome in proposed system.

2.2.1 Existing System

In the previous versions of Digital rights protection, the images were mostly embedded with watermarks. Still there are a lot of techniques for embedding watermark in the multimedia content.

Watermarking the audio digital signals has not been developed with complete authorization and security. Existing system used Spatial Spread Spectrum Algorithm, where watermark contents are easily detected and removed.

2.2.2 Drawbacks of the Existing System

The drawbacks and difficulties that is faced in the existing system are analyzed and are presented below:

1.It consumes more time.

2.It is not feasible.

3.Less security and unauthorized.

4.Easily detected and removed.

2.2.3 Proposed System

Watermarking the audio data is done so as to protect its piracy of unauthorized persons –is a great solution to the multimedia content provider who owns the particular content.

In Watermarking algorithms certain parameters addressed are: Robustness, security, inevitability, transparency, complexity, capacity and verification method.

Uses an Advanced Encryption Standard – **Rijndael Algorithm** for encryption of secret file.

2.2.4 Objectives of the Proposed System

The system proposes a real time solution for protecting the piracy of digital content.

More than one digital content can be watermarked in a single process.

The system uses a highly secure algorithm, RIJNDAEL Algorithm.

The system had implemented a symmetric key encryption.

3. PROGRAMMING ENVIRONMENT

3.1 Hardware Requirement

Processor	Intel Pentium IV
CPU	500 MHz
RAM	256 MB
Monitor	17" Samsung color monitor
Hard Disk	40 GB
Keyboard	Standard Keyboard with 104 Keys
Mouse	Serial Mouse

3.2 Software Requirement

Tools	C#.NET Framework
Platform	Windows XP Pentium IV PC with 256 MB RAM Network facility

4. DESCRIPTION OF THE LANGUAGE USED

Microsoft .NET is one of the latest and new technologies introduced by Microsoft Corporation. Nowadays we use to connect to the Internet using a computer and remote computer responses via a web page and a collection of web pages are called as Web Sites. The Concept in .NET is that these websites can integrate with other sites and services using Standard Protocols like HTTP.

Microsoft .NET Platform comprises of four core components such as

1..NET Building Block Services such as file storage, calendar called Passport .NET

2..NET Device Software that will run on latest Internet Devices like Mobile Phones.

3..NET user experience such as integrating this technology to user created documents (integrates with XML). For example if you code XML via a .NET Language like C#, it will automatically create XML document

4..NET infrastructure which includes .NET Framework (Common Language Runtime & .NET Framework Class Libraries) Microsoft Visual Studio.NET such as Visual Basic.NET, Visual C++.NET etc.NET Enterprise Servers and Microsoft Windows.NET

5.Can build robust, scalable, distributed applications with the help of .NET and the part that helps to develop these applications is called the .NET Framework.

6.The .NET Framework contains Common Language Runtime (CLR) and the.NET Framework class libraries also called as Base Class Libraries.

7.All the .NET languages (like C-sharp, VisualBasic.NET, Visual

C++. NET etc) have the .NET Framework class libraries built into them. The .NET class Libraries also supports File I/O, database operations, XML (Extensible Markup Language) and SOAP (Simple Object Access Protocol). For example XML Pages can be developed by using C-sharp language.

When someone talks about .NET development, then it is understood that they are talking about .NET Framework. It includes a Runtime environment and a set of Class Libraries which is being used by a new language called C-sharp abbreviated as C#.

The Basic Requirements needed to begin C-sharp Programming are

1 .NET Framework Software Development Kit

2 An Editor (like Notepad or DOS Editor) to write source codes.

3. Visual C#. NET or Visual C++ 6.0 included with Visual Studio 6.0 (optional)

As a first step .NET SDK should be installed in the system to begin C-sharp Programming. It can be downloaded from the Microsoft's Website. It is also available with almost all the popular computing magazine CD'S. It also comes with the complete documentation in the form of HTML Help.

At the time of this writing only beta version of the kit is available. This kit enables us to compile & execute the source code in C#, Visual Basic by using its built-in Command line Compiler (csc and vbc) and runtime Just In Time (JIT) Compiler. This is similar to Java Compiler (javac) and Java Interpreter (java) included with Java Development Kit.

It is possible to develop applications with C-sharp and Visual Basic by using Visual Studio.NET languages like Visual C#.NET and Visual Basic.NET. This will help us to develop window based applications easily and with limited effort because we don't have to devote too much time in designing the user interface (Usage of WinForms). The only work left for us is to do is to write the codings appropriately as per the .NET Standards.

The choice between C# and VB.NET is largely one of subjective preference. Some people like C#'s terse syntax, others like VB.NET's natural language, case-insensitive approach. Both have access to the same framework libraries. Both will perform largely equivalently (with a few small differences which are unlikely to affect most people, assuming VB.NET is used with Option Strict on). Learning the .NET framework itself is a much bigger issue than learning either of the languages, and it's perfectly possible to become fluent in both. The advantages of C# over VB.NET are listed below.

1. Support for optional parameters - very handy for some COM interoperability

2. Support for late binding with Option Strict off - type safety at compile time goes out of the window, but legacy libraries which don't have strongly typed interfaces become easier to use.

3. Support for named indexers.

4. Various legacy VB functions (provided in the Microsoft.VisualBasic namespace, and can be used by other languages with a reference to the Microsoft.VisualBasic.dll). Many of these can be harmful to performance if used unwisely, however, and many people believe they should be avoided for the most part.

5. The with construct: it's a matter of debate as to whether this is an advantage or not, but it's certainly a difference.

6. Simpler (in expression - perhaps more complicated in understanding) event handling, where a method can declare that it handles an event, rather than the handler having to be set up in code.

7. The ability to implement interfaces with methods of different names. (Arguably this makes it harder to find the implementation of an interface, however.)

8. Catch ... When ... clauses, which allow exceptions to be filtered based on runtime expressions rather than just by type.

9. The VB.NET part of Visual Studio .NET compiles your code in the background. While this is considered an advantage for small projects, people creating very large projects have found that the IDE slows down considerably as the project gets larger.

10. XML documentation generated from source code comments.

11. Operator overloading.

12. Language support for unsigned types.

13. The using statement, which makes unmanaged resource disposal simple.

14. Explicit interface implementation, where an interface which is already implemented in a base class can be reimplementation separately in a derived class. Arguably this makes the class harder to understand, in the same way that member hiding normally does.

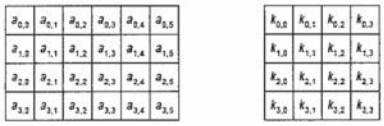
15. Unsafe code. This allows pointer arithmetic etc, and can improve performance in some situations. However, it is not to be used lightly, as a lot of the normal safety of C# is lost (as the name implies). Note that unsafe code is still managed code, i.e. it is compiled to IL, JITted, and run within the CLR.

5. ALGORITHM ANALYSIS

Rijndael is an iterated block cipher with a variable block length and a variable key length. The block length and the key length can be independently specified to 128, 192 or 256 bits.

The algorithm performs four different transformations in a specified order to encrypt and decrypt the secret data into and from the cover data respectively.

These different transformations operate on the intermediate result, called State: the intermediate cipher result. The State can be pictured as a rectangular array of bytes. This array has four rows, the number of columns is denoted by **Nb** and is equal to the block length divided by 32. The Cipher Key is similarly pictured as a rectangular array with four rows. The number of columns of the Cipher Key is denoted by **Nk** and is equal to the key length divided by 32.



Example of State (Nb=6) and Cipher key (Nk=4) Layout

5.1 Round Transformation

The round transformation is composed of four different transformations, as given below:

1.Byte Substitution

2.Shift Row

3.Mix Column

4.Add Round Key

5.1.1 Byte Substitution

The ByteSub Transformation is a non-linear byte substitution, operating on each of the State bytes independently. The substitution table (or S-box) is invertible and is constructed by the composition of two transformations:

1. First, taking the multiplicative inverse in GF(2^8), '00' is mapped onto itself.

2. Then, applying an affine (over GF(2)) transformation defined by:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

The application of the described S-box to all bytes of the State is denoted by:

ByteSub(State)

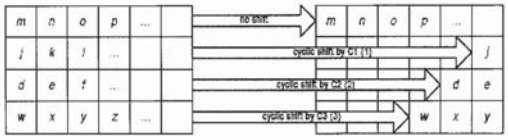


Figure Description: The above figure represents the effect of Shift row transformation on the State.

The inverse of ShiftRow is a cyclic shift of the 3 bottom rows over **Nb**-C1, **Nb**-C2 and **Nb**-C3 bytes respectively so that the byte at position j in row i moves to position (j + **Nb**-Ci) mod **Nb**.

5.1.3 Mix Column Transformation

In MixColumn, the columns of the State are considered as polynomials over GF(28) and multiplied modulo x4 + 1 with a fixed polynomial c(x), given by

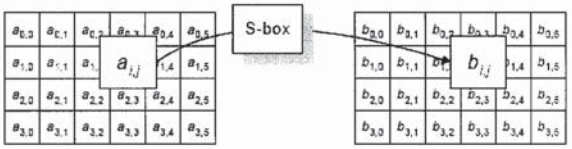
$$c(x) = \text{'03'}\ x3 + \text{'01'}\ x2 + \text{'01'}\ x + \text{'02'}.$$

This polynomial is coprime to x4 + 1 and therefore invertible. This can be written as a matrix multiplication. Let b(x) = c(x) ⊡a(x),

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

The application of this operation on all columns of the State is denoted by

MixColumn(State).



ByteSub acts on the individual bytes of the State.The inverse of ByteSub is the byte substitution where the inverse table is applied. This is obtained by the inverse of the affine mapping followed by taking the multiplicative inverse in GF(2^8).

5.1.2 Shift Row Transformation

In ShiftRow, the rows of the State are cyclically shifted over different offsets. Row 0 is not shifted, Row 1 is shifted over C1 bytes, row 2 over C2 bytes and row 3 over C3 bytes. The shift offsets C1, C2 and C3 depend on the block length **Nb**. The different values are specified in Table below.

Nb	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

The operation of shifting the rows of the State over the specified offsets is denoted by:

ShiftRow(State)

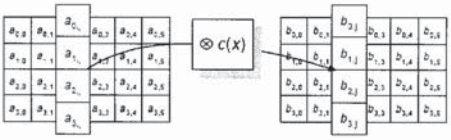


Figure Description: The above figure illustrates the effect of the MixColumn transformation on the State.

The inverse of MixColumn is similar to MixColumn. Every column is transformed by multiplying it with a specific multiplication polynomial d(x), defined by

$$(\text{'03'}\ x^3 + \text{'01'}\ x^2 + \text{'01'}\ x + \text{'02'}) \boxdot d(x) = \text{'01'}$$

It is given by: $d(x) = \text{'0B'}\ x^3 + \text{'0D'}\ x^2 + \text{'09'}\ x + \text{'0E'}$.

5.1.4 The Round Key addition

In this operation, a Round Key is applied to the State by a simple bitwise EXOR. The Round Key is derived from the Cipher Key by means of the key schedule. The Round Key length is equal to the block length **Nb**.

The transformation that consists of EXORing a Round Key to the State is denoted by: AddRoundKey(State,RoundKey).

Figure Description: In the key addition the Round Key is bitwise EXORed to the State.

AddRoundKey is its own inverse.

6. DESIGN AND IMPLEMENTATION

This chapter tells about how the Piracy Protection Software for Mp3 is designed and how the user can make use of it and what all processes are taking place are all presented and also if an input is given by the user the output that is produced also discussed clearly. And also the detail study about the flow of data inside the package is also told clearly. The software used for the package is also given and its advantages are also discussed. The overall step-by-step process of designing and how it is implemented, the hardware and software requirements, and using the package in given in design and implementation.

6.1. Modules Used:

1. Design
2. Encoding and Insertion of Secret file
3. Detection and Decoding
4. Implementation

6.1.1 Design Module:

The designing module includes the data collection and the system design. The work done in the design module includes the architectural skeleton of overall functionalities of the system including the input to be given and the output to be produced including form design.

The System consists of six panels each with its specified task. Entry panel enables the user to opt the language. Then either Embedding process or extraction process can be selected. In case of Embedding, the secret file to be embedded is first to be selected and its size will be displayed. For embedding, a cover file (MP3) is needed and it can be selected correspondingly. Then to ensure security password is given and is confirmed. After completion the embedded MP3 is saved in a specific location and this ends the encryption process.

In order to extract the secret file (Decryption), the file to be recovered is selected and corresponding password is given. If the password matches, the secret file is extracted from the cover file and is saved separately.

6.1.2. Encoding and Insertion of Secret file

In the watermark insertion and encoding module the Secret file to be embedded is encrypted using Rijndael Algorithm an Advanced Encryption Standard. The watermark is then inserted in to the MP3 file.

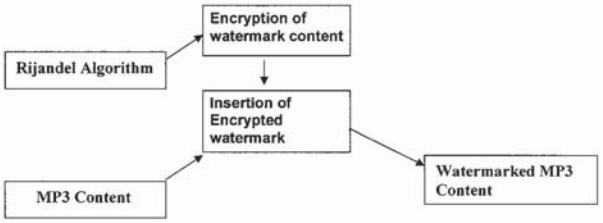


Fig 6.1.2 Insertion of secret data

Figure Description: This figure describes how the watermark is inserted into the MP3 file and how the encryption process is made is given

6.1.3. Detection and Decoding

In the decoding and watermark detection module the watermark is detected in the MP3 data without any damage or loss to both parent and watermark files. After detecting the watermark Decryption is performed using the Rijndael decryption algorithm.

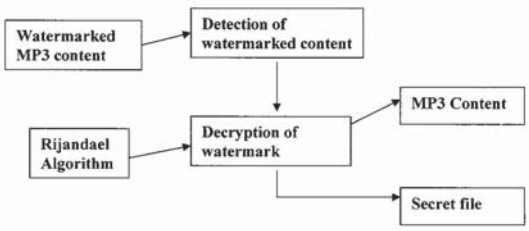


Fig 6.1.3 Extraction of secret data

Figure Description: This figure describes how the decryption process is done and the detection of watermark is done

6.1.4. Implementation

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.OleDb;
using System.Threading;
using System.Text;
using System.IO;
using System.Runtime.InteropServices;
using DataHidder.utils;

namespace DataHidder
{
    /// <summary>
    /// Summary description for Form1.
    /// </summary>
    public class MainFrame : System.Windows.Forms.Form
    {
        // some constants
        string      currentLanguage             = "English";
        const string stdTitle                   = "[ DataHidder 1.0 ]";
        const uint  CF_PROLOG_SIZE_ASSUMPTION   = 100;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.Panel panWelcome;
        private System.Windows.Forms.LinkLabel linkLabel1;
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.Container components = null;
        private Panel[][] tourList;
        private int       tourIndex = 0;
```

```
        private int      bExtract;
        private System.Windows.Forms.Button butBack;
        private System.Windows.Forms.Button butCancel;
        private System.Windows.Forms.Button butNext;
        private System.Windows.Forms.ComboBox cbLangs;
        private System.Windows.Forms.Label labChooseLang;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Panel panWhatToDo;
        private System.Windows.Forms.RadioButton radEmbed;
        private System.Windows.Forms.RadioButton radExtract;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Panel panChooseFiles;
        private System.Windows.Forms.Label labChooseFiles;
        private System.Windows.Forms.Label labTotSecrets;
        private System.Windows.Forms.Label labTotSecretFSize;
        private System.Windows.Forms.ListView lvSecretFiles;
        private System.Windows.Forms.Button butAddSecret;
        private System.Windows.Forms.Button butDelSecret;
        private System.Windows.Forms.OpenFileDialog ofn;
        private utils.TextEnum Strings   = null;
        private long cbSecretFiles      = 0;
        private long cbHidable          = 0;
        private System.Windows.Forms.Panel panChooseMP3Emb;
        private System.Windows.Forms.Button butDelMP3Emb;
        private System.Windows.Forms.Button butAddMP3Emb;
        private System.Windows.Forms.ListView lvMP3sEmb;
        private System.Windows.Forms.Label labChooseMP3Emb;
        private System.Windows.Forms.Label labStep;
        private System.Windows.Forms.Label labData2Hide;
        private System.Windows.Forms.Label labSize2Hide;
        private System.Windows.Forms.Panel panPassword;
        private System.Windows.Forms.Label labEnterPW;
        private System.Windows.Forms.Label labPW1;
        private System.Windows.Forms.TextBox txtPW1;
        private System.Windows.Forms.TextBox txtPW2;
        private System.Windows.Forms.Label labPW2;
        private System.Windows.Forms.Panel panEmbEnd;
        private System.Windows.Forms.Label llabMoreMp3s;
        private System.Windows.Forms.Label labEmbEnd;
        private System.Windows.Forms.ProgressBar pbEmb;
        private System.Windows.Forms.Button butSaveSecMp3s;
        private uint preInjDataSize      = 0;
        private System.Windows.Forms.SaveFileDialog sfd;
        private System.Windows.Forms.Panel panChooseMP3Ex;
        private System.Windows.Forms.Label labChooseMP3ExInfo;
        private System.Windows.Forms.Button butDelMP3Ex;
        private System.Windows.Forms.Button butAddMP3Ex;
        private System.Windows.Forms.ListView lvMP3Ex;
        private utils.MP3Trojan injector  = null;
        private utils.MP3Trojan extractor = null;
        private System.Windows.Forms.ProgressBar pbExEnd;
        private System.Windows.Forms.Label labExtEnd;
        private System.Windows.Forms.Panel panExtEnd;
        private System.Windows.Forms.Button butSaveHiddens;
        private string[] strMp3sToEx = null;

#region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.groupBox1 = new
System.Windows.Forms.GroupBox();
            this.groupBox2 = new
System.Windows.Forms.GroupBox();
            this.butBack = new System.Windows.Forms.Button();
            this.butCancel = new System.Windows.Forms.Button();
            this.butNext = new System.Windows.Forms.Button();
            this.panWelcome = new
System.Windows.Forms.Panel();
            this.label2 = new System.Windows.Forms.Label();
            this.labChooseLang = new
System.Windows.Forms.Label();
            this.cbLangs = new
System.Windows.Forms.ComboBox();
            this.linkLabel1 = new
System.Windows.Forms.LinkLabel();
            this.label1 = new System.Windows.Forms.Label();
            this.groupBox3 = new
System.Windows.Forms.GroupBox();
            this.panExtEnd = new System.Windows.Forms.Panel();
            this.butSaveHiddens = new
System.Windows.Forms.Button();
            this.pbExEnd = new
System.Windows.Forms.ProgressBar();
            this.labExtEnd = new System.Windows.Forms.Label();
            this.panChooseMP3Ex = new
System.Windows.Forms.Panel();
            this.butDelMP3Ex = new
System.Windows.Forms.Button();
            this.butAddMP3Ex = new
System.Windows.Forms.Button();
            this.lvMP3Ex = new
System.Windows.Forms.ListView();
            this.labChooseMP3ExInfo = new
System.Windows.Forms.Label();
            this.panChooseMP3Emb = new
System.Windows.Forms.Panel();
            this.llabMoreMp3s = new
System.Windows.Forms.Label();
            this.labSize2Hide = new
System.Windows.Forms.Label();
            this.labData2Hide = new
System.Windows.Forms.Label();
            this.butDelMP3Emb = new
System.Windows.Forms.Button();
            this.butAddMP3Emb = new
System.Windows.Forms.Button();
```

```
            this.lvMP3sEmb = new
System.Windows.Forms.ListView();
            this.labChooseMP3Emb = new
System.Windows.Forms.Label();
            this.panEmbEnd = new System.Windows.Forms.Panel();
            this.butSaveSecMp3s = new
System.Windows.Forms.Button();
            this.pbEmb = new
System.Windows.Forms.ProgressBar();
            this.labEmbEnd = new System.Windows.Forms.Label();
            this.panPassword = new
System.Windows.Forms.Panel();
            this.txtPW2 = new System.Windows.Forms.TextBox();
            this.labPW2 = new System.Windows.Forms.Label();
            this.txtPW1 = new System.Windows.Forms.TextBox();
            this.labPW1 = new System.Windows.Forms.Label();
            this.labEnterPW = new System.Windows.Forms.Label();
            this.panChooseFiles = new
System.Windows.Forms.Panel();
            this.butDelSecret = new
System.Windows.Forms.Button();
            this.butAddSecret = new
System.Windows.Forms.Button();
            this.labTotSecretFSize = new
System.Windows.Forms.Label();
            this.labTotSecrets = new
System.Windows.Forms.Label();
```

```
            this.lvSecretFiles = new
System.Windows.Forms.ListView();
            this.labChooseFiles = new
System.Windows.Forms.Label();
            this.panWhatToDo = new
System.Windows.Forms.Panel();
            this.radExtract = new
System.Windows.Forms.RadioButton();
            this.radEmbed = new
System.Windows.Forms.RadioButton();
            this.ofn = new
System.Windows.Forms.OpenFileDialog();
            this.labStep = new System.Windows.Forms.Label();
            this.sfd = new
System.Windows.Forms.SaveFileDialog();
            this.panWelcome.SuspendLayout();
            this.groupBox3.SuspendLayout();
            this.panExtEnd.SuspendLayout();
            this.panChooseMP3Ex.SuspendLayout();
            this.panChooseMP3Emb.SuspendLayout();
            this.panEmbEnd.SuspendLayout();
            this.panPassword.SuspendLayout();
            this.panChooseFiles.SuspendLayout();
            this.panWhatToDo.SuspendLayout();
            this.SuspendLayout();
            // groupBox1
            this.groupBox1.Location =
            new System.Drawing.Point(8, 8);
```

```
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size =
            new System.Drawing.Size(152, 267);
            this.groupBox1.TabIndex = 1;
            this.groupBox1.TabStop = false;
            // groupBox2
            this.groupBox2.Location =
            new System.Drawing.Point(16, 280);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size =
            new System.Drawing.Size(480, 8);
            this.groupBox2.TabIndex = 2;
            this.groupBox2.TabStop = false;
            // butBack
            this.butBack.Enabled = false;
            this.butBack.Location =
            new System.Drawing.Point(184, 296);
            this.butBack.Name = "butBack";
            this.butBack.Size = new System.Drawing.Size(88, 23);
            this.butBack.TabIndex = 3;
            this.butBack.Text = "(b)";
            this.butBack.Click += new
System.EventHandler(this.butBack_Click);
            // butCancel
            this.butCancel.Location =
            new System.Drawing.Point(392, 296);
            this.butCancel.Name = "butCancel";
            this.butCancel.Size = new System.Drawing.Size(88, 23);
```

```
            this.butCancel.TabIndex = 4;
            this.butCancel.Text = "(c)";
            this.butCancel.Click +=
             new System.EventHandler(this.button2_Click);
            // butNext
            this.butNext.Location =
             new System.Drawing.Point(288, 296);
            this.butNext.Name = "butNext";
            this.butNext.Size = new System.Drawing.Size(88, 23);
            this.butNext.TabIndex = 5;
            this.butNext.Text = "(n)";
            this.butNext.Click +=
             new System.EventHandler(this.butNext_Click);
            // panWelcome
            this.panWelcome.Controls.Add(this.label2);
            this.panWelcome.Controls.Add(this.labChooseLang);
            this.panWelcome.Controls.Add(this.cbLangs);
            this.panWelcome.Controls.Add(this.linkLabel1);
            this.panWelcome.Location =
            new System.Drawing.Point(8, 8);
            this.panWelcome.Name = "panWelcome";
            this.panWelcome.Size =
            new System.Drawing.Size(320, 256);
            this.panWelcome.TabIndex = 6;
            // label2
            this.label2.Font =
```

```
                new System.Drawing.Font("Times New Roman", 8.25F,
            System.Drawing.FontStyle.Bold,
            System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.label2.Location =
                new System.Drawing.Point(120, 200);
                this.label2.Name = "label2";
                this.label2.Size = new System.Drawing.Size(104, 16);
                this.label2.TabIndex = 4;
                // labChooseLang
                this.labChooseLang.Location =
                new System.Drawing.Point(24, 144);
                this.labChooseLang.Name = "labChooseLang";
                this.labChooseLang.Size =
                new System.Drawing.Size(128, 16);
                this.labChooseLang.TabIndex = 1;
                this.labChooseLang.Text = "(choose ur lang)";
                // cbLangs
                this.cbLangs.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
                this.cbLangs.Location =
                new System.Drawing.Point(176, 140);
                this.cbLangs.Name = "cbLangs";
                this.cbLangs.Size = new System.Drawing.Size(128, 21);
                this.cbLangs.Sorted = true;
                this.cbLangs.TabIndex = 2;
                this.cbLangs.SelectedIndexChanged += new
System.EventHandler(this.cbLangs_SelectedIndexChanged);
                // linkLabel1
                this.linkLabel1.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
                this.linkLabel1.Font = new System.Drawing.Font("Times
New Roman", 13F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
                this.linkLabel1.LinkBehavior =
System.Windows.Forms.LinkBehavior.HoverUnderline;
                this.linkLabel1.Location = new
System.Drawing.Point(64, 218);
                this.linkLabel1.Name = "linkLabel1";
                this.linkLabel1.Size =
                 new System.Drawing.Size(205, 16);
                this.linkLabel1.TabIndex = 0;
                this.linkLabel1.LinkClicked += new
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabel
1_LinkClicked);
                //
                // label1
                //
                this.label1.Location = new System.Drawing.Point(110,
117);
                this.label1.Name = "label1";
                this.label1.TabIndex = 2;
                this.label1.Text = "label1";
                //
                // groupBox3
                //
                this.groupBox3.Controls.Add(this.panWelcome);
                this.groupBox3.Controls.Add(this.panExtEnd);
                this.groupBox3.Controls.Add(this.panChooseMP3Ex);
                this.groupBox3.Controls.Add(this.panChooseMP3Emb);
                this.groupBox3.Controls.Add(this.panEmbEnd);
                this.groupBox3.Controls.Add(this.panPassword);
                this.groupBox3.Controls.Add(this.panChooseFiles);
                this.groupBox3.Controls.Add(this.panWhatToDo);
                this.groupBox3.Location = new
System.Drawing.Point(168, 8);
                this.groupBox3.Name = "groupBox3";
                this.groupBox3.Size = new System.Drawing.Size(336,
268);
                this.groupBox3.TabIndex = 3;
                this.groupBox3.TabStop = false;
                // panExtEnd

        this.panExtEnd.Controls.Add(this.butSaveHiddens);
                this.panExtEnd.Controls.Add(this.pbExEnd);
                this.panExtEnd.Controls.Add(this.labExtEnd);
                this.panExtEnd.Location = new System.Drawing.Point(8,
8);
                this.panExtEnd.Name = "panExtEnd";
                this.panExtEnd.Size = new System.Drawing.Size(320,
256);
                this.panExtEnd.TabIndex = 3;
                this.panExtEnd.Visible = false;
                //
                // butSaveHiddens
                //
                this.butSaveHiddens.Location = new
System.Drawing.Point(96, 168);
                this.butSaveHiddens.Name = "butSaveHiddens";
                this.butSaveHiddens.Size = new
System.Drawing.Size(136, 24);
                this.butSaveHiddens.TabIndex = 5;
                this.butSaveHiddens.Text = "(save)";
                this.butSaveHiddens.Click += new
System.EventHandler(this.butSaveHiddens_Click);
                //
                // pbExEnd
                //
                this.pbExEnd.Location = new System.Drawing.Point(32,
128);
                this.pbExEnd.Name = "pbExEnd";
                this.pbExEnd.Size = new System.Drawing.Size(256, 14);
                this.pbExEnd.TabIndex = 4;
                //
                // labExtEnd
                //
                this.labExtEnd.Location = new
System.Drawing.Point(24, 16);
                this.labExtEnd.Name = "labExtEnd";
                this.labExtEnd.Size = new System.Drawing.Size(280,
88);
                this.labExtEnd.TabIndex = 3;
                this.labExtEnd.Text = "(lalala)";
```

```
            //
            // panChooseMP3Ex
            //

        this.panChooseMP3Ex.Controls.Add(this.butDelMP3Ex);

        this.panChooseMP3Ex.Controls.Add(this.butAddMP3Ex);
            this.panChooseMP3Ex.Controls.Add(this.lvMP3Ex);

        this.panChooseMP3Ex.Controls.Add(this.labChooseMP3ExInfo);
            this.panChooseMP3Ex.Location = new
System.Drawing.Point(8, 8);
            this.panChooseMP3Ex.Name = "panChooseMP3Ex";
            this.panChooseMP3Ex.Size = new
System.Drawing.Size(320, 256);
            this.panChooseMP3Ex.TabIndex = 15;
            this.panChooseMP3Ex.Visible = false;
            //
            // butDelMP3Ex
            //
            this.butDelMP3Ex.Enabled = false;
            this.butDelMP3Ex.Location = new
System.Drawing.Point(216, 216);
            this.butDelMP3Ex.Name = "butDelMP3Ex";
            this.butDelMP3Ex.Size = new System.Drawing.Size(88,
23);
            this.butDelMP3Ex.TabIndex = 12;
            this.butDelMP3Ex.Text = "(del)";
            this.butDelMP3Ex.Click += new
System.EventHandler(this.butDelMP3Ex_Click);
            //
            // butAddMP3Ex
            //
            this.butAddMP3Ex.Location = new
System.Drawing.Point(16, 216);
            this.butAddMP3Ex.Name = "butAddMP3Ex";
            this.butAddMP3Ex.Size = new System.Drawing.Size(88,
23);
            this.butAddMP3Ex.TabIndex = 11;
            this.butAddMP3Ex.Text = "(add)";
            this.butAddMP3Ex.Click += new
System.EventHandler(this.butAddMP3Ex_Click);
            //
            // lvMP3Ex
            //
            this.lvMP3Ex.FullRowSelect = true;
            this.lvMP3Ex.Location = new System.Drawing.Point(16,
88);
            this.lvMP3Ex.MultiSelect = false;
            this.lvMP3Ex.Name = "lvMP3Ex";
            this.lvMP3Ex.Size = new System.Drawing.Size(288,
120);
            this.lvMP3Ex.TabIndex = 10;
            this.lvMP3Ex.View =
System.Windows.Forms.View.Details;
            this.lvMP3Ex.SelectedIndexChanged += new
System.EventHandler(this.lvMP3Ex_SelectedIndexChanged);
            //
            // labChooseMP3ExInfo
            //
            this.labChooseMP3ExInfo.Location = new
System.Drawing.Point(24, 16);
            this.labChooseMP3ExInfo.Name =
"labChooseMP3ExInfo";
            this.labChooseMP3ExInfo.Size = new
System.Drawing.Size(280, 56);
            this.labChooseMP3ExInfo.TabIndex = 0;
            this.labChooseMP3ExInfo.Text = "(lalala)";
            //
            // panChooseMP3Emb
            //

        this.panChooseMP3Emb.Controls.Add(this.llabMoreMp3s);

        this.panChooseMP3Emb.Controls.Add(this.labSize2Hide);

        this.panChooseMP3Emb.Controls.Add(this.labData2Hide);

        this.panChooseMP3Emb.Controls.Add(this.butDelMP3Emb);

        this.panChooseMP3Emb.Controls.Add(this.butAddMP3Emb);

        this.panChooseMP3Emb.Controls.Add(this.lvMP3sEmb);

        this.panChooseMP3Emb.Controls.Add(this.labChooseMP3Emb);
            this.panChooseMP3Emb.Location = new
System.Drawing.Point(8, 8);
            this.panChooseMP3Emb.Name = "panChooseMP3Emb";
            this.panChooseMP3Emb.Size = new
System.Drawing.Size(320, 256);
            this.panChooseMP3Emb.TabIndex = 6;
            this.panChooseMP3Emb.Visible = false;
            //
            // llabMoreMp3s
            //
            this.llabMoreMp3s.Location = new
System.Drawing.Point(8, 232);
            this.llabMoreMp3s.Name = "llabMoreMp3s";
            this.llabMoreMp3s.Size = new
System.Drawing.Size(304, 18);
            this.llabMoreMp3s.TabIndex = 13;
            this.llabMoreMp3s.Text = "(more Mp3s)";
            this.llabMoreMp3s.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            //
            // labSize2Hide
            //
            this.labSize2Hide.Location = new
System.Drawing.Point(168, 216);
            this.labSize2Hide.Name = "labSize2Hide";
```

```
            this.labSize2Hide.Size = new System.Drawing.Size(136,
16);
            this.labSize2Hide.TabIndex = 12;
            this.labSize2Hide.Text = "(size)";
            this.labSize2Hide.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // labData2Hide
            //
            this.labData2Hide.Location = new
System.Drawing.Point(16, 216);
            this.labData2Hide.Name = "labData2Hide";
            this.labData2Hide.Size = new System.Drawing.Size(152,
16);
            this.labData2Hide.TabIndex = 11;
            this.labData2Hide.Text = "(data to hide)";
            //
            // butDelMP3Emb
            //
            this.butDelMP3Emb.Enabled = false;
            this.butDelMP3Emb.Location = new
System.Drawing.Point(216, 184);
            this.butDelMP3Emb.Name = "butDelMP3Emb";
            this.butDelMP3Emb.Size = new
System.Drawing.Size(88, 23);
            this.butDelMP3Emb.TabIndex = 9;
            this.butDelMP3Emb.Text = "(del)";
            this.butDelMP3Emb.Click += new
System.EventHandler(this.butDelMP3Emb_Click);
            //
            // butAddMP3Emb
            //
            this.butAddMP3Emb.Location = new
System.Drawing.Point(16, 184);
            this.butAddMP3Emb.Name = "butAddMP3Emb";
            this.butAddMP3Emb.Size = new
System.Drawing.Size(88, 23);
            this.butAddMP3Emb.TabIndex = 8;
            this.butAddMP3Emb.Text = "(add)";
            this.butAddMP3Emb.Click += new
System.EventHandler(this.butAddMP3Emb_Click);
            //
            // lvMP3sEmb
            //
            this.lvMP3sEmb.FullRowSelect = true;
            this.lvMP3sEmb.Location = new
System.Drawing.Point(16, 80);
            this.lvMP3sEmb.MultiSelect = false;
            this.lvMP3sEmb.Name = "lvMP3sEmb";
            this.lvMP3sEmb.Size = new System.Drawing.Size(288,
97);
            this.lvMP3sEmb.TabIndex = 7;
            this.lvMP3sEmb.View =
System.Windows.Forms.View.Details;
            this.lvMP3sEmb.SelectedIndexChanged += new
System.EventHandler(this.lvMP3sEmb_SelectedIndexChanged);
            //
            // labChooseMP3Emb
            //
            this.labChooseMP3Emb.Location = new
System.Drawing.Point(24, 16);
            this.labChooseMP3Emb.Name = "labChooseMP3Emb";
            this.labChooseMP3Emb.Size = new
System.Drawing.Size(272, 56);
            this.labChooseMP3Emb.TabIndex = 6;
            this.labChooseMP3Emb.Text = "(choose MP3s to add
secret data in)";
            //
            // panEmbEnd
            //
            this.panEmbEnd.Controls.Add(this.butSaveSecMp3s);
            this.panEmbEnd.Controls.Add(this.pbEmb);
            this.panEmbEnd.Controls.Add(this.labEmbEnd);
            this.panEmbEnd.Location = new
System.Drawing.Point(8, 8);
            this.panEmbEnd.Name = "panEmbEnd";
            this.panEmbEnd.Size = new System.Drawing.Size(320,
256);
            this.panEmbEnd.TabIndex = 14;
            this.panEmbEnd.Visible = false;
            //
            // butSaveSecMp3s
            //
            this.butSaveSecMp3s.Location = new
System.Drawing.Point(80, 184);
            this.butSaveSecMp3s.Name = "butSaveSecMp3s";
            this.butSaveSecMp3s.Size = new
System.Drawing.Size(136, 24);
            this.butSaveSecMp3s.TabIndex = 2;
            this.butSaveSecMp3s.Text = "(save)";
            this.butSaveSecMp3s.Click += new
System.EventHandler(this.butSaveSecMp3s_Click);
            //
            // pbEmb
            //
            this.pbEmb.Location = new System.Drawing.Point(32,
136);
            this.pbEmb.Name = "pbEmb";
            this.pbEmb.Size = new System.Drawing.Size(256, 14);
            this.pbEmb.TabIndex = 1;
            //
            // labEmbEnd
            //
            this.labEmbEnd.Location = new
System.Drawing.Point(24, 16);
            this.labEmbEnd.Name = "labEmbEnd";
            this.labEmbEnd.Size = new System.Drawing.Size(280,
88);
            this.labEmbEnd.TabIndex = 0;
            this.labEmbEnd.Text = "(lalala)";
```

```
            //
            // panPassword
            //
            this.panPassword.Controls.Add(this.txtPW2);
            this.panPassword.Controls.Add(this.labPW2);
            this.panPassword.Controls.Add(this.txtPW1);
            this.panPassword.Controls.Add(this.labPW1);
            this.panPassword.Controls.Add(this.labEnterPW);
            this.panPassword.Location = new
System.Drawing.Point(8, 8);
            this.panPassword.Name = "panPassword";
            this.panPassword.Size = new System.Drawing.Size(320,
256);
            this.panPassword.TabIndex = 13;
            this.panPassword.Visible = false;
            //
            // txtPW2
            //
            this.txtPW2.Location = new System.Drawing.Point(176,
160);
            this.txtPW2.Name = "txtPW2";
            this.txtPW2.PasswordChar = '?';
            this.txtPW2.Size = new System.Drawing.Size(120, 20);
            this.txtPW2.TabIndex = 5;
            this.txtPW2.Text = "textBox1";
            //
            // labPW2
            //
            this.labPW2.Location = new System.Drawing.Point(24,
161);
            this.labPW2.Name = "labPW2";
            this.labPW2.Size = new System.Drawing.Size(144, 16);
            this.labPW2.TabIndex = 4;
            this.labPW2.Text = "(pw)";
            //
            // txtPW1
            //
            this.txtPW1.Location = new System.Drawing.Point(176,
132);
            this.txtPW1.Name = "txtPW1";
            this.txtPW1.PasswordChar = '?';
            this.txtPW1.Size = new System.Drawing.Size(120, 20);
            this.txtPW1.TabIndex = 3;
            this.txtPW1.Text = "textBox1";
            //
            // labPW1
            //
            this.labPW1.Location = new System.Drawing.Point(24,
137);
            this.labPW1.Name = "labPW1";
            this.labPW1.Size = new System.Drawing.Size(144, 16);
            this.labPW1.TabIndex = 2;
            this.labPW1.Text = "(pw)";
            //
            // labEnterPW
            //
            this.labEnterPW.Location = new
System.Drawing.Point(24, 16);
            this.labEnterPW.Name = "labEnterPW";
            this.labEnterPW.Size = new System.Drawing.Size(272,
72);
            this.labEnterPW.TabIndex = 1;
            this.labEnterPW.Text = "(enter PW)";
            //
            // panChooseFiles
            //
            this.panChooseFiles.Controls.Add(this.butDelSecret);
            this.panChooseFiles.Controls.Add(this.butAddSecret);

        this.panChooseFiles.Controls.Add(this.labTotSecretFSize);
            this.panChooseFiles.Controls.Add(this.labTotSecrets);
            this.panChooseFiles.Controls.Add(this.lvSecretFiles);
            this.panChooseFiles.Controls.Add(this.labChooseFiles);
            this.panChooseFiles.Location = new
System.Drawing.Point(8, 8);
            this.panChooseFiles.Name = "panChooseFiles";
            this.panChooseFiles.Size = new
System.Drawing.Size(320, 256);
            this.panChooseFiles.TabIndex = 6;
            this.panChooseFiles.Visible = false;
            //
            // butDelSecret
            //
            this.butDelSecret.Enabled = false;
            this.butDelSecret.Location = new
System.Drawing.Point(208, 176);
            this.butDelSecret.Name = "butDelSecret";
            this.butDelSecret.Size = new System.Drawing.Size(88,
23);
            this.butDelSecret.TabIndex = 5;
            this.butDelSecret.Text = "(del)";
            this.butDelSecret.Click += new
System.EventHandler(this.butDelSecret_Click);
            //
            // butAddSecret
            //
            this.butAddSecret.Location = new
System.Drawing.Point(16, 176);
            this.butAddSecret.Name = "butAddSecret";
            this.butAddSecret.Size = new System.Drawing.Size(88,
23);
            this.butAddSecret.TabIndex = 4;
            this.butAddSecret.Text = "(add)";
            this.butAddSecret.Click += new
System.EventHandler(this.butAddSecret_Click);
            //
            // labTotSecretFSize
            //
            this.labTotSecretFSize.Location = new
System.Drawing.Point(160, 224);
            this.labTotSecretFSize.Name = "labTotSecretFSize";
```

```
            this.labTotSecretFSize.Size = new
System.Drawing.Size(144, 16);
            this.labTotSecretFSize.TabIndex = 3;
            this.labTotSecretFSize.Text = "label3";
            this.labTotSecretFSize.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
            //
            // labTotSecrets
            //
            this.labTotSecrets.Location = new
System.Drawing.Point(16, 224);
            this.labTotSecrets.Name = "labTotSecrets";
            this.labTotSecrets.Size = new System.Drawing.Size(136,
16);
            this.labTotSecrets.TabIndex = 2;
            this.labTotSecrets.Text = "(total byte)";
            //
            // lvSecretFiles
            //
            this.lvSecretFiles.FullRowSelect = true;
            this.lvSecretFiles.Location = new
System.Drawing.Point(16, 72);
            this.lvSecretFiles.MultiSelect = false;
            this.lvSecretFiles.Name = "lvSecretFiles";
            this.lvSecretFiles.Size = new System.Drawing.Size(288,
80);
            this.lvSecretFiles.TabIndex = 1;
            this.lvSecretFiles.View =
System.Windows.Forms.View.Details;
            this.lvSecretFiles.SelectedIndexChanged += new
System.EventHandler(this.lvSecretFiles_SelectedIndexChanged);
            //
            // labChooseFiles
            //
            this.labChooseFiles.Location = new
System.Drawing.Point(24, 16);
            this.labChooseFiles.Name = "labChooseFiles";
            this.labChooseFiles.Size = new
System.Drawing.Size(272, 48);
            this.labChooseFiles.TabIndex = 0;
            this.labChooseFiles.Text = "(choose files to embed)";
            //
            // panWhatToDo
            //
            this.panWhatToDo.Controls.Add(this.radExtract);
            this.panWhatToDo.Controls.Add(this.radEmbed);
            this.panWhatToDo.Location = new
System.Drawing.Point(8, 8);
            this.panWhatToDo.Name = "panWhatToDo";
            this.panWhatToDo.Size = new
System.Drawing.Size(320, 256);
            this.panWhatToDo.TabIndex = 7;
            this.panWhatToDo.Visible = false;
            //
            // radExtract
            //
            this.radExtract.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((System.Byte)(0)));
            this.radExtract.Location = new
System.Drawing.Point(40, 144);
            this.radExtract.Name = "radExtract";
            this.radExtract.Size = new System.Drawing.Size(168,
32);
            this.radExtract.TabIndex = 1;
            this.radExtract.Text = "(extr data)";
            this.radExtract.CheckedChanged += new
System.EventHandler(this.radExtract_CheckedChanged);
            //
            // radEmbed
            //
            this.radEmbed.Font = new
System.Drawing.Font("Microsoft Sans Serif", 9.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((System.Byte)(0)));
            this.radEmbed.Location = new
System.Drawing.Point(40, 40);
            this.radEmbed.Name = "radEmbed";
            this.radEmbed.Size = new System.Drawing.Size(168,
40);
            this.radEmbed.TabIndex = 0;
            this.radEmbed.Text = "(embed data)";
            this.radEmbed.CheckedChanged += new
System.EventHandler(this.radEmbed_CheckedChanged);
            //
            // labStep
            //
            this.labStep.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((System.Byte)(0)));
            this.labStep.Location = new System.Drawing.Point(24,
298);
            this.labStep.Name = "labStep";
            this.labStep.Size = new System.Drawing.Size(114, 16);
            this.labStep.TabIndex = 6;
            //
            // MainFrame
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5,
13);
            this.BackColor = System.Drawing.Color.White;
            this.ClientSize = new System.Drawing.Size(522, 328);
            this.Controls.Add(this.labStep);
            this.Controls.Add(this.butNext);
            this.Controls.Add(this.butCancel);
            this.Controls.Add(this.butBack);
            this.Controls.Add(this.groupBox2);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.groupBox3);
```

```
            this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog;
            this.MaximizeBox = false;
            this.Name = "MainFrame";
            this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "(title)";
            this.Closing += new
System.ComponentModel.CancelEventHandler(this.MainFrame_Closing);
            this.Closed += new
System.EventHandler(this.MainFrame_Closed);
            this.panWelcome.ResumeLayout(false);
            this.groupBox3.ResumeLayout(false);
            this.panExtEnd.ResumeLayout(false);
            this.panChooseMP3Ex.ResumeLayout(false);
            this.panChooseMP3Emb.ResumeLayout(false);
            this.panEmbEnd.ResumeLayout(false);
            this.panPassword.ResumeLayout(false);
            this.panChooseFiles.ResumeLayout(false);
            this.panWhatToDo.ResumeLayout(false);
            this.ResumeLayout(false);

        }
        #endregion
```

6.2 OVERALL FUNCTIONALITIES OF THE SYSTEM

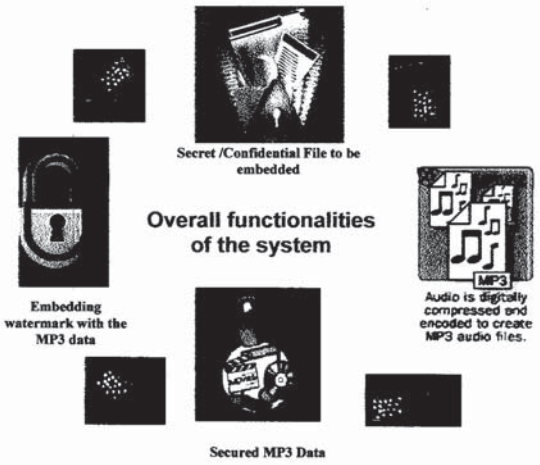


Fig 6.2 Overall functionalities of the system

Figure Description: This figure describes the over all functionality of how the piracy protection software for MP3 is working.

7. CONCLUSION AND FUTHER SCOPE

7.1 Broad Conclusion

Practical software for piracy protection of watermark in transaction of digital contents is developed. In this software, the form of the particular watermark embedded in the digital contents in a transaction is not determined by the content provider and the customer. With this security analysis, it is shown that the software is secure against any possible attacks from either the content provider or the customer. Besides the security feature, the software developed is economic with respect to storage requirements.

7.2 Application Area

The piracy protection software that has been developed can be used by the music industry in releasing their musical contents to the market in order to reinforce access control and protection to the multimedia contents. It is very useful for the music companies who sell their musical contents over the internet, they can easily determine the unethical party in case of the unauthorized distribution of digital contents.

7.3 Suggestion for Further Scope

The further enhancements to this piracy protection software are used to achieve further more benefits in the watermarking field. They are found and listed.

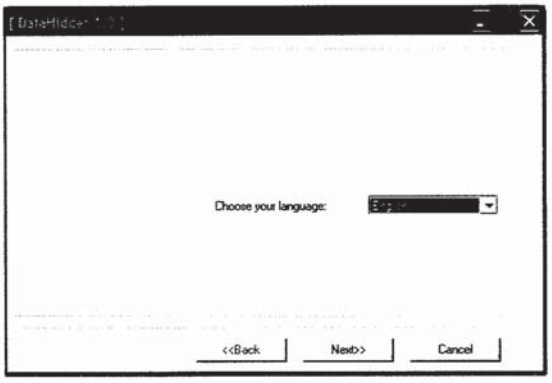
- To perform video watermarking.
- Present the system in other interactive languages.
- Develop watermarking scheme by using several approaches and implement hybrid watermarking.
- Increase the robustness of watermark.

8. APPENDIX

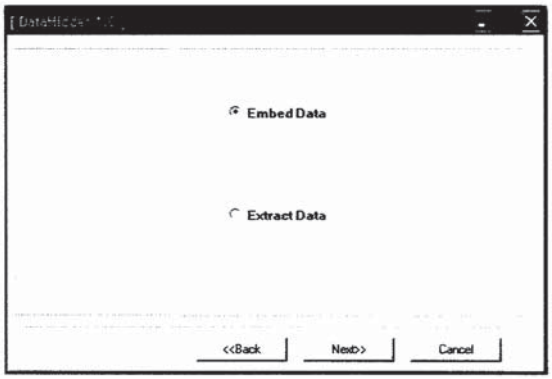
A: SCREEN SHOTS

EMBEDDING PROCESS

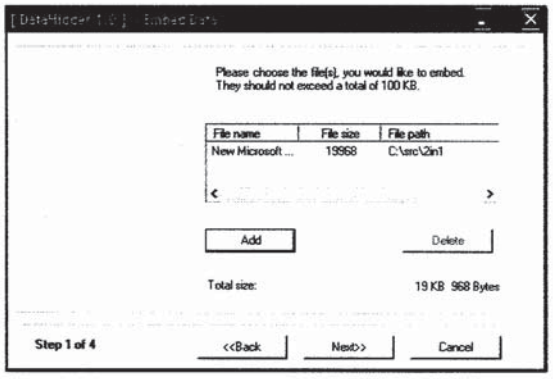
SELECTION OF LANGUAGE



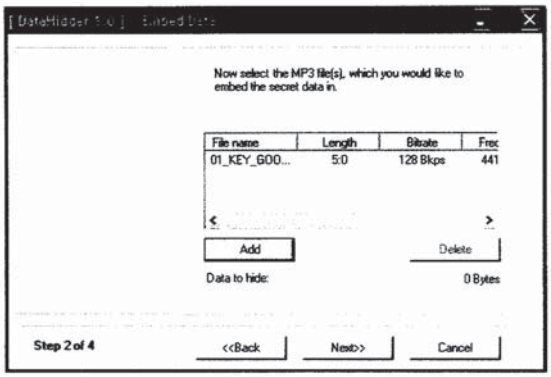
SELECTING EMBED OPTION



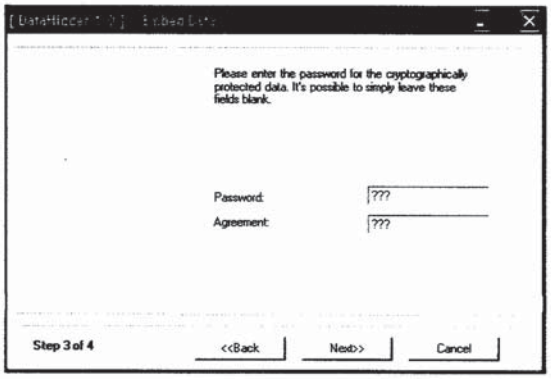
SELECTION OF SECRET FILE TO BE EMBEDDED



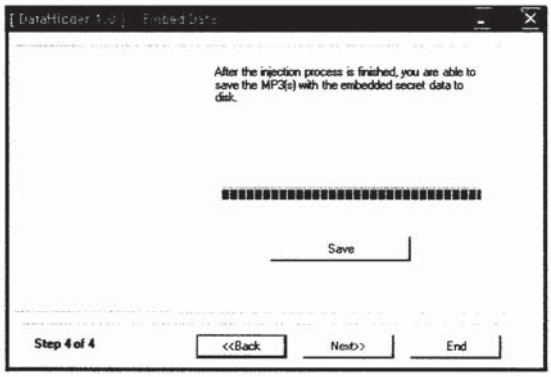
SELECTION OF MP3 FILE(S) TO EMBED THE SECRET FILE



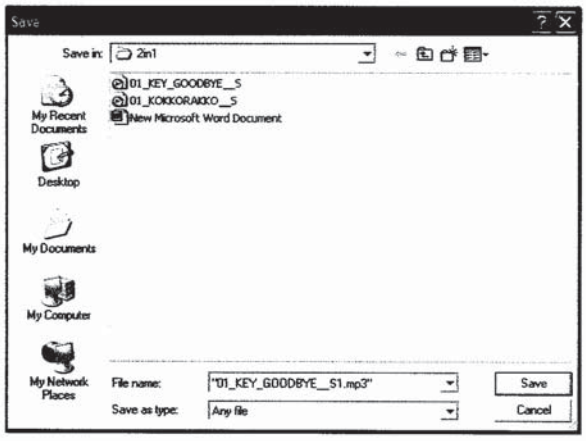
ENTERING AND CONFIRMATION OF PASSWORD



SAVING THE INJECTION PROCESS

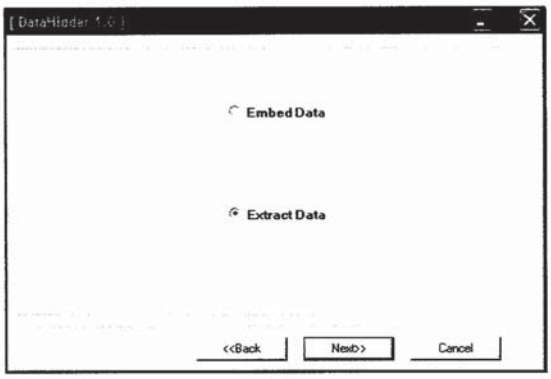


SELECTION OF LOCATION TO SAVE THE PROJECT

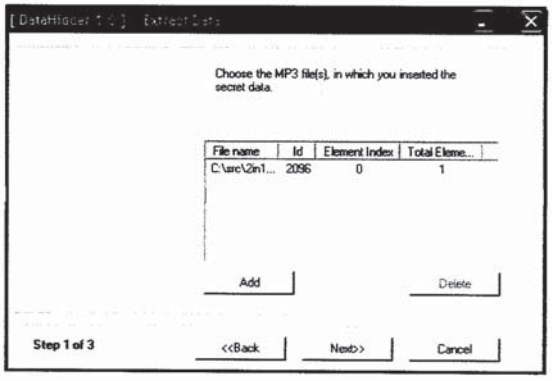


EXTRACTION

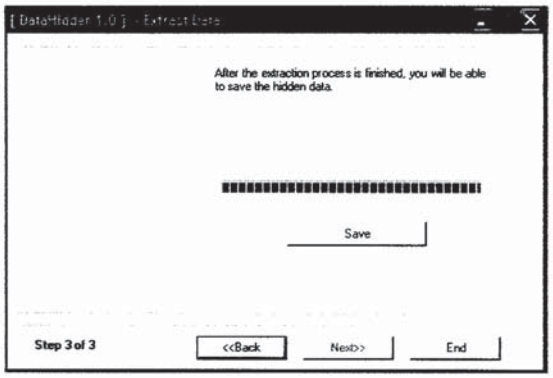
SELECTING EXTRACT OPTION



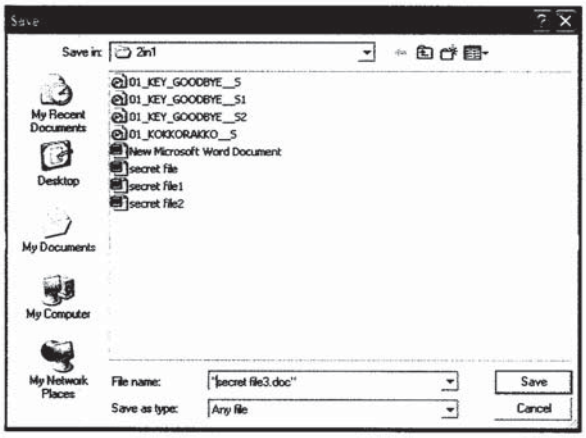
SELECTION OF MP3 FILE(S) IN WHICH SECRET FILE IS INJECTED



SAVING THE EXTRACTED FILE(S)



SAVING THE EXTRACTED FILE(S)



REFERENCES

1. Jun, J.M., Lee, B.M., Kim, K.K. and Won, D.H., "Digital watermarking and practical distribution protocol for digital contents copyright protection," in proceedings of the WISA'2000, page no. 251-264.

2. Gopalakrishanan, K., Memon, N.D. and Vora, P. "Protocols for watermark verification," IEEE Multimedia, 2001, page no. 66-70.

3. Memon, N. and Wong, P. W., "A Buyer-Seller Watermarking Protocol," IEEE Transactions on Image Processing, 2001, page no. 643-649.

4. Shing-chi Cheung and Ho-fung Leung, "A Commutative Encrypted Protocol for Piracy Protection of watermarks in Digital Contents", in proceedings of the 37th Hawaii international Conference on System Sciences", 2004, page no. 1-10.