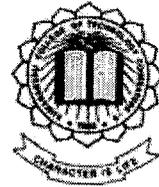




P-2287



TEXT FILE HIDING IN AUDIO FILES USING LOW BIT ENCODING

By

S.A.SARAVANRAJ
(Registration Number: 71205621039)

Of

**KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE**

A PROJECT REPORT
Submitted to the

**FACULTY OF INFORMATION AND COMMUNICATION
ENGINEERING**

*In partial fulfillment of the requirements
For the award of the degree*

Of

MASTER OF COMPUTER APPLICATION

JUNE 2008

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

DEPARTMENT OF COMPUTER APPLICATION**Bonafide Certificate**

Certified that this project report titled **TEXT FILE HIDING IN AUDIO FILES USING LOW BIT ENCODING** is the bonafide work of **Mr.S.A.SARAVANRAJ (Registration Number: 71205621039)** who carried out the research under my supervision. Certified further, that to the best of my Knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.


Supervisor


Head of the Department

Submitted to Project and Viva Examination held on 01.07.08


Internal Examiner


External Examiner

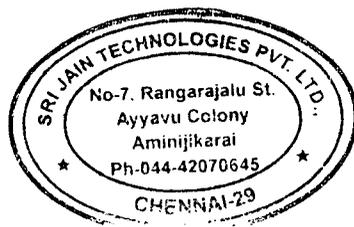
To Whomsoever It May Concern

This is to certify that Mr.SARAVANRAJ.S.A (71205621039), Student of Kumaraguru College of Technology, Coimbatore doing his Final Year, M.C.A (Computer Applications) has successfully completed his project entitled as "TEXT FILE HIDING IN AUDIO(WAV) FILES USING LOWBIT ENCODING STEGNOGRAPHY" under the guidance of Mr.Nandakumar.S, Senior Software Engineer, from December 2007 to May 2008.

During his Project duration his conduct and contribution has been excellent.

We wish all the best for his future Endeavors.

For Sri Jain Technologies Pvt.Ltd




(Project Co-ordinator)

ABSTRACT

This Project Titled “Text File Hiding in Audio files using Low Bit Encoding” is the software developed for hiding information which uses the technology called as Steganography which is the art of hiding information in ways that prevent its detection. A message in cipher text may arouse suspicion while an invisible message is not. The conventional way of protecting information was to use a standard symmetric or asymmetric key system in encryption. Steganography can also be used to place a hidden “trademark” in images, music, and software, a technique referred to as watermarking.

Steganography, if however used along with cryptography then the message will become quite secure as far as cryptanalytic attacks are concerned. Now, if this cipher text is embedded in an image, video, voice, etc., it is even more secure. If an encrypted message is intercepted, the interceptor knows the text is an encrypted message. With Steganography, the interceptor may know the object contains a message.

When performing data hiding on audio, one must exploit the weakness of the Human Auditory System (HAS), while at the same time being aware of the extreme sensitivity of the human auditory system. First the audio file which behaves as carrier file is selected. Then a message or a text file to be embedded in audio is selected. Then a key file is selected. The key file contains characters whose ASCII values are taken for encryption. Then the file is embedded in audio using low bit encoding mechanism and extraction of embedded message is being done vice-versa.

ACKNOWLEDGEMENT

I would like to take this opportunity to say thank you to the people who have helped to make this project.

First I would like to express my heartfelt thanks to our principal Dr. Joseph V. Thanikal, B.E., M.E., Ph.D., PDF., CEPIT., for having given me the opportunity to do this project.

I express my deep sense of gratitude and profound thanks to Dr. M. Gururajan, Ph.D., HOD, Computer Application, for providing moral support towards this project work.

I tender my special thanks to Mrs. V. Geetha M.C.A., M.Phil, Assistant Professor and Project Coordinator, Department of Computer Applications for his constant encouragement.

I sincerely express my humble gratitude to Mr. C. Rajan Krupa, M.C.A., Senior Lecturer, Department of Computer Application, who has been my guide with valuable and holistic suggestions and extended kind of operation and encouragement to make this project.

I also express my deep sense of gratitude and profound thanks to Mr. S. Nandakumar, Senior Software Engineer, Sri Jain Technologies Ltd, who has been my guide and gave valuable suggestions and encouragement.

I acknowledge my hearty thanks to all my beloved friends on their valuable co-operation in the proceeding of my work.

I wish to credit my special accordance to my parents for their encouragement and prayers to have a successful project.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF FIGURES	vii
1. INTRODUCTION	
1.1 ABOUT THE PROJECT	1
1.2 STEGANOGRAPHY IN AUDIO	1
1.3 PROJECT DESCRIPTION	3
1.4 ORGANIZATION PROFILE	4
2. SYSTEM STUDY AND ANALYSIS	
2.1 EXISTING SYSTEM	6
2.2 PROPOSED SYSTEM	6
2.3 FEASIBILITY STUDY	6
2.3.1 ECONOMICAL FEASIBILITY	6
2.3.2 TECHNICAL FEASIBILITY	7
2.3.3 SOCIAL FEASIBILITY	7
3. DEVELOPMENT ENVIRONMENT	
3.1 HARDWARE REQUIREMENTS	8
3.2 SOFTWARE REQUIREMENTS	8
3.3 PROGRAMMING ENVIRONMENT	9
4. SYSTEM DESIGN AND DEVELOPMENT	
4.1 ELEMENTS OF DESIGN	11
4.1.1 INPUT DESIGN	11
4.1.2 OUTPUT DESIGN	11
4.2 DATA FLOW DIAGRAM	12

4.3 USECASE DIAGRAM	13
4.4 CLASS DIAGRAM	15
5. ARCHITECTURAL DESIGN	
5.1 MODULE DESIGN	16
5.2 AUDIO FILE FORMAT	17
6. SYSTEM TESTING AND IMPLEMENTATION	
6.1 GOALS OF TESTING	23
6.2 TYPES OF TESTING	24
6.2.1 FUNCTIONAL TESTING	24
6.2.2 STRUCTURAL TESTING	24
6.3 DIFFERENT LEVELS OF TESTING	24
6.3.1 UNIT TESTING	24
6.3.3 SYSTEM TESTING	25
7. PERFORMANCE AND LIMITATIONS	
7.1 MERITS OF THE SYSTEM	26
7.2 LIMITATIONS OF THE SYSTEM	26
7.3 FUTURE ENHANCEMENT	26
8. APPENDICES	
8.1 SNAPSHOTS	27
9. REFERENCES	37

LIST OF FIGURES

<u>FIGURE DESCRIPTION</u>	<u>PAGE NO</u>
Figure 4.2.1 DATA FLOW DIAGRAM	12
Figure 4.3.1 USECASE DIAGRAM: EMBEDDING	13
Figure 4.3.2 USECASE DIAGRAM: EXTRACTION	14
Figure 4.4.1 CLASS DIAGRAM	15

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT:

We are of the belief that the easiest way to keep something from prying eyes is to place it right in front of the person looking for it and make it look as innocuous as possible.

Everyone has a taste for a certain kind of music. Hence, it is more than likely that the person will have that kind of music on the storage device of his computer. Also, it is quite common case where people share and transfer different music files to one another. If one were able to hide the message can be. Also, transfer of this message can be done quite conveniently without raising any eyebrows.

Our aim is to come up with a technique of hiding the message in the audio file in such a way, that there would be no perceivable changes in the audio file after the message insertion. At the same time, if the message that is to be hidden were encrypted, the level of security would be raised to quite a satisfactory level. Now, even if the hidden message were to be discovered the person trying to get the message would only be able to lay his hands on the encrypted message with no way of being able to decrypt it.

1.2 STEGANOGRAPHY IN AUDIO:

Data hiding in audio signals is especially challenging, because the Human Auditory System (HAS) operates over a wide dynamic range. The HAS perceives over a range of power greater than one billion to one and a range of frequencies greater than thousand to one. Sensitivity to additive random noise is also acute.

The perturbations in a sound file can be detected as low as one part in ten million which is 80dB below ambient level. However there are some 'holes' available. While the HAS has a large dynamic range, it has a fairly small differential range. As a result, loud sounds tend to mask out the quieter sounds.

Additionally, the HAS is unable to perceive absolute phase, only relative phase. Finally there are some environmental distortions so common as to be ignored by the listener in most cases.

1.2.1 Low-bit encoding:

Low-bit encoding is the one of the simplest way to embed data into other data structures. By replacing the least significant bit of each sampling point by a coded binary string, we can encode a large amount of data in an audio signal.

Ideally, the channel capacity is 1 kb per second (kbps) per 1 kilohertz(kHz), e.g., in a noiseless channel, the bit rate will be 8 kbps in an 8 kHz sampled sequence and 44 kbps in a 44kHz sampled sequence. In return for this large channel capacity, audible noise is introduced. The impact of this noise is a direct function of the content of the host signal, e.g., crowd noise during a live sports event would mask low-bit encoding noise that would be audible in a string quartet performance.

Adaptive data attenuation has been used to compensate this variation. The major advantage of this method is its poor immunity to manipulation. Encoded information can be destroyed by channel noise, re-sampling, etc., unless it is encoded using redundancy techniques.

In order to be robust, these techniques reduce the data rate which could result in the requirement of a host of higher magnitude, often by one to two orders of magnitude. In practice, this method is useful only in closed, digital-to-digital environments.

1.3.2. Extract module (To extract the text file from the audio file)

In this module, the first step is the process of selecting the encrypted audio file. This is the file that a user has to extract information from the output audio. Second process involved in selecting a new text file to display the embedded message. Symmetric encryption method is used here, so the key selected during the embedding process is used in decrypting the message. All the process done till now are displayed using a list box and finally the embedded message can be viewed with the help of a file or in a textbox.

1.4 ORGANIZATION PROFILE

COMPANY PROFILE.

At Sri Jain Technologies, We go beyond providing software solutions. We work with our client's technologies and business changes that shape their competitive advantages.

Founded in 2000, Sri Jain Technologies (P) Ltd. is a software and service provider that helps organizations deploy, manage, and support their business-critical software more effectively. Utilizing a combination of proprietary software, services and specialized expertise, Sri Jain Technologies (P) Ltd. helps mid-to-large enterprises, software companies and IT service providers improve consistency, speed, and transparency with service delivery at lower costs. Sri Jain Technologies (P) Ltd. helps companies avoid many of the delays, costs and risks associated with the distribution and support of software on desktops, servers and remote devices. Our automated solutions include rapid, touch-free deployments, ongoing software upgrades, fixes and security patches, technology asset inventory and tracking, software license optimization, application self-healing and policy management. At Sri Jain Technologies, we go beyond providing software solutions. We work with our clients' technologies and business processes that shape their competitive advantages.

Our Vision

“Dreaming a vision is possible and realizing it is our goal”.

Our Mission

We have achieved this by creating and perfecting processes that are in par with the global standards and we deliver high quality, high value services, reliable and cost effective IT products to clients around the world.

CHAPTER 2

SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM:

Now a day, several methods are used for communicating secret messages for defense purposes or in order to ensure the privacy of communication between two parties. So we go for hiding information in ways that prevent its detection. Some of the methods used for privacy communication are the use of invisible links; covert channels are some of existing systems that are used to convey the messages.

2.2 PROPOSED SYSTEM:

The proposed system uses audio file as a carrier medium which add another step in security. The objective of the newly proposed system is to create a system that makes it very difficult for an opponent to detect the existence of a secret message by encoding it in the carrier medium as a function of some secret key and that remains as the advantage of this system.

2.3 FEASIBILITY STUDY:

ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

This section describes the hardware components with which the application was developed and the minimum hardware configuration with which the system operates best.

PROCESSOR	:	Pentium IV
PROCESSOR SPEED	:	2.4 GHz
RAM	:	512 Mb
HARD DISK SIZE	:	80 GB
KEYBOARD	:	Multimedia Keyboard
MOUSE	:	Optical
DISPLAY	:	1024 X 768

3.2 SOFTWARE REQUIREMENTS

This section describes the software in which the application was developed and using the same software would make it more compatible.

OPERATING SYSTEM	:	Windows XP
FRONT END	:	Microsoft Visual Studio .Net 2003
PROGRAMMING LANGUAGE	:	VB .Net

3.3 PROGRAMMING ENVIRONMENT

ABOUT SOFTWARE

What is .NET?

When .NET was announced in late 1999, Microsoft positioned the technology as a platform for building and consuming Extensible Markup Language (XML) Web services. XML Web services allow any type of application, be it a Windows or browser-based application running on any type of computer system, to consume data from any type of server over the Internet. The reason this idea is so great is the way in which the XML messages are transferred over established standard protocols that exist today. Using protocols such as SOAP, HTTP, and SMTP, XML Web services make it possible to expose data over the wire with little or no modifications to your existing code.

Figure 1.1 presents a high-level overview of the .NET Framework and how XML Web services are positioned.

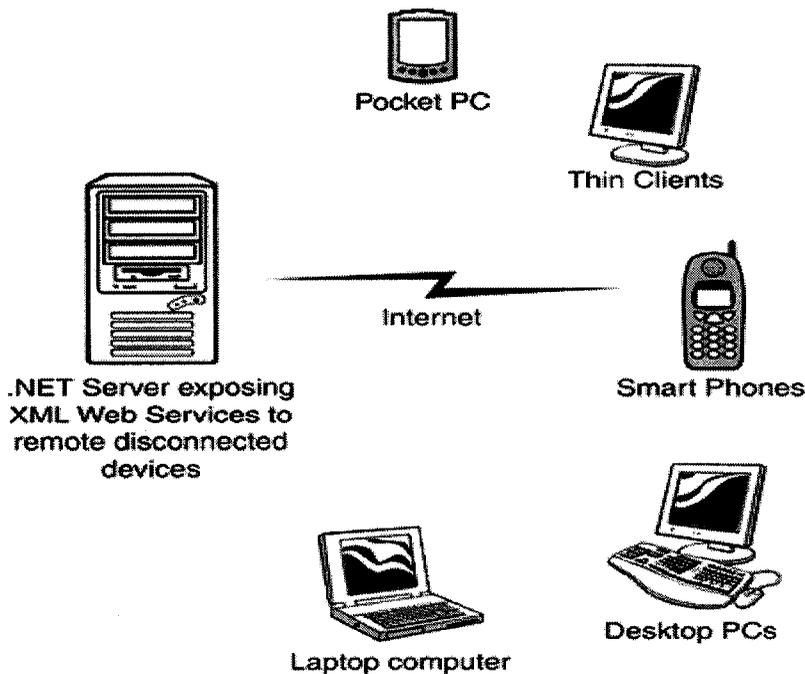


Figure 1.1 Stateless XML Web services model.

Since the initial announcement of the .NET Framework, it's taken on many new and different meanings to different people. To a developer, .NET means a great environment for creating robust distributed applications. To an IT manager, .NET means simpler deployment of applications to end users, tighter security, and simpler management. To a CTO or CIO, .NET means happier developers using state-of-the-art development technologies and a smaller bottom line. To understand why all these statements are true, you need to get a grip on what the .NET Framework consists of, and how it's truly a revolutionary step forward for application architecture, development, and deployment.

Microsoft Visual Basic.Net

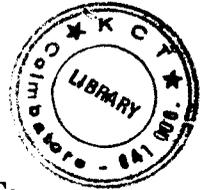
With its release for the .NET platform, the Visual Basic language has undergone dramatic changes.

For example:

- The language itself is now fully object-oriented.
- Applications and components written in Visual Basic .NET have full access to the .NET Framework, an extensive class library that provides system and application services.
- All applications developed using Visual Basic .NET run within a managed runtime environment, the .NET common language runtime.

Visual Basic .NET is the next generation of Visual Basic, but it is also a significant departure from previous generations. Experienced Visual Basic 6 developers will feel comfortable with Visual Basic .NET code and will recognize most of its constructs. However, Microsoft has made some changes to make Visual Basic .NET a better language and an equal player in the .NET world. These include such additions as a Class keyword for defining classes and an Inherits keyword for object inheritance, among others. Visual Basic 6 code can't be compiled by the Visual Basic .NET compiler without significant modification. The good news is that Microsoft has provided a migration tool

CHAPTER 4



P-2287

SYSTEM DESIGN AND DEVELOPMENT:

4.1 ELEMENTS OF DESIGN

4.1.1 Input Design

Input Design is the part or overall system design, which requires very careful attention. Often the collection of input data is the most expensive part of the system. In terms of both the equipment used and the number of people involved in it is the point of most contracts for the user with the computer system and it is prone to error. If data going into system is incorrect, then the processing the output will magnify these errors. Input design is the process of converting an external user oriented description of the input system into a machine-oriented format.

4.1.2 Output Design

One of the most important features of an information system for users is the output that is produced. Without quality output the entire system might appear to be so unnecessary that users will avoid using it, possible causing the system to fail, right output must be developed while ensuring the output element is designed so that people will find the system easy to use effectively.

Output screens are the tools to convey information to the users since the design of the output screen is very important for attracting the users; the output screens are designed in such a way that it is very interactive and informative. The outputs from the computer systems are primarily to communicate the results of processing to users.

The output screen in this project gives information how to extract the hidden file from the audio file. The above information is displayed on an output screen with appropriate format. The software generates an acknowledgement on successful

4.2 DATA FLOW DIAGRAM:

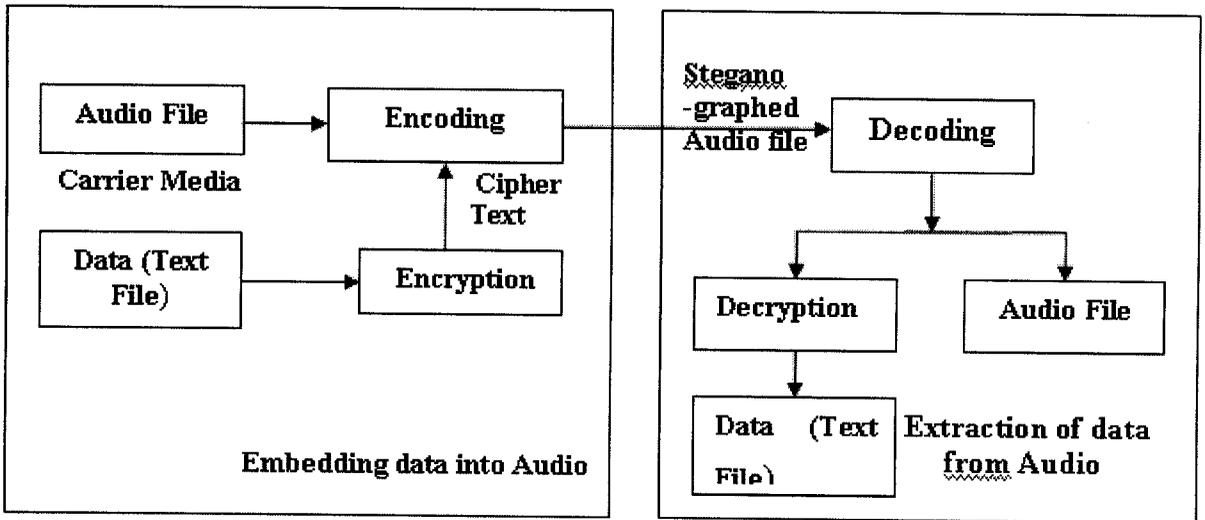


FIGURE 1.1

4.3 USECASE DIAGRAM:

4.3.1 USECASE DIAGRAM1: (EMBEDDING MODULE):

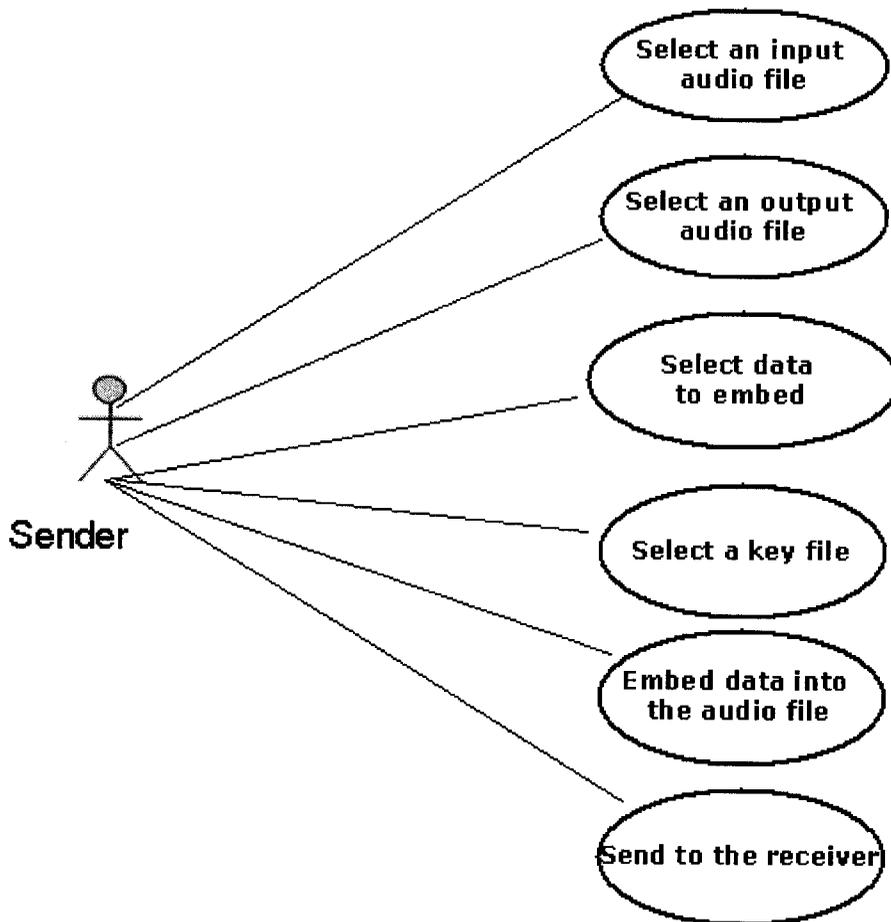


FIGURE 1.2

4.3.2 USECASE DIAGRAM2: (EXTRACTION MODULE):

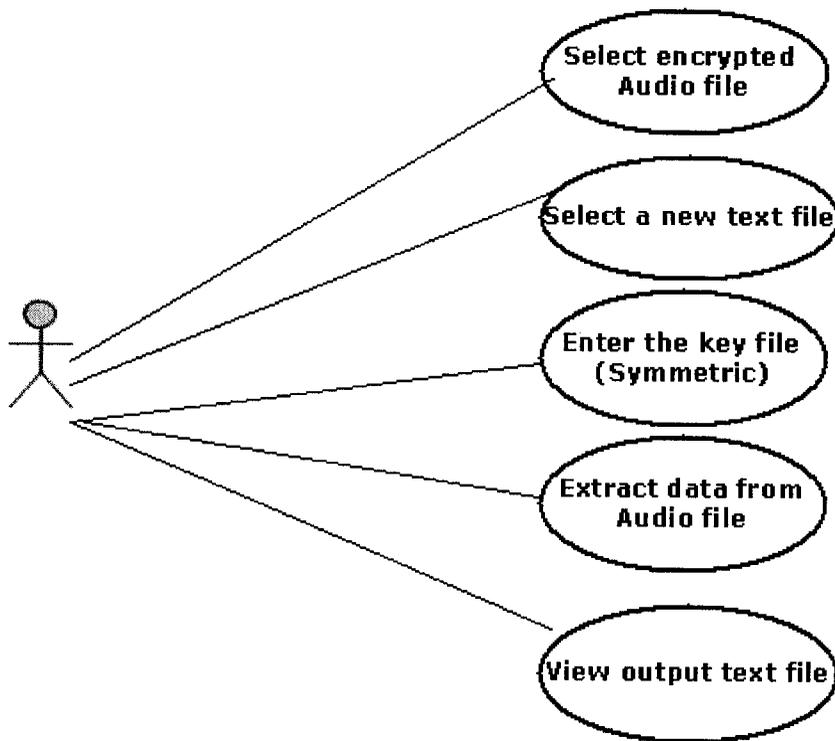


FIGURE1.3

4.4 CLASS DIAGRAM

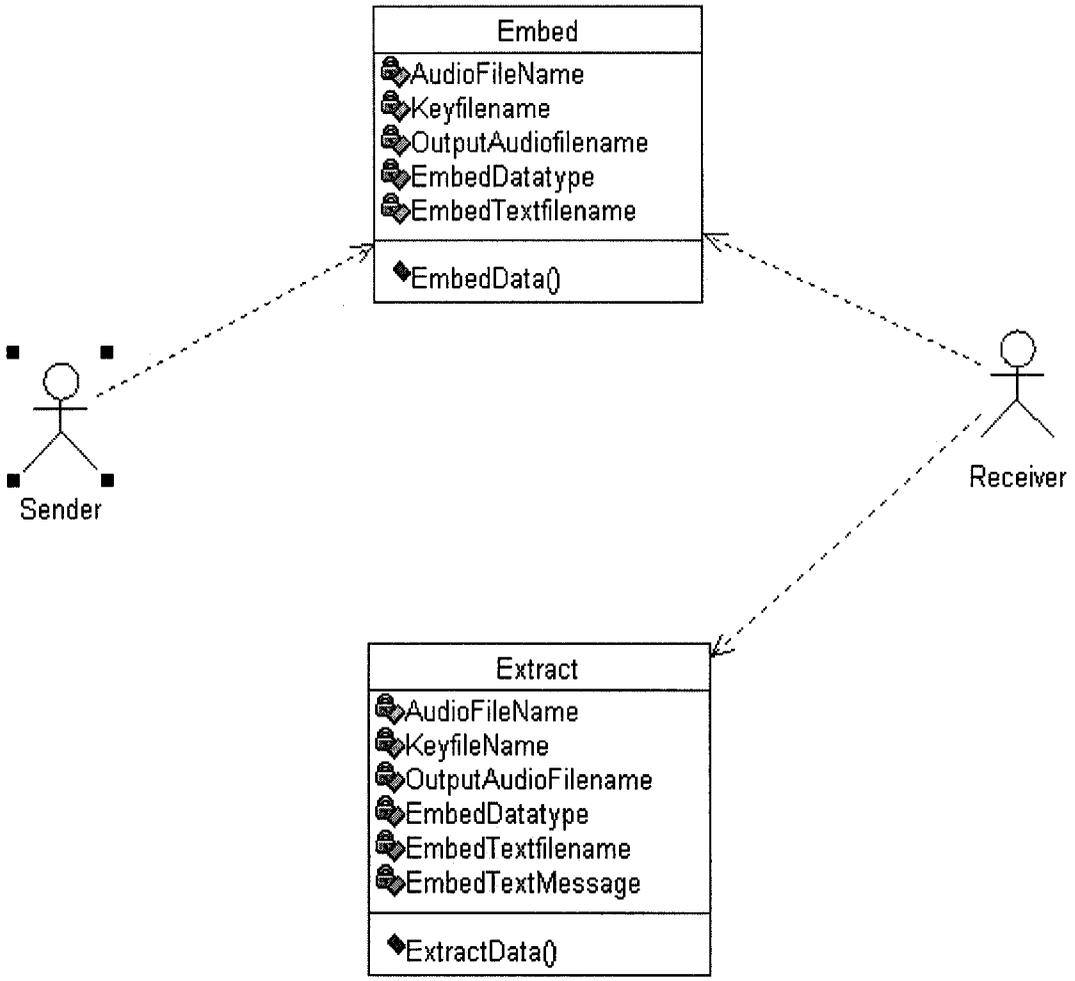


FIGURE 1.4

CHAPTER 5

ARCHITECTURAL DESIGN

5.1 MODULE DESIGN

The entire project is mainly divided into two major modules. They are

- Embed module
- Extract module

Embed module

In this module, the first step is selecting an input audio file. The selection is made through opening a new dialog box and the path selected is displayed through a textbox. The second step is selecting an output audio file in which text data or a text file is embedded. The third step is choosing a text file or typing any text message for embedding. Fourth step is selecting a key file. In the fifth step what ever the files that we have selected are viewed and verification of the path is done. In the sixth process data is embedded in to the audio file using low bit encoding technique. After embedding the content both the audio files are played and a listener cannot find any difference between the audios.

Extract module

In this module, the first step is the process of selecting the encrypted audio file. This is the file that a user has to extract information from the output audio. Second process involved in selecting a new text file to display the embedded message. Symmetric encryption method is used here, so the key selected during the embedding process is used in decrypting the message. All the process done till now are displayed using a list box and finally the embedded message can be viewed with the help of a file or in a textbox.

5.2 AUDIO FILE FORMAT:

An audio format is a medium for storing sound and music. It is a container format for storing audio data on a computer system. There are numerous file formats for storing audio files.

The general approach towards storing digital audio formats is to sample the audio voltage (corresponding to a certain position in the membrane of a speaker) in regular intervals (e.g. 44,100 times per second for CD audio or 48,000 or 96,000 times per second for DVD video) and store the value with a certain resolution (e.g. 16 bits per sample in CD audio). Therefore sample rate, resolution and number of channels are key parameters in audio file formats.

5.2.1 TYPES OF FORMATS:

There are three major groups of audio file formats:

- 1) Common formats, such as WAV, AIFF and AU.
- 2) Formats with lossless compression, such as FLAC, Monkey's Audio (Filename extension APE), WavPack, Shorten, TTA, Apple Lossless, and lossless Windows Media Audio (WMA).
- 3) Formats with lossy compression, such as MP3, Vorbis, lossy Windows Media Audio (WMA) and AAC.

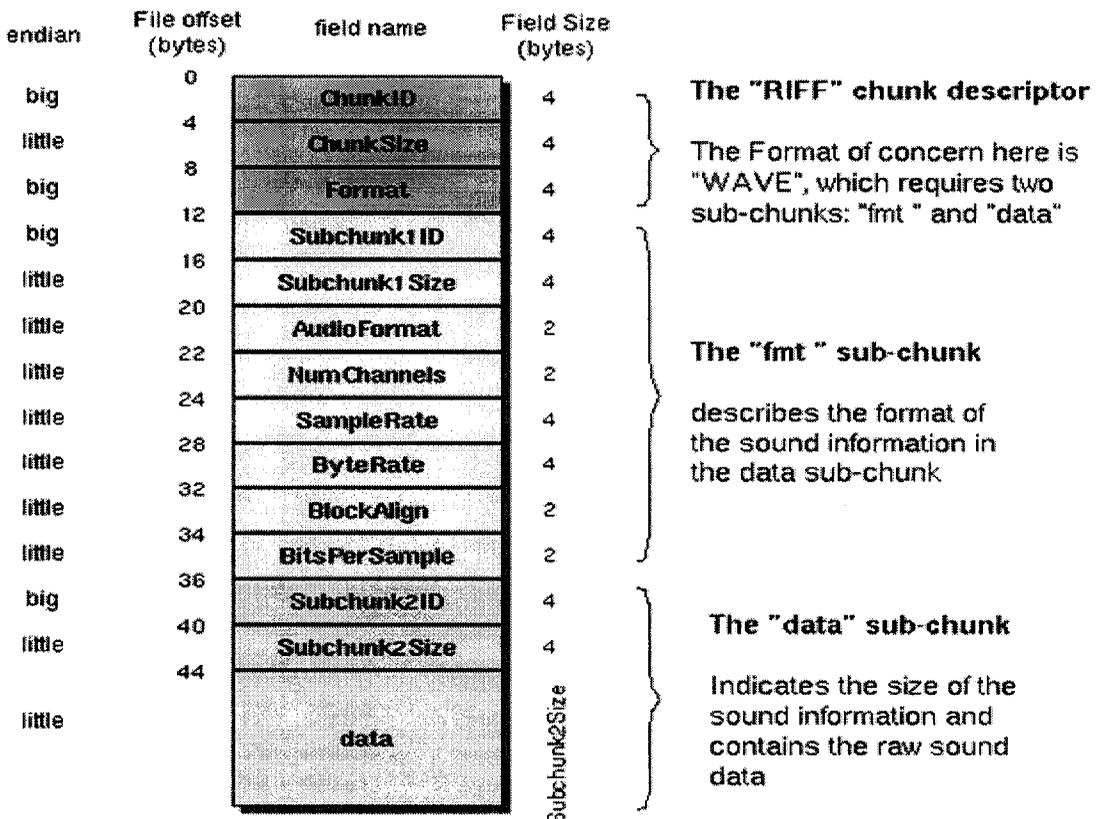
WAV is a flexible file format designed to store more or less any combination of sampling rates or bitrates. This makes it an adequate file format for storing and archiving an original recording.

5.2.2 WAV FORMAT:

WAV (or **WAVE**), short for Waveform audio format, is a Microsoft and IBM audio file format standard for storing audio on PCs. It is a variant of the RIFF bit stream format method for storing data in "chunks".

A WAVE file is often just a RIFF file with a single "WAVE" chunk which consists of two sub-chunks -- an "fmt" chunk specifying the data format and a "data" chunk containing the actual sample data.

The Canonical WAVE file format



The canonical WAVE format starts with the RIFF header:

- 0 - 4 **ChunkID** Contains the letters "RIFF" in ASCII form

- 4 - 4 **ChunkSize** 36 + SubChunk2Size, or more precisely:
4 + (8 + SubChunk1Size) + (8 + SubChunk2Size)
This is the size of the rest of the chunk
following this number. This is the size of the
entire file in bytes minus 8 bytes for the
two fields not included in this count:
ChunkID and ChunkSize.

- 8 - 4 **Format** Contains the letters "WAVE"

The "WAVE" format consists of two subchunks: "fmt" and "data":

The "fmt" subchunk describes the sound data's format:

- 12 - 4 **Subchunk1ID** Contains the letters "fmt"

- 16 - 4 **Subchunk1Size** 16 for PCM. This is the size of the
rest of the Subchunk which follows this
number.

- 20 - 2 **AudioFormat** PCM = 1 (i.e. Linear quantization)
Values other than 1 indicate some
form of compression.

- 22 - 2 **NumChannels** Mono = 1, Stereo = 2, etc.

- 24 - 4 **SampleRate** 8000, 44100, etc.

- 28 - 4 **ByteRate** = SampleRate * No Channels * BitsPerSample/8

- 32 - 2 **BlockAlign** = NumChannels * BitsPerSample/8
The number of bytes for one sample including
all channels.

The "data" subchunk contains the size of the data and the actual sound:

- 36 - 4 **Subchunk2ID** Contains the letters "data"
 - 40 - 4 **Subchunk2Size** = NoSamples * NoChannels* BitsPerSample/8
This is the number of bytes in the data.
You can also think of this as the size
of the read of the subchunk following this
number.
 - 44 * **Data** The actual sound data.
-

As an example, here are the opening 72 bytes of a WAVE file with bytes shown as hexadecimal numbers:

52 49 46 46 24 08 00 00 57 41 56 45 66 6d 74 20 10 00 00 00 01 00 02 00
22 56 00 00 88 58 01 00 04 00 10 00 64 61 74 61 00 08 00 00 00 00 00
24 17 1e f3 3c 13 3c 14 16 f9 18 f9 34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d

Here is the interpretation of these bytes as a WAVE sound file:

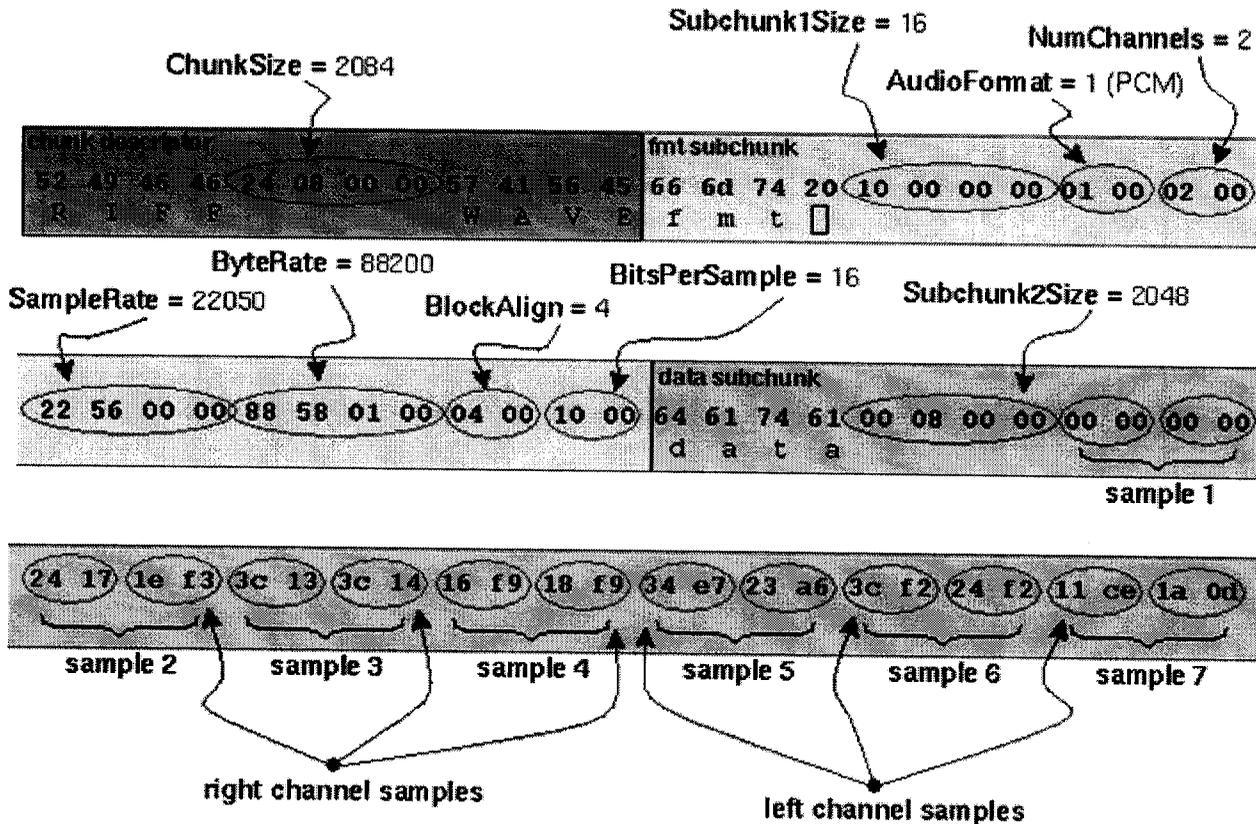


FIGURE 1.2

Though a WAV file can hold compressed audio, the most common WAV format contains uncompressed audio in the pulse-code modulation (PCM) format. PCM audio is the standard audio file format for CDs at 44,100 samples per second, 16 bits per sample. Since PCM uses an uncompressed, lossless storage method, which keeps all the samples of an audio track, professional users or audio experts may use the WAV format for maximum audio quality. WAV audio can also be edited and manipulated with relative ease using software.

The usage of the WAV format has more to do with its familiarity, its simplicity and simple structure, which is heavily based on the IFF file format. Because of this, it continues to enjoy widespread use with a variety of software applications.

Limitations:

The WAV format is limited to files that are less than 4 GB in size; due to its use of a 32 bit unsigned integer to record the file size header (some programs limit the file size to 2 GB). It is sometimes necessary to go over this limit, especially when higher sampling rates or bit resolutions are required. The **W64** format was therefore created for use in Sound Forge. Its 64-bit header allows for much longer recording times.

CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

Testing is the process of executing the program with the intent of finding errors. During testing, the program to be tested is executed with a set of test cases and the output of the program for the test cases is evaluated to determine the program is performing as it is expected. Error is the testing fundamental and is defined as the difference between the actual output of a software and a correct output i.e., difference between the actual and ideal testing is usually relied upon to detect these faults in the coding phase for this, different levels of testing are used which performs different tasks and aim to the test different aspects of the system.

6.1 GOALS OF TESTING:

The famous statement by *Dijkstra* (in Dahl et al. 1972) is a perfect synthesis of the goals of the testing. If the results delivered by the system are different from the expected ones in just one case, in this unequally shows that the system is incorrect: by contrast, a correct behavior of the system on a finite number of cases does not guarantee correctness in the general case. For instance, we could have built a program that behaves properly for even integer numbers but not odd numbers. Clearly, any number of tests with even input values will face to show the error.

Testing should be based on *sound* and *systematic techniques* so that, after testing, we may have a better understanding of the product's reliability.

Testing should help *locate errors*, not just detect their presence. The result of testing should not be viewed as simply providing a Boolean answer to the question of whether the software works properly or not.

Tests should be *organized* in a way that helps to isolate errors. This information

Testing should be *repeatable*, i.e., tests should be arranged in such a way that separating the same experiment-supplying the same input data to the same piece of code – produces the same results.

Finally testing should be *accurate* – this will increase the reliability of testing. Here we should observe that the accuracy of the testing activity depends on the level of precision and may be even formality of software specifications.

6.2 TYPES OF TESTING:

Functional Testing

Here the structure of the program is not considered. Only the test cases are decided solely on the basis of the requirements or specification of the program or module and the internal details of the module or the program is not considered for the selection of test cases. This is also called “*Black Box Testing*”.

Structural Testing

It is considered with testing the implementation of the program. The intention of the structural testing is not to exercise all the different input and output conditions but to exercise the different programming and data files used in the program. This testing is also called “*White Box Testing*”.

6.3 DIFFERENT LEVELS OF TESTING:

Unit Testing

In it different modules are tested against the specifications produced during design for the modules. It is essential for verification of the code produced during the code phase and the goal is to test the internal logic of the module.

System Testing

Here the entire software is tested. The reference document for this process is the requirement document and the goal is to see whether the software needs its requirements. The system was tested for various test cases with various inputs.

CHAPTER 7

PERFORMANCE AND LIMITATIONS

7.1 MERITS

Our aim is to come up with a technique of hiding the message in the audio file in such a way, that they would be no perceivable changes in the audio file after the message insertion. At the same time, if the message that is to be hidden were encrypted, the level of security would be raised to quite satisfactory level.

7.2 LIMITATIONS OF THE SYSTEM

Everyone has a taste for certain kind of music. Hence, it is more than likely that the person will have that kind of music on the storage device of his computer also; it is quite common case where people shared and transfer different music files to one another. If one were able to hide the message can be.

7.3 FUTURE ENHANCEMENT

- * Can be sent through the network, and that can be extracted at the other end.
- * Support for other audio formats such as (wav, mp3) can easily be added which needs additional sound encoding/decoding method to be implemented.

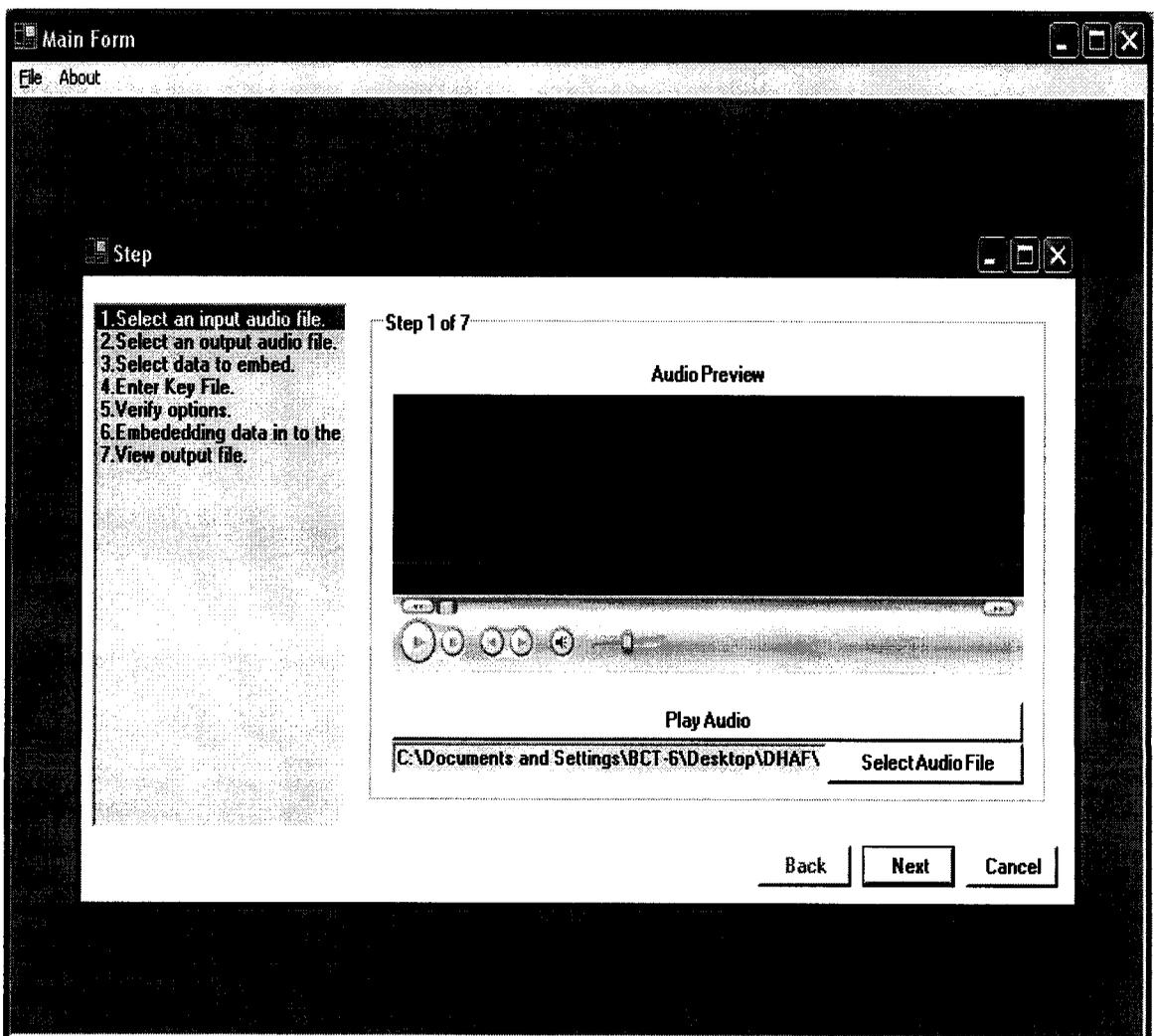
CHAPTER 8

APPENDICES

8.1 SNAP SHOTS:

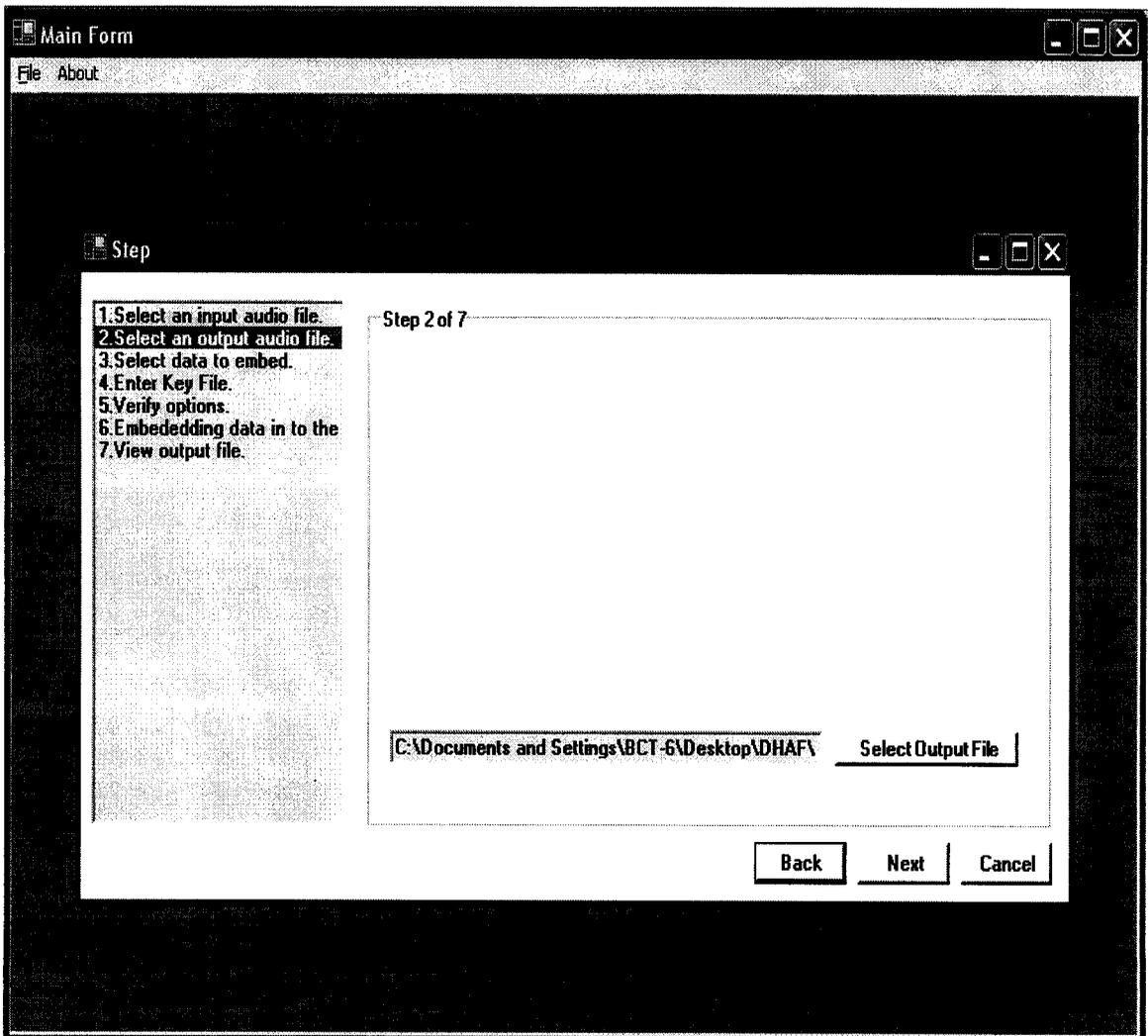
EMBEDDING MODULE:

Step 1: Selection of an input audio file. This is the wav file in which the user is going to hide the text file.



EMBEDDING MODULE

Step 2: Selection of an output Audio file. This is the name of the file given by the user to save the message embedded audio file.



EMBEDDING MODULE:

Step 3: Selection of the text file (or the message) to be hidden and sent to the destination.

The screenshot shows a software application window titled "Main Form" with a menu bar containing "File" and "About". Inside the window, there is a smaller window titled "Step" which contains a list of seven steps. Step 3, "Select data to embed", is highlighted. The main area of the "Step" window is titled "Step 3 of 7" and contains two radio buttons: "Select File" (unselected) and "Enter Text" (selected). Below the radio buttons is a text input field containing the word "welcome". At the bottom of the main area, there is a label "FileName" followed by an empty text input field and a button labeled "Select Text File". At the very bottom of the "Step" window are three buttons: "Back", "Next", and "Cancel".

1. Select an input audio file.
2. Select an output audio file.
3. Select data to embed.
4. Enter Key File.
5. Verify options.
6. Embedding data in to the
7. View output file.

Step 3 of 7

Select File Enter Text

welcome

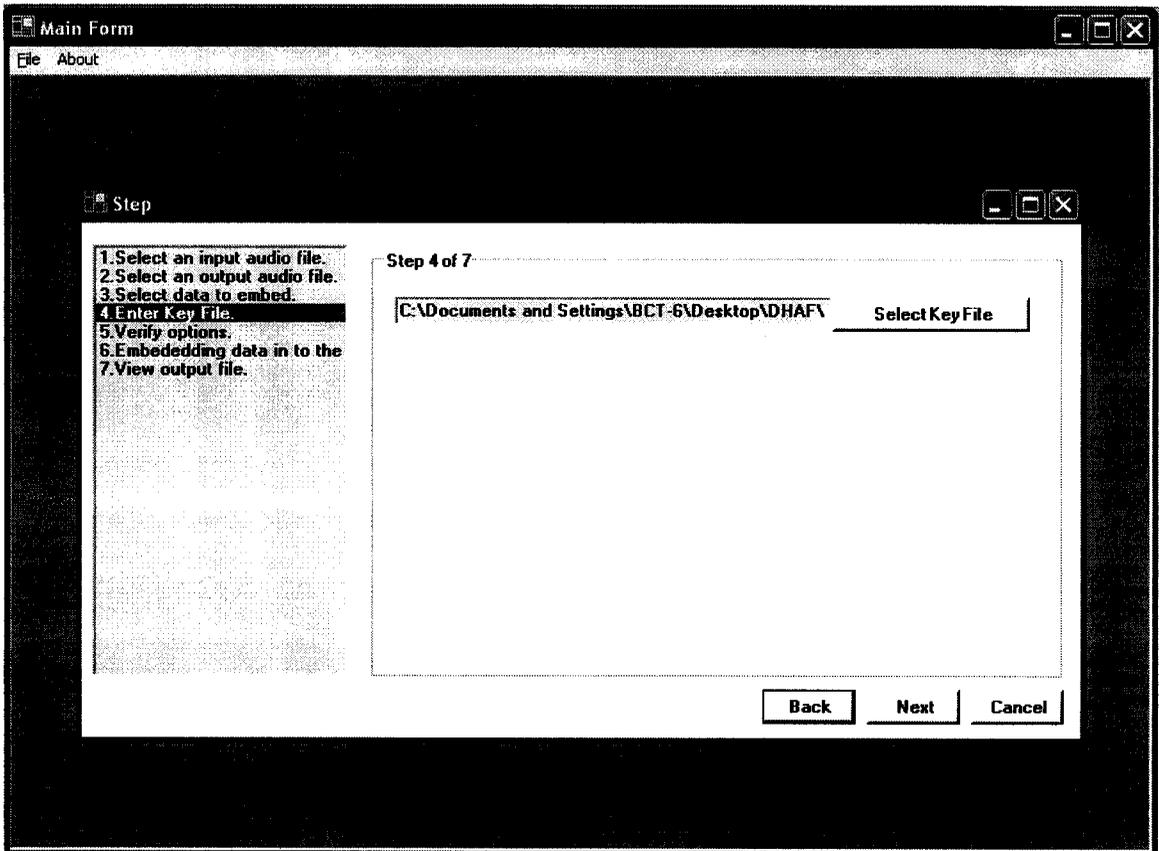
FileName

Select Text File

Back Next Cancel

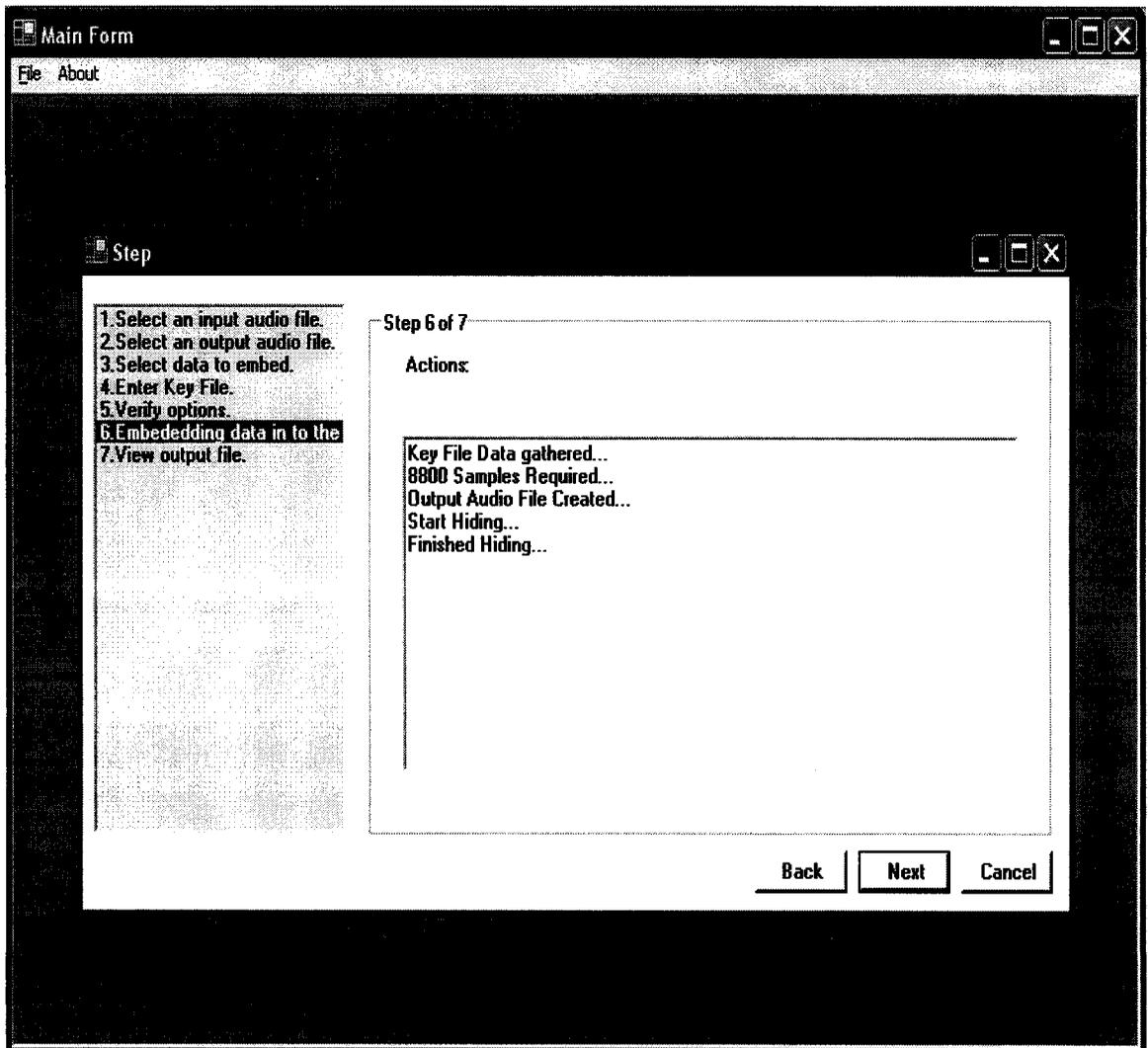
EMBEDDING MODULE:

Step 4: Selection of the Key file (Used for Encryption)



EMBEDDING MODULE:

Step 6: Actual process of embedding the text file into the selected audio file.

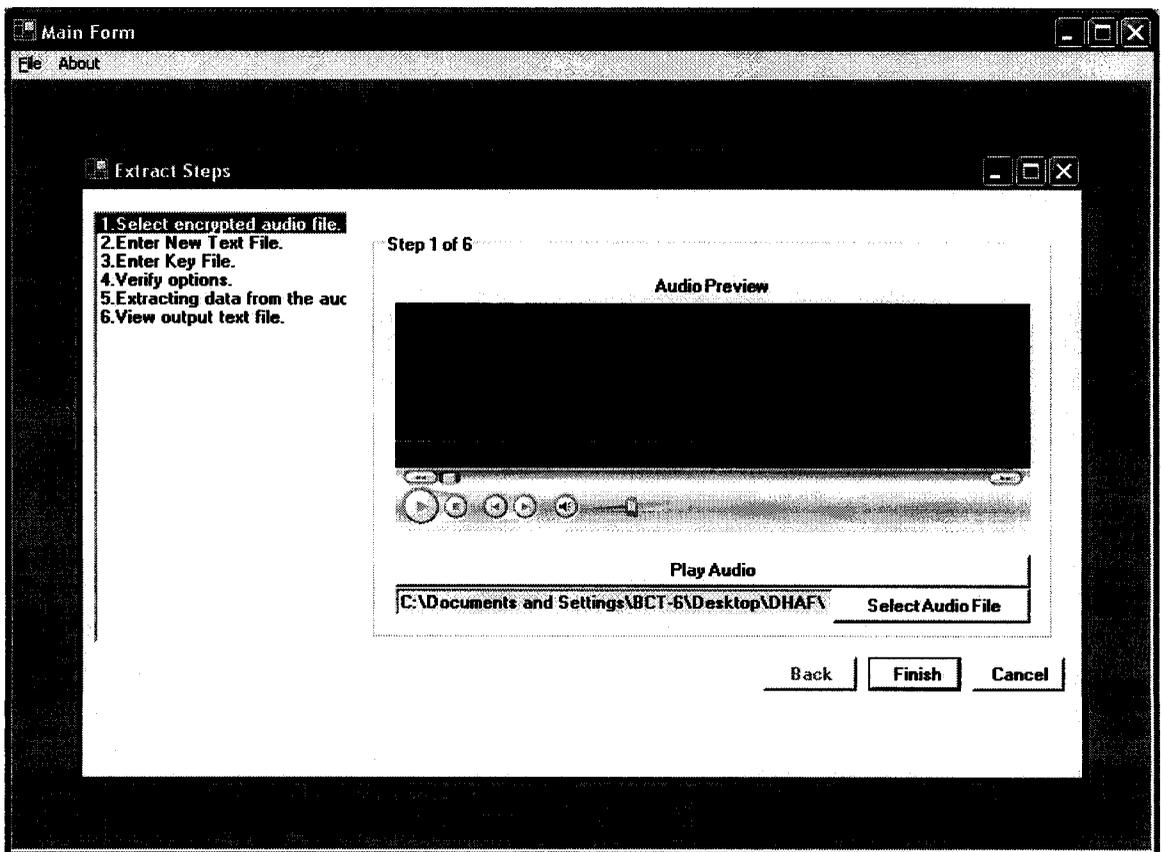


The data file is successfully hidden in the selected audio file. After embedding process, when we play the encrypted audio file it will listen the same as before without any changes.

EXTRACTION MODULE:

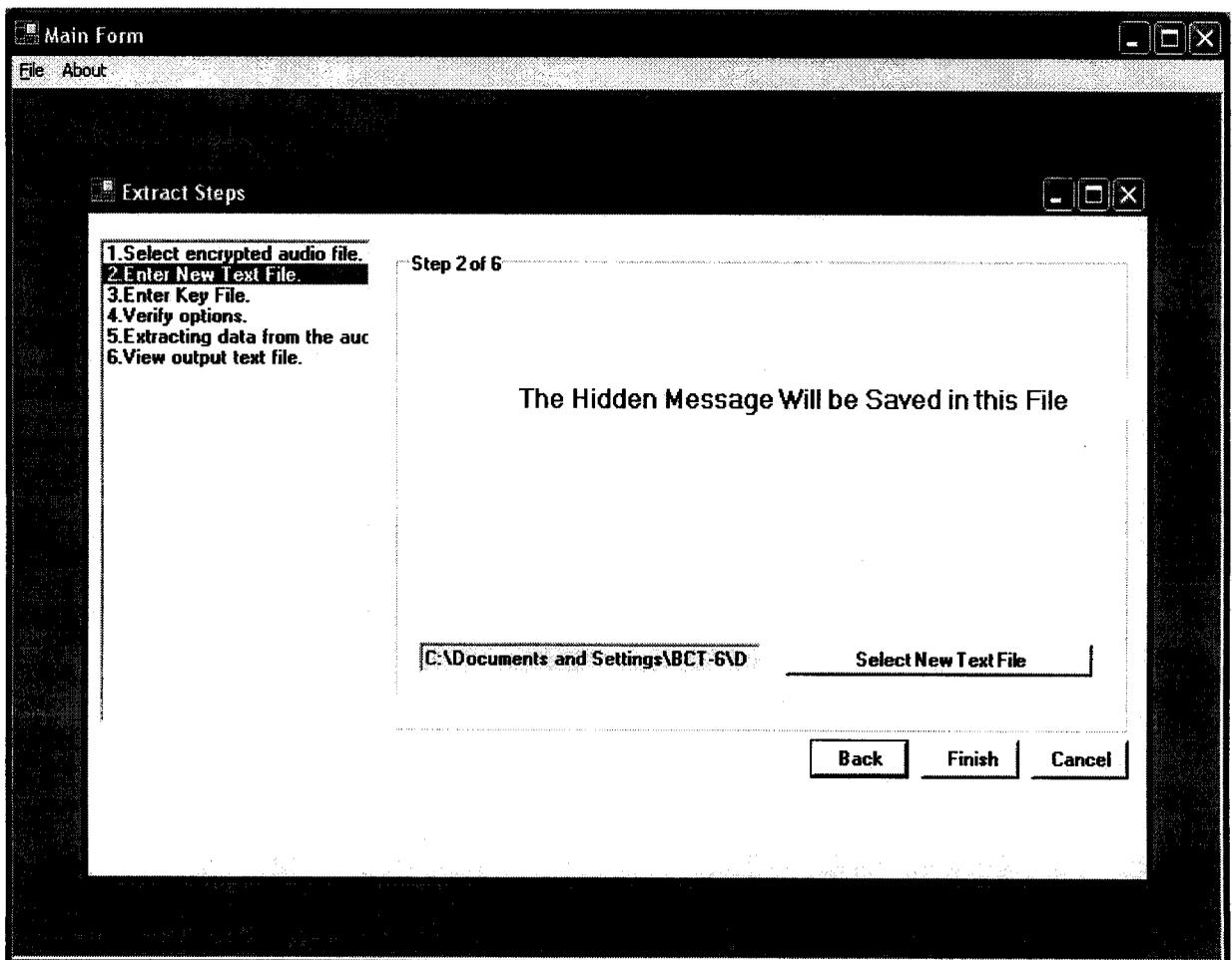
The Hidden message is extracted from the audio file and can be viewed through the following process.

Step 1: Selection of the Encrypted audio file. This is the name of the file given in step 2 of the Embedding module.



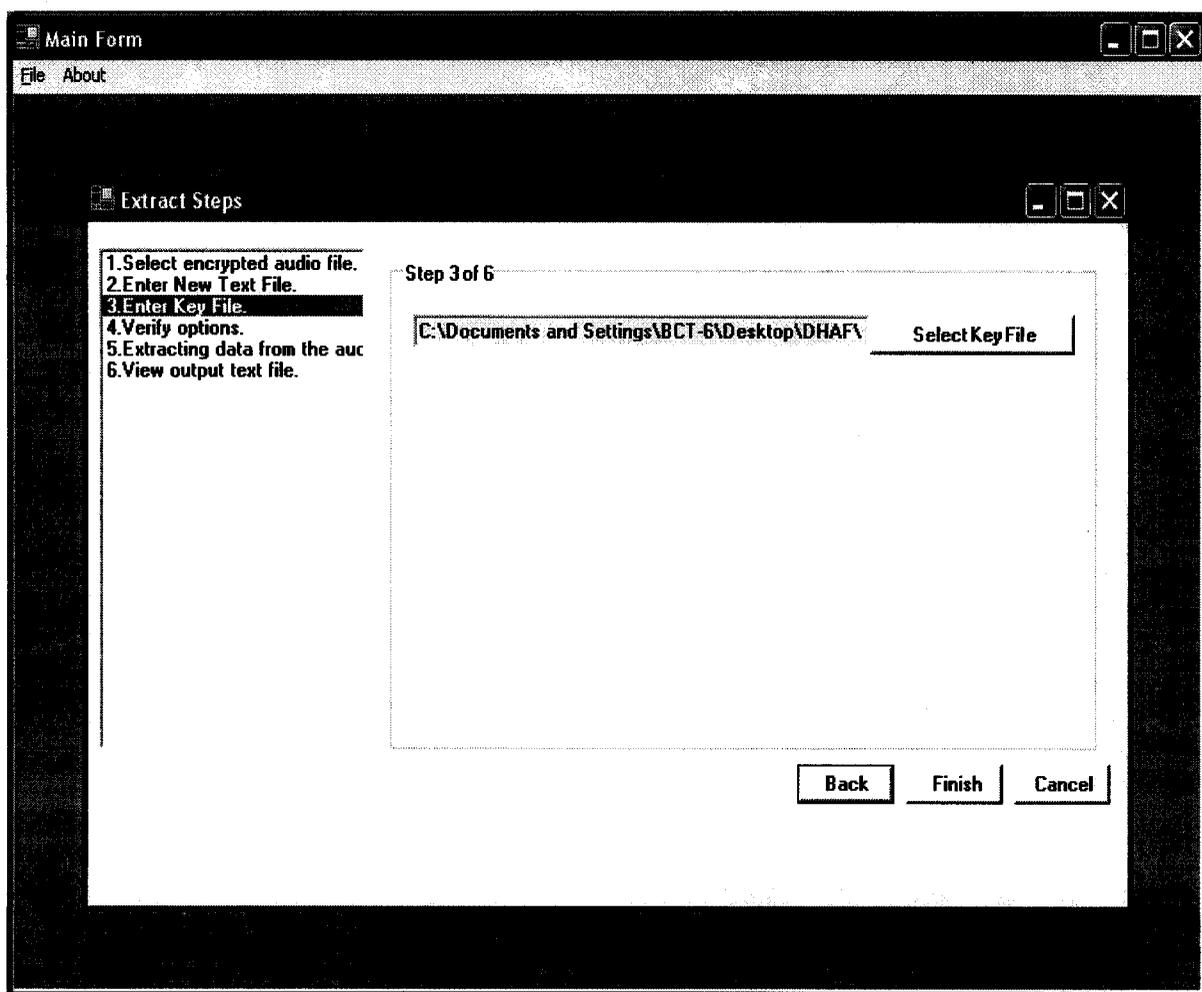
EXTRACTION MODULE:

Step 2: Selection of a new text file. This is the name of the file given by the receiver to save the embedded message.



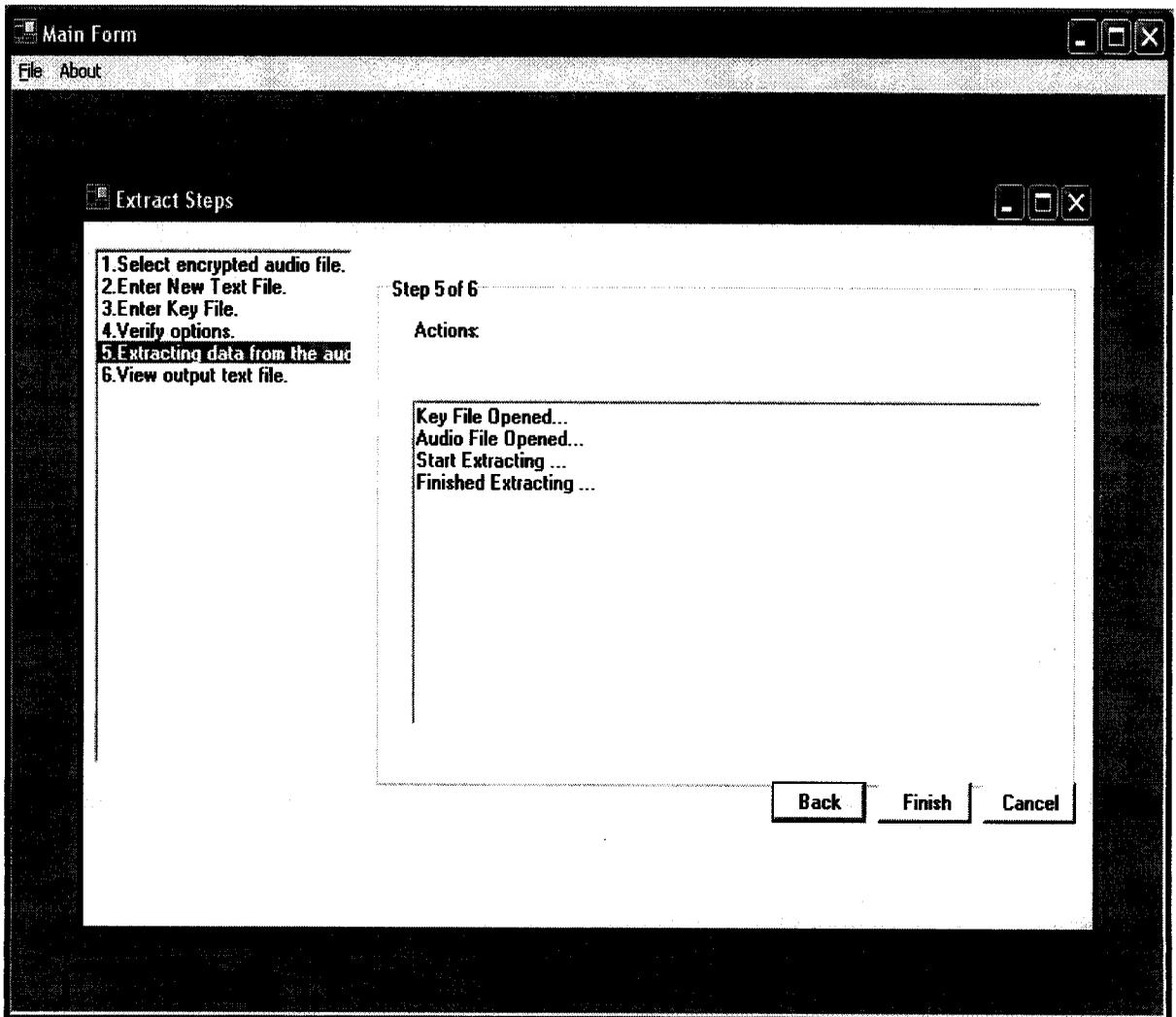
EXTRACTION MODULE:

Step 3: Selection of the Key file. This is the same file as selected in step 4 of the Embedding module.(Symmetric Encryption)



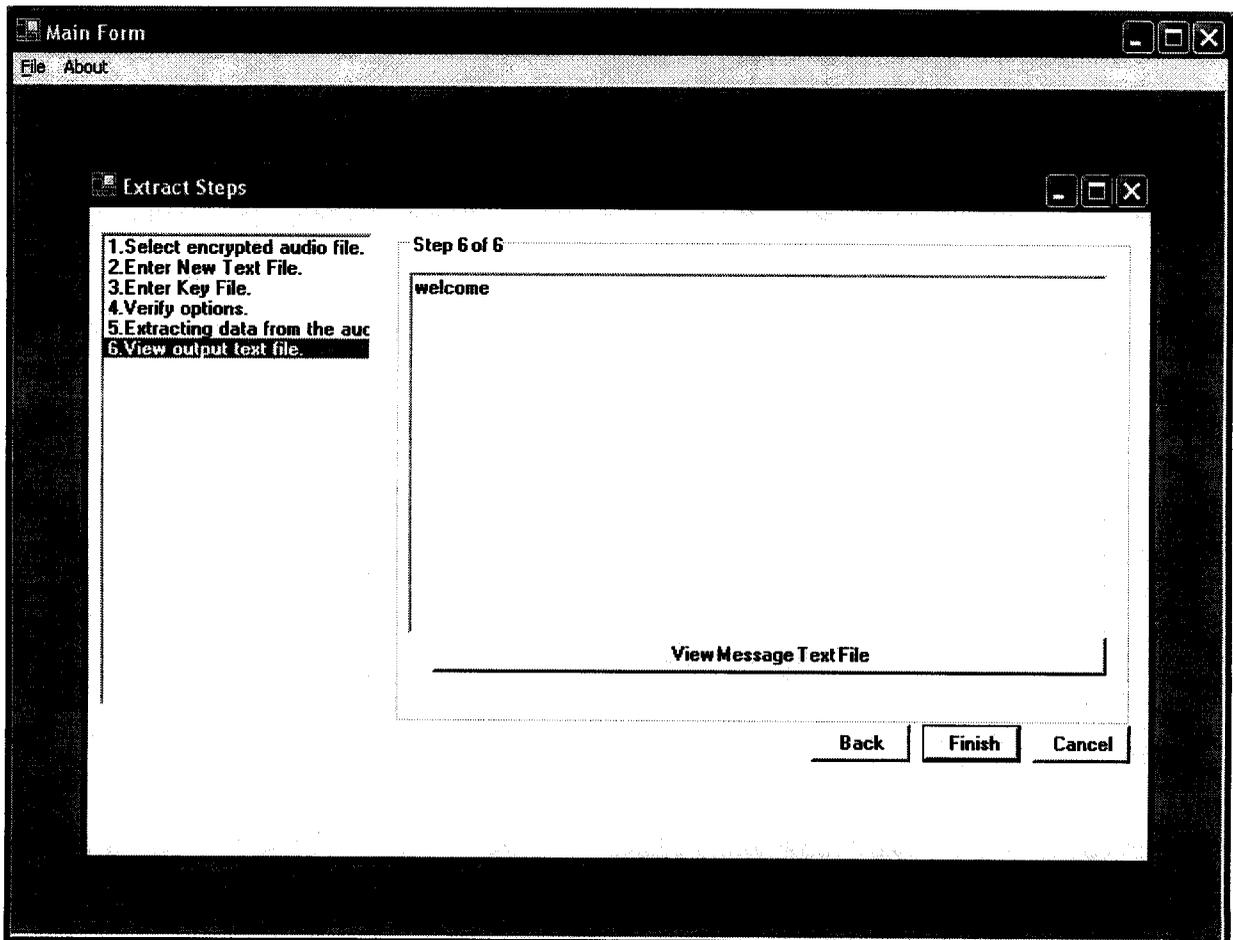
EXTRACTION MODULE:

Step 5: Extracting the data from the encrypted audio file.



EXTRACTION MODULE:

Step 6: Viewing the extracted message.



CHAPTER 9

REFERENCES:

Visual vb.net programming by Harold Davis.

Introduction to .net by Balaguruswamy.

www.garykessler.net/library/steganography.html

http://www.frontlinedefenders.org/manual/en/esecman/chapter2_8.html

<http://www.computerworld.com/securitytopics/security/story/0,10801,71726,00.html>

www.aes.org/events/113/papers/I.cfm

en.wikipedia.org/wiki/WAV

www.borg.com/~jglatt/tech/wave.htm