STEGANOGRAPHY IN AUDIO

PROJECT REPORT

Submitted by

P. SHIRANTHI MORAIS
M. SUDHA
D. SUJATHA

Guided by
Ms. N. Rajathi B.E.,

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING
OF BHARATHIAR UNIVERSITY
DURING THE ACADEMIC YEAR 2002-2003

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore – 641 006.

Estd. 1984

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore - 641006

Department of Computer Science and Engineering

ISO 9001:2000

CERTIFICATE

This is to certify that the project work entitled

'STEGANOGRAPHY IN AUDIO'

is a bonafide record of work done by

P. Shiranthi Morais
M. Sudha
D. Sujatha

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

Bharathiar University,Coimbatore

.....	.....
Head of the Department	Guide

Submitted for the University Examination held on 17.3.2003
and the University register number is 9927K0163, 9927K0165, 9927K0825

.....	.....
Internal Examiner	External Examiner

ACKNOWLEDGEMENT

Firstly I thank the Lord Almighty for His immense grace and blessings at each and every stage of this project.

I sincerely thank our Principal Dr.K.K.Padmanaban, Kumaraguru College of Technology, Coimbatore for the patronage and innumerable facilities provided by him for the project work.

I am profoundly grateful to Prof.S.Thangasamy, Head, Department of Computer Science, Kumaraguru College of Technology, Coimbatore, for his support and enthusiasm.

I am thankful to Ms.N.Rajathi, our guide and Senior Lecturer, Department of Information Technology, Kumaraguru College of Technology, Coimbatore, for her valuable guidance and constant monitoring throughout the course of my project.

I am also thankful to all the faculty members of Department of Computer Science, Kumaraguru College of Technology, for their cooperation and encouragement during my study period.

I would like to thank my parents for their love and moral support and also for their constant encouragement. I would also like to thank all my friends who gave me valuable tips and encouragement to complete this project successfully.

SYNOPSIS

Steganography is the practice of embedding secret messages in other messages in a way that prevents an observer from learning that anything unusual is taking place. Steganography simply takes one piece of information and hides it within another. Computer files (images, sounds recordings, even disks) contains unused or insignificant areas of data. Steganography takes advantage of these areas, replacing them with information. The files can then be exchanged without anyone knowing what really lies inside of them.

This project aims at attaining Steganography that provides operations to load an audio file such as wav or mp3 file, embed text into the audio file and try to recover the text from the file in which the text was embedded. The application is authenticated by means of password protection and cryptography which gives security to the system. This additional security is provided for the application because the security of the system has to be based on the assumption that the "Eaves Dropper" has full knowledge of the design and implementation details of the Steganographic system.

The text to be embedded is compressed, then encrypted and embedded in the unused bits of wav or mp3 file, by giving a password key. The file is sent to the receiver. The receiver decompresses the text, once again password is given so as to decrypt it to get the actual text.

This project is designed in such a way so as to incorporate any changes if required in the future.

CONTENTS

CONTENTS

INTRODUCTION

INTRODUCTION

CURRENT STATUS OF PROBLEM, TAKEN UP:

"The goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present".

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography is defined as the science or even possibly art of hiding likely critical information within other, seemingly benign information. To do this, unused, useless, redundant, and/or unnecessary bits of the original file are replaced with the its of the critical information.

The point of Steganography is to prevent the detection of information leakage. The secret message is encrypted first and embedded in the bit stream. However Steganography is frequently used in conjunction with cryptography. In this method, plain text would first be encrypted and then fed to the Steganography program. This ensures that if the hidden message is retrieved, the plain text will still be encrypted.

This project provides operations to load an audio file such as wav or mp3 file, embed text into the *unused bits* of audio file and try to recover the text from the file in which the text was embedded. The application is authenticated by means of password protection and cryptography which gives security to the system. This additional security is provided for the application because the security of the system has to be based on the assumption that the "Eaves Dropper" has full knowledge of the design and implementation details of the Steganographic system.

The text to be embedded is compressed, then encrypted and embedded in the unused bits of wav or mp3 file, by giving a password key. The file is sent to the receiver. The receiver decompresses the text, once again password is given so as to decrypt it to get the actual text. If the password is not correct, then the receiver cannot retrieve the actual message. Huffman Compression algorithm is used for the compression and decompression of data.

RELEVANCE AND IMPORTANCE:

As there is a need for lot of security required to maintain the confidentiality of the data, a new system of encryption of data has to be developed for this purpose. In Cryptography, where the enemy is allowed to detect the hidden message but in Steganography the person doesn't even know whether there is a secondary message present in it.

Steganography hides the information in some of the medias like text files, picture files, audio files, and video files. Steganography is done only in text and picture files such as bmp, jpeg.

The main goal of our project is to provide secure data exchange. We have proposed a solution to hide the information in the unused bits of audio file such as wav, mp3. It provides simple and secure process and ensures full performance. Even the currently available steganographic techniques can prone to security lapse. So we are providing a modicum of protection by enhancing with a strong cryptographic algorithm so that a high security can be provided to the message. And also the password key protection is provided.

LITERATURE SURVEY

LITERATURE SURVEY

Stganography:

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from greek, literally means, "covered writing". It includes a vast array of secret communication methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications.

Steganography is the practice of hiding information in computer pictures or music and relies on the fact that digital images and mp3 music files. Steganography and Cryptography are cousins in the spy craft family. Cryptography scrambles a message so it cannot be understood. Steganography hides the message so it cannot be seen. A message in cipher text, for instance might arouse suspicion on the part of the recipient while an "invisible" message created with Steganographic methods will not.

Ancient Methods of Concealing Data:

Throughout history, people have hidden information by a multitude of methods and variations. For example, ancient Greeks wrote text on wax covered tablets. To pass a hidden message, a person would scrape wax off a tablet, write a message on the underlying wood and again cover the tablet with wax to make it appear blank and unused. Another ingenious method was to shave the head of a messenger and tattoo a message or image on the messenger's head. After the hair grew back, the message would be undetected until the head was shaved again.

Invisible inks offered a common form of invisible writing. Early in World War II, Steganographic technology consisted almost exclusively of these inks. With invisible ink, a seemingly innocent letter could contain a very different message written between the lines. Documents themselves can hide information document text can conceal a hidden message through the use of null ciphers (unencrypted messages), which camouflage the real message in an innocent sounding missive. Open coded messages, which are plain text passages, "sound" innocent because they purport about ordinary occurrences. Because many open-coded messages don't seem to be cause for suspicion, and therefore "sound" normal and innocent, mail filters can detect the suspect communications while "innocent" messages are allowed to flow through.

Pitfalls in the existing system:

Steganography projects are existing only in text and image files. Existing systems don't have any authentication i.e. any user can read the message. We have proposed a solution to hide text information in unused bits of audio files (wav or mp3) – layer III model, which includes authentication methods such as password key protection and cryptographic key method. Researches are still going on in the field of Steganography.

SYSTEM REQUIREMENTS

SYSTEM REQUIREMENTS

Hardware Requirements:

CPU Type	-	Pentium III
Speed	-	650 MHz
RAM	-	64 MB
Cache Memory	-	256 KB
Hard Disk	-	20 GB
Floppy Disk	-	1.44 MB
Serial Ports	-	2
Parallel Ports	-	2

Software Requirements:

Operating System	:	Windows 98 / Windows XP
Front End	:	MICROSOFT VISUAL C++ 6.0

Software Requirements Specification:

Recent advances in computing power and recent interest in privacy has led to the development of techniques to hide messages in otherwise innocuous computer files such as digital pictures and digitized audio. These techniques are now referred to in the aggregate as Steganography. Using these techniques, it is possible to send a secret message to someone in the know and no one else will even know that the message is there.

Steganography, the art of embedding one file into another so as to communicate in a secret manner is a very old technique. The requirement of this software is to enable a text file to be embedded into an audio file. The text file of any size should be able to be embedded into any audio file such as wav, mp3.

The way of embedding of text into the audio file should be in such a way that there remains no explicit difference between the actual audio file and the tampered audio file. Steganography as such is by means secure but taking into consideration the modern era, every possible condition is to be taken into account.

Assuming that the hacker or the eaves dropper knows the existence to Steganography and its usage, novel methods are to be considered in order to make situations much better. Thus there arises a need for further security.

The additional security can be provided by means of password. The password used must be agreed between the two users in order to avoid confusion. This helps in providing additional security to the system.

SOLUTION STRATEGY

SOLUTION STRATEGY

Our System will hide information in wav or mp3 files during the compression process. The data is first compressed, encrypted and then hidden in the wav or mp3 bit stream. Any opponent can uncompress the bit stream and recompress it; this will delete the hidden information - actually this is the only attack we know yet, but at the expense of severe quality loss.

Two nested iteration loops are utilized in the Layer-3 encoder; the inner iteration loop (rate loop) and the outer iteration loop (noise control/distortion loop). These two loops are involved in a "negotiation" for digital real estate.

The hiding process takes place at the heart of the Layer III encoding process namely in the inner loop. The inner loop quantizes the input data and increases the quantizer step size until the quantized data can be coded with the available number of bits. Another loop checks that the distortions introduced by the quantization do not exceed the threshold defined by the psycho acoustic model.

The part2_3_length variable contains the number of main data bits used for scale factors and Huffman code data in the wav or mp3 bit stream. We encode the bits as its parity by changing the end loop condition of the inner loop. Only randomly chosen part2_3_length values are modified; the selection is done using a pseudo random bit generator based on SHA-1.

Implementation Summary:

Any message may be plain text, cipher text, or anything can be embedded in a bit stream. We provide operations to load an audio file such as wav or mp3 file, embed a text into the audio file and try to recover the text from the file in which the text was embedded. The application is authenticated by means of password protection and cryptography which gives security to the system.

The text to be embedded is compressed, then encrypted and embedded in the unused bits of wav or mp3 file, by giving a password key. The file is sent to the receiver. The receiver decompresses the text, once again password is given so as to decrypt it to get the actual text. Huffman Compression algorithm is used for the compression and decompression of the data.

Encryption:

we are providing a modicum of protection by enhancing with a strong cryptographic algorithm so that a high security can be provided to the message. Encryption algorithms refer to steps that a PC takes to transform plain text into cipher text. A key is a piece of information, usually a number that allows the sender to encode a message only for the receiver.

Need to encrypt data:

Transmission can have only a limited amount of physical security. So it is reasonable to assume that at some point someone may intercept your transmission. Whether we expect an interception or generally suspect that interceptions may occur, we should transmit information in a format that is useless to any interceptor. At the simplest level this

DESIGN DOCUMENT

Modules:

The two main modules of our project are

- Transmitting Module.
- Receiving Module.

Phases of Transmitting Module:

1. Compression of the text using Huffman Compression algorithm.
2. Encryption of the text using encryption key.
3. Embedding of the encrypted text into the audio file.

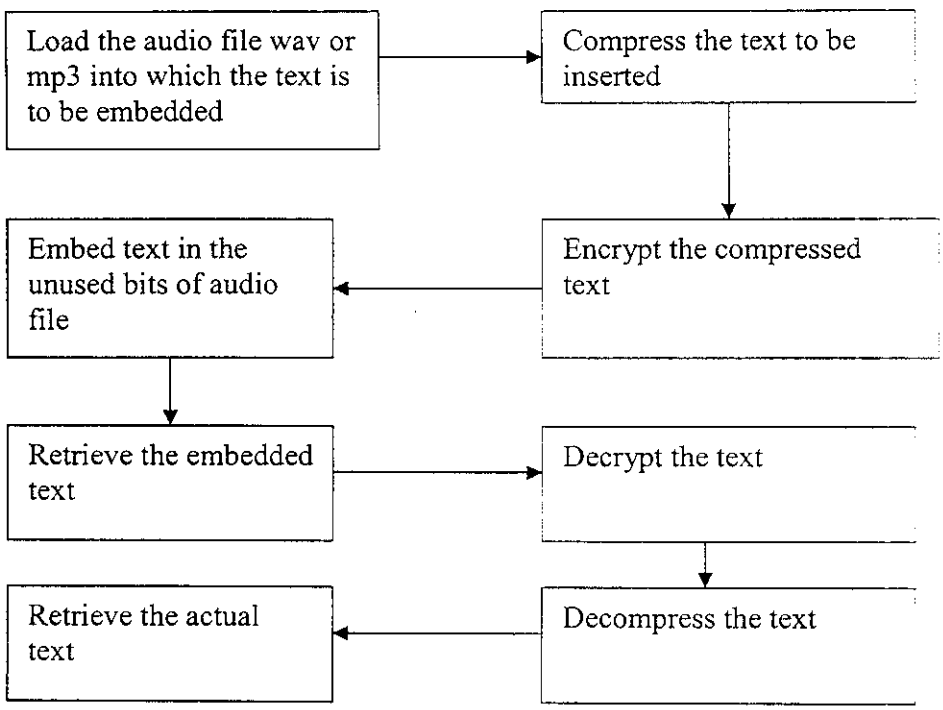
Phases of Receiving Module:

1. Retrieve the text from the audio file.
2. Decrypt the text using decryption key.
3. Decompression of the text.

Here we will attempt to use an algorithm to hide a message, encrypted with a key, in an audio file, so that it can only be decoded using the same key. We would be investigating the audio format wav or mp3 file.

The hiding process takes place at the heart of the Layer III encoding process namely in the inner loop. The inner loop quantizes the input data and increases the quantizer step size until the quantized data can be coded with the available number of bits. Another loop checks that the distortions introduced by the quantization do not exceed the threshold defined by the psycho acoustic model.

DATA FLOW DIAGRAM:



Once the text to be hidden is obtained from the user, it is compressed and then encrypted. While encrypting the text, a password (secret) key is obtained. Then the text is embedded into audio file. During retrieval the file is loaded, and the password should be given correctly to retrieve the text. But however the password doesn't match the user is not requested again to enter the password. But instead it is reported that the file contains some wrong text.

IMPLEMENTATION DETAILS

IMPLEMENTATION DETAILS

Why we have selected VC++?

We have chosen Visual C++ as our language for our project and Windows 98/XP platform. We could have done this project using turbo C or C++. But the thing is that the any good software package should allow the user to view the data in a graphical manner to allow better interpretation. For this purpose, we have to use GUI interface. Currently only Microsoft windows is the only user friendlier Operating System and the best compiler in Windows is Visual C++.

We could have also even coded in Visual Basic but the thing with Visual Basic is that it produces lot of bloatware. A typical hello world application will compile to around 23K in Visual C++ but would require around 1.3-1.4 MB with all the run time libraries in Visual Basic.

Many thousands of programmers have used Visual C++ to speed up areas of their lagging code.

Feature Overview

- Fast compact 32-bit DLLs and EXEs for Microsoft Windows 95/98/ME /NT/2000/XP.
- 32-bit protected mode code generation for maximum performance.
- Pointers and indexed pointers for direct memory access.
- 80-bit Extended-precision math.
- Register variables for increased performance: up to six unique register variables; Integer class (2) or floating-point (4).
- Unsigned Integer types: BYTE (8-bit), WORD (16-bit) and DWORD (32-bit).

- Signed Integer types: INTEGER (16-bit), LONG (32-bit) and QUAD (64-bit).
- Native null-terminated strings.
- User-defined TYPEs and UNIONs.
- Use of all available memory (up to 2GB) for arrays and dynamic strings.
- Optional requirement that variables be declared before use.
- Built-in 32-bit Inline Assembler with 80486 and Pentium opcodes
- Inline Assembler includes Floating-Point and MMX instructions.
- Direct export of Subs and Functions.
- Import Subs and Functions from the entire Win32 or any 32-bit DLL.
- Multi-thread application support: Create, Suspend, Resume, Status and Close.
- High-speed Serial Communications.
- Easy to use syntax highlighting Integrated Development Environment (IDE) and debugger.
- Code Compatibility – A program designed for Windows 95 can work with Windows NT too. The only hiccup being that we have to recompile.

File Organization

Visual C++ does not work in a single file. It works with a group of files. The entire group of files reside in a directory and is termed as the Workspace. The Workspace contains all the files required for the Project. This may be the resource files, the images which we use, header files and libraries which we need for our project. Our project consists of the following files.

(i) Resource Files:

A resource file may contain a collection of icons, menus, dialog boxes, string tables, user-defined binary data and other types of items.

TESTING

The implementation phase is less creative than system design. No software project is assumed completed until it is successfully tested and implemented.

An elaborate test data is prepared and the system is tested using test data. While testing errors are note and corrections are made. The corrections are also noted for future use. Both the hardware and software securities are made to run the developed system successfully in future.

Objectives of Testing:

- Testing is the process of executing a program with the indent of finding an error.
- A good test case is one that has high probability of finding yet undiscovered errors.
- A successful test is one that uncovers and yet undiscovered errors.

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and effectively before the live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

The candidate system is subjected to a variety of tests online response volume, stress, recovery, security and usability tests. A series of testing are performed before the system is read for the user acceptance testing.

FUTURE ENHANCEMENTS

Future enhancements to this project could be dealt as below:

- Picture file and even audio file can be embedded within audio file such as wav or mp3.
- Our project works with audio formats like wav & mp3. It can also be made to work using audio formats like RIFF etc.,
- Encryption with the password can be added, using modern encryption algorithm.
- The field of Steganography could be vastly used for military and defence purposes.

CONCLUSION

The best known application that implements Steganography in Audio, is authenticated using password protection and cryptographic key method. This project provides simple and secure process and ensures full performance. The implementation of this project was simple, making use of a user-friendly and GUI language, Visual C++ 6.0. The usage has been provided to be simple even allowing a person who is not familiar with Graphical Interfaces to work with this model.

REFERENCES

1. Neil F. Johnson and Sushil Jajodia

   "Exploring Steganography : Seeing the Unseen"

   IEEE Computer, 31, no 2:26-34, February 1998.

2. W. Bender, D. Gruthi, N. Morimoto and A. Lu

   "Techniques for data hiding"

   In *IBM Systems Journal, Vol. 35, Nos. 3-4*, pages 313-336, February 1996.

3. Peter Aiken, Scott Jarol

   "Visual C++ Multimedia"

   Adventure Set Edition 3.

4. David J. Kruglinski, George Shepherd, Scot Wingo

   "Programming Microsoft VC++"

   WP Publishers, 1998.

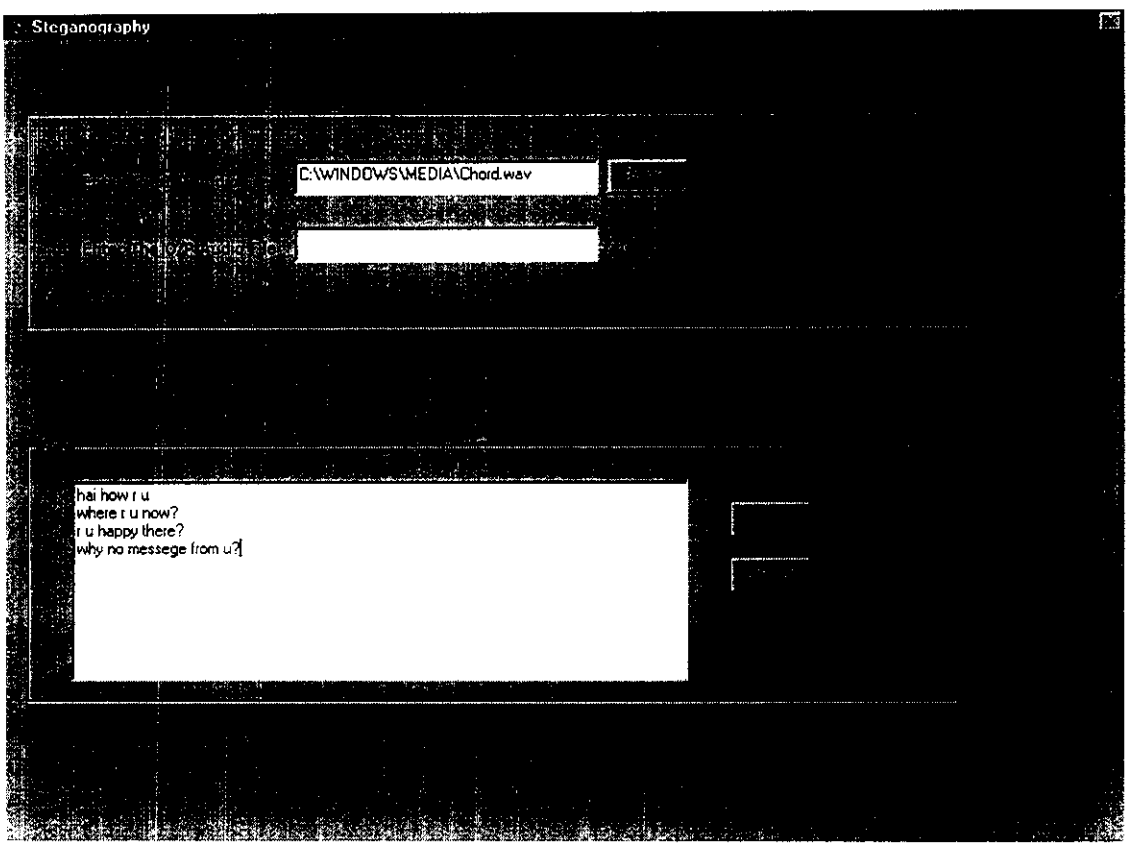
Online resources:

www. Steganography.com

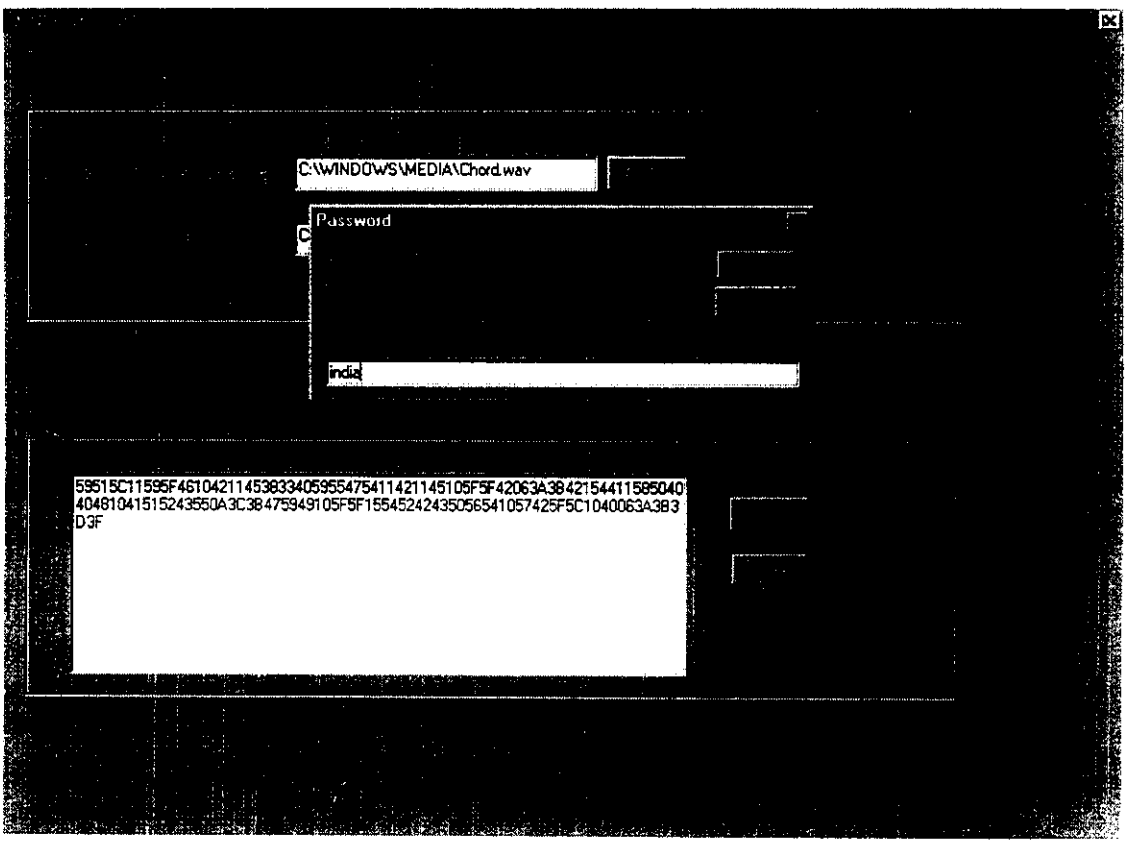
www.sci.crypt.research

www.StegoArchive.com

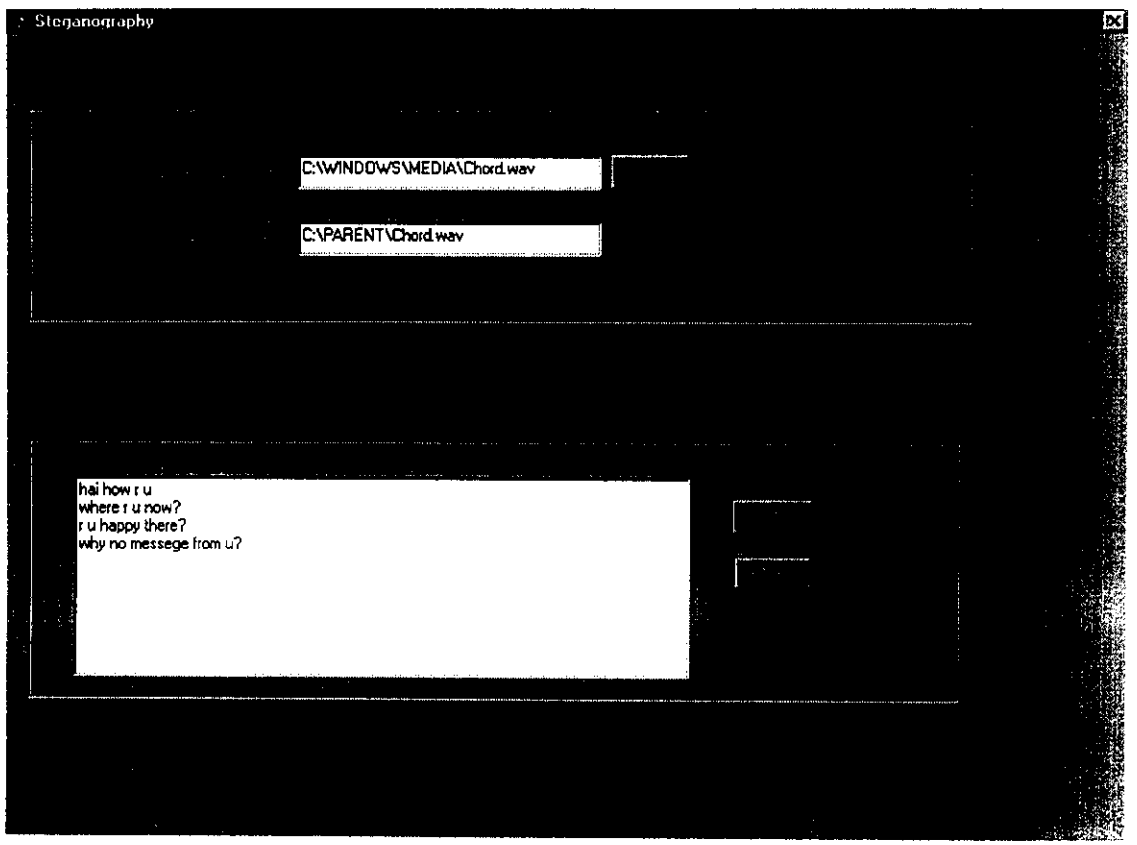
Entering the data to be hidden in the selected audio file:



Encoding and Embedding the compressed data using password key:



Retrieval of the hidden data from the selected audio file:



Audio File Properties: