ROBUSTNESS OF AUDIO STEGANOGRAPHY USING GENETIC ALGORITHM

By

MADHAN RAJ.K

Reg. No. 0920107009

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF ENGINEERING

IN

COMMUNICATION SYSTEMS

APRIL 2011

BONAFIDE CERTIFICATE

Certified that this project report entitled "**ROBUSTNESS OF AUDIO STEGANOGRAPHY USING GENETIC ALGORITHM**" is the bonafide work of Mr.Madhan Raj.K [Reg. no. 0920107009] 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 or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide

Ms.A.Amsaveni

Head of the Department

Dr. Rajeswari Mariappan

The candidate with university Register no. 0920107009 is examined by us in the project viva-voce examination held on ..21.-.04.-2011..

D. M 21/4/11

Internal Examiner

21/4/11

External Examiner

ACKNOWLEDGEMENT

First I would like to express my praise and gratitude to the Lord, who has showered his grace and blessing enabling me to complete this project in an excellent manner.

I express my sincere thanks to our beloved Director **Dr.J.Shanmugam**, Kumaraguru College of Technology, for his kind support and for providing necessary facilities to carry out the work.

I express my sincere thanks to our beloved Principal **Dr.S.Ramachandran**, Kumaraguru College of Technology, who encouraged me in each and every steps of the project work.

I would like to express my deep sense of gratitude to our HOD, the ever active **Dr.Rajeswari Mariappan**, Department of Electronics and Communication Engineering, for her valuable suggestions and encouragement which paved way for the successful completion of the project work.

In particular, I wish to thank with everlasting gratitude to the project coordinator **Ms.D.Mohanageetha(Ph.d).**, Associate Professor, Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success.

I am greatly privileged to express my heartfelt thanks to my project guide **Ms.A.Amsaveni(Ph.d),** Assistant Professor-SRG, Department of ECE, throughout the course of this project work and I wish to convey my deep sense of gratitude to all the teaching and non-teaching staffs of ECE Department for their help and cooperation.

Finally, I thank my parents and my family members for giving me the moral support and abundant blessings in all of my activities and my dear friends who helped me to endure my difficult times with their unfailing support and warm wishes.

ABSTRACT

Steganography is a technique used to transmit hidden information by modifying a stego signal in an imperceptible manner. Due to the rapid development in the multimedia data available in digital format (image, audio, video) has opened many challenges and opportunities for innovation. The multimedia data like image, audio can be used as a stego signal for performing steganography. Since the availability of various image steganalysis technique, image steganography become less secure. So in order to make the communication secure, the audio signal can be used as a stego signal. In this technique the secret data bits are embedded in the audio signal. The secret data bits are embedded in the vague and deeper layers of the audio signal using substitution technique. The substitution technique uses an intelligent algorithm called genetic algorithm to embed the secret data bits. The genetic algorithm uses crossover, mutation and fitness function to choose the bit location for embedding the secret data bit. This technique improves the robustness of the stego signal and makes difficult for the attacker to retrieve the secret data by performing steganalysis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DCT	-------	**Discrete Cosine Transform**
DSSS	-------	**Direct Sequence Spread Spectrum**
DWT	-------	**Discrete Wavelet Transformation**
FHSS	-------	**Frequency Hopping Spread Spectrum**
HAS	-------	**Human Auditory System**
HVS	-------	**Human Visual System**
MATLAB	-------	**MATrix LABoratory**
PSNR	-------	**Peak Signal to Noise Ratio**
PN	-------	**Pseudorandom Noise**
PBIS	-------	**Pattern-Based Image Steganography**
SNR	-------	**Signal to Noise Ratio**
WAV	-------	**Wave File**

CHAPTER 1
INTRODUCTION

Nowadays the Internet as a whole does not use secure links, thus information transit may be vulnerable to interception as well. The important of reducing a chance of the information being detected during the transmission is being an issue now days. So we have to develop a technique that is capable of matching the security requirements.

1.1 MOTIVATION

The rapid development of the Internet and the digital information revolution caused significant changes in the global society, ranging from the influence on the world economy to the way people nowadays communicate. Since multimedia data available in a digital format (text, images, audio, video), opened many challenges and opportunities for innovation. This provides opportunities for the hackers to hack the information and the hacker is capable of destroy the retrieved information or modifying the retrieved information. This makes the people to think of the security aspects of the information that is being transmitted over a network. So with the help of technological development, the information security requirements should be maintained.

1.2 PROJECT GOAL

The goal of this project is to design an algorithm for embedding secret information(text data) into the audio file(wav) and decode the secret information from the modified audio file(stego file) at the destination without degrading the quality of the audio file that has been used for embedding the secret information. The algorithm developed for embedding the secret information into the audio file should satisfy the following properties:

- ✓ Transparency
- ✓ Robustness
- ✓ Capacity

1.3 OVERVIEW

The basic steps for embedding secret data into audio file have been shown in the Figure 1.1

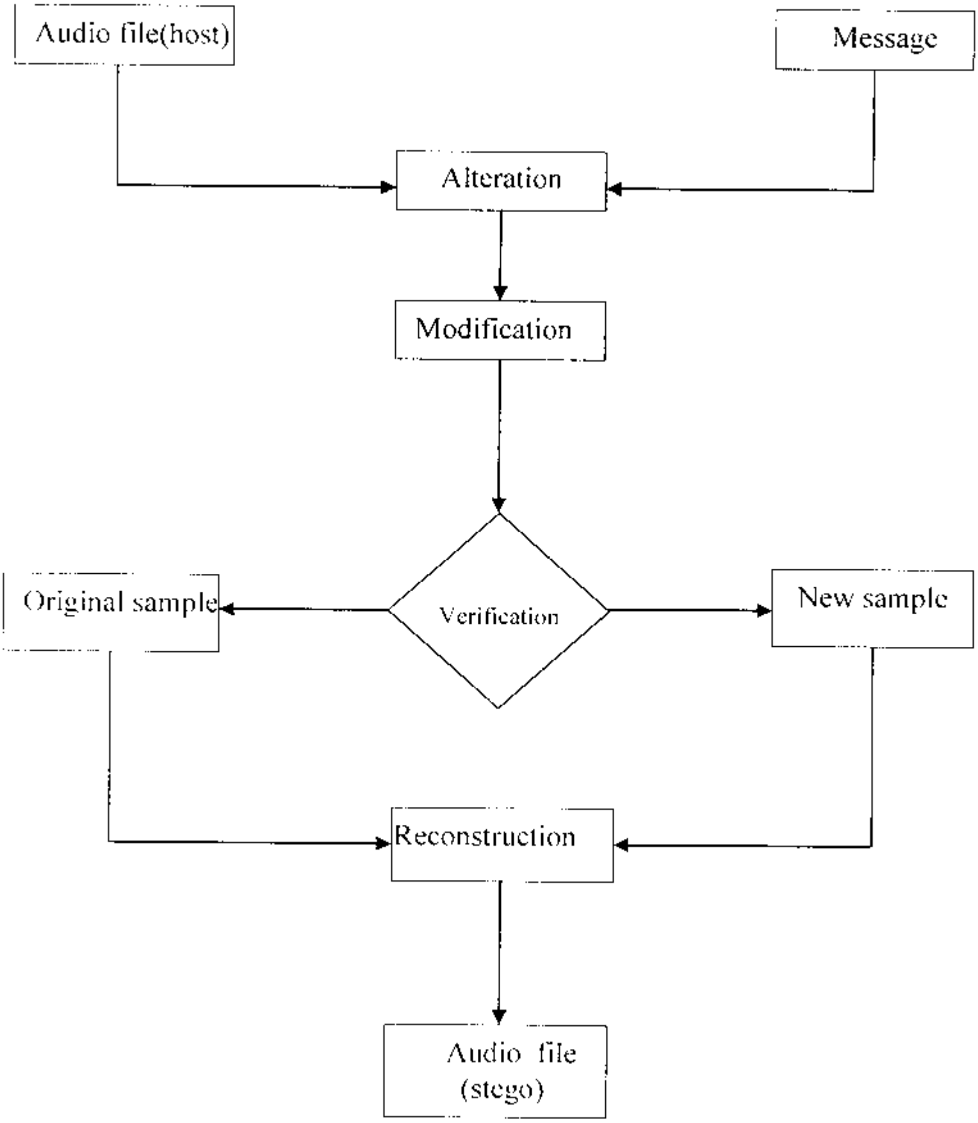


Figure 1.1 Overview of the Design Steps of the Project

1.4 SOFTWARES USED

- ✓ Matlab R2008b

1.5 ORGANIZATION OF THE REPORT

- ✓ **Chapter 2** Introduction To Steganography
- ✓ **Chapter 3** Literature Review
- ✓ **Chapter 4** Steganography Methods & Techniques
- ✓ **Chapter 5** Genetic Algorithm
- ✓ **Chapter 6** Analysis Of Audio Steganography Using Genetic Algorithm
- ✓ **Chapter 7** Simulation Results
- ✓ **Chapter 8** Conclusion & Future Work
- ✓ **Chapter 9** Bibliography

CHAPTER 2

INTRODUCTION TO STEGANOGRAPHY

2.1 INTRODUCTION

Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless messages. Steganography means "covered writing" in Greek. As the goal of steganography is to hide the *presence* of a message and to create a covert channel, it can be seen as the complement of cryptography, whose goal is to hide the *content* of a message. Steganography is often confused with cryptology because the two are similar in the way that they both are used to protect important information.

The difference between the two is that Steganography involves hiding information so it appears that no information is hidden at all. If a person or persons views the object that the information is hidden inside of he or she will have no idea that there is any hidden information, therefore the person will not attempt to decrypt the information. Steganography in the modern day sense of the word usually refers to information or a file that has been concealed inside a digital Picture, Video or Audio file. What Steganography essentially does is to exploit human perception; human senses are not trained to look for files that have information hidden inside of them, although there are programs available that can do what is called Steganalysis (Detecting use of Steganography.) The most common use of Steganography is to hide a file inside another file. When information or a file is hidden inside a carrier file, the data is usually encrypted with a password and shown in figure 2.1

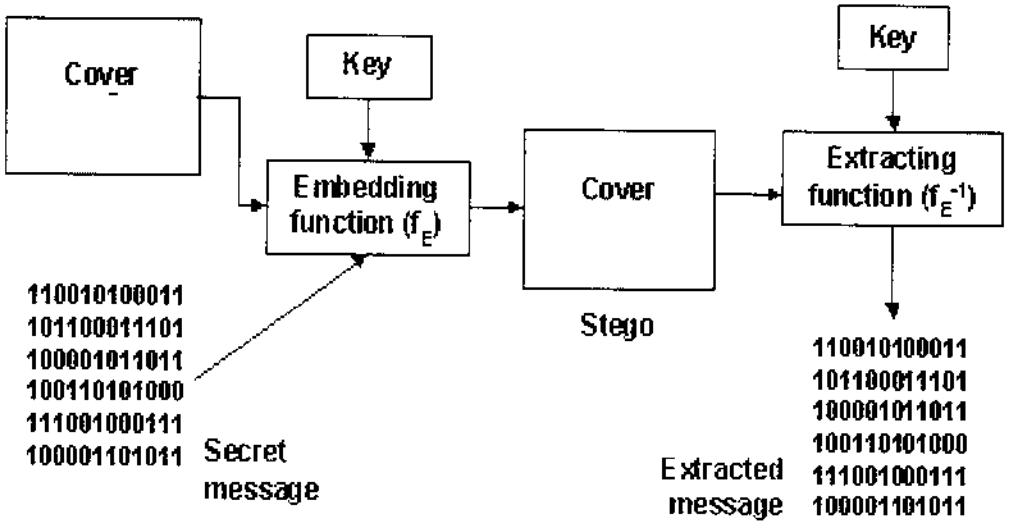


Figure 2.1 Steganography System

2.2 STEGANOGRAPHY TERMS

Carrier File – A file which has hidden information inside of it.

Steganalysis – The process of detecting hidden information inside of a file.

Stego-Medium – The medium in which the information is hidden.

Redundant Bits – Pieces of information inside a file which can be overwritten or altered without damaging the file.

Payload – The information which is to be concealed.

Some examples of use of Steganography in past times are:

1. During World War 2 invisible ink was used to write information on pieces of paper so that the paper appeared to the average person as just being blank pieces of paper. Liquids such as urine, milk, vinegar and fruit juices were used, because when each one of these substances are heated they darken and become visible to the human eye.

2. In Ancient Greece they used to select messengers and shave their head, they would then write a message on their head. Once the message had been written the hair was allowed to grow back. After the hair grew back the messenger was sent to deliver the message, the recipient would shave off the messengers hair to see the secret message.

3. Another method used in Greece was where someone would peel wax off a tablet that was covered in wax, write a message underneath the wax then re-apply the wax. The recipient of the message would simply remove the wax from the tablet to view the message.

2.3 STEGANOGRAPHY VS CRYPTOGRAPHY

Basically, the purpose of cryptography and steganography is to provide secret communication. However, steganography is not the same as cryptography. Cryptography hides the contents of a secret message from a malicious people, whereas steganography even conceals the existence of the message. Steganography must not be confused with cryptography, where we transform the message so as to make it meaning obscure to a malicious people who intercept it. Therefore, the definition of breaking the system is different. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganographic system need the attacker to detect that steganography has been used and he is able to read the embedded message.

In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. It makes no attempt to disguise or hide the encoded message. Basically, cryptography offers the ability of transmitting information between persons in a way that prevents a third party from reading it. Cryptography can also provide authentication for verifying the identity of someone or something.

In contrast, steganography does not alter the structure of the secret message, but hides it inside a *cover-image* 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. In other word, steganography prevents an unintended recipient from suspecting that the data exists. In addition, the security of classical steganography system relies on secrecy of the data encoding system. Once the encoding system is known, the steganography system is defeated.

It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. The resulting *stego-image* can be transmitted without revealing that secret information is being exchanged. Furthermore, even if an attacker were to defeat the steganographic technique and detect the message from the *stego-object*, he would still require the cryptographic decoding key to decipher the encrypted message.

2.4 HISTORY OF STEGANOGRAPHY

The history of steganography falls into three ages of steganography:

- ✓ Old Age of Steganography
- ✓ Middle Age of Steganography
- ✓ Modern Age of Steganography

2.4.1 Old Age of Steganography

Since man first started communicating over written messages, the need for secrecy was in high demand. In the past, messages could easily be intercepted and since there were no secrecy devices, the third party was able to read the message. This all changed during the time of the Greeks, around 500 B.C., when Demaratus first used the technique of Steganography. Steganography is the use of hiding a message so it looks like a message does not exist at all. The official definition, according to Dictionary.com, is hiding a secret message within a larger one in such a way that others cannot discern the presence of contents of the hidden message. The word Steganography derived from two Greek words steganos, meaning "covered," and graphein, meaning "to write.

Demaratus was a Greek citizen who lived in Persia because he was banished from Greece. While in Persia, he witnessed Xerxes, the leader of the Persians, build one of the greatest naval fleets the world has ever known. Xerxes was going to use this fleet to attack Greece in a surprise attack. Demaratus still felt a love for his homeland and so decided he should warn Greece about the secret attack. He knew it would be hard to send the message over to Greece without it being intercepted. This is when he came up with the idea of using a wax tablet to hide his message. Demaratus knew that blank wax tablets could be sent to Greece without anyone being the wiser.

To hide his message, he scraped all the wax away from the tablet leaving only the wood from underneath. He then scraped his message into the wood and when he finished, recovered the wood with the wax. The wax covered his message and it looked like it was just a blank wax tablet. Demaratus' message was hidden and so he sent this to Greece. The hidden message was never discovered by the Persians and successfully made it to Greece. Because of this message, Greece was able to defeat the invading Persian force.

This was the first known case of the use of Steganography and since then, the complexity of Steganography has exploded. The reason for this is because it was forced to in order to keep the secrecy of the message intact. If you think about it, even though the message is hidden, in order to read it,

all you have to do is find it. If the message was intercepted and the person looked in the right place, the message could be easily read.

Soon after Demaratus' successful use of steganography, other forms started showing up. This also included the use of cryptology. The first known case of the use of cryptology actually came before the first known case of steganography. The use of steganography involves the coder to be a little more creative than that of a cryptology. Even though the focus of this paper is Steganography, it is impossible not to mention the history without mentioning cryptology. The two are like cousins in the secrecy family. With, cryptology, you can actually see that a message is there, but it was been scrambled so that hopefully a third party cannot read what the real message has to say. Cryptology has advanced just as much as Steganography and with the modern computers, cryptology has grown even more complex.

Another form of steganography was employed by another Greek, Histaiaeus, soon after the wax technique was no longer usable. Histaiaeus wanted to start an uprising against the Persian King and needed a form of secrecy to hide his message about the revolt. This is when he came up with the shaved head technique. Histaiaeus decided to shave the head of one of his slaves and tattooed the message on his bald scalp. When the hair grew back, he sent the slave to Persia to deliver the message. When the slave reached his destination, he shaved his head and showed the message to the intended recipient. There were a few problems with this technique though. The message could not be urgent for the fact that it takes time for the hair to grow back. Secondly, the message could not be deleted unless the messenger was killed. Which brings up the third point, the messenger could only deliver one message because tattoos cannot be erased.

The Ancient Chinese soon developed their own uses of steganography. They would write their secret message on a piece of fine silk. Once the message was complete, they would crunch the silk into a little ball. They would then cover the ball with wax and then the messenger would swallow the ball. There are quite a couple of sources that mention this technique that the Chinese used, but no one mentions how the ball of wax was recovered from the messenger. One is only left to imagine the possible, and probably painful, ways the ball could have come out.

Around 100 A.D. transparent inks made their way into the secrecy world. Pliny the Elder discovered that the "milk" of the thithymallus plant could easily be used as a transparent ink. If you wrote a message with the milk, it soon evaporated and left virtually no residue. It appeared as if the message completely erased. But once the milk completely dried, and was heated, it would begin to char and turn a brown color. So, this message could be written on anything that was not too flammable, which made it quite convenient. The reason it turned brown was because the milk was loaded with carbon and when carbon is heated, it tends to char.

Another form of early steganography was developed in the sixteenth century by an Italian scientist Giovanni Porta. His technique involved hiding the secret message inside a hard-boiled egg. Porta discovered that there was a way of writing on a hard-boiled egg so that the ink would become transparent on the outside of the egg but still be visible on the egg white itself. His ink was made by mixing one ounce of alum and a pint of vinegar. The ink would seep through the egg shell because of the porous nature of the shell. It would then alter the color of the egg white inside.

Due to the nature of the solution, when completely dried, turned transparent but not before altering the egg white. Because there was no notable difference on the outside of the shell, the only way to read the message would be to actually take the shell off the egg. This technique actually worked for a long time because no one thought it could be possible to write on the inside of the egg without disturbing the egg shell. The one problem with this technique is the fact that hard boiled eggs tend to rot quit quickly. The egg would have to be delivered almost immediately after the message disappeared on the outside. Not only would the rotting egg affect the visibility of the message, but the smell of the rotting egg would probably become too strong to bear.

2.4.2 Middle Age of Steganography

Invisible ink was also used during the American Revolutionary war. The technique used during this war was quite simple and easily noticeable. The technique the generals and solders used was one where they would start with a simple letter. The letter itself had no real meaning. In between all the lines of letters are gaps, empty spaces. This is to distinguish one line from the next. This is the place they used to hide the secret message using the invisible ink. They would write the message in the gaps with the invisible ink and then when the ink dried, it looked like a normal letter again. The downfall of this system was the way in which to retrieve the hidden message. In order to read the message, all you had to do was hold it up to a light and the ink glowed. So, if the enemy intercepted the letter and just so happened to read the letter with a light behind it. would easily see the secret message. There is no report on just how successful this technique really was during that war.

Another wartime technique used was the grille system. This technique involved strategically placing letters within a seemingly ordinary text. The secret message was sent and then the receiver was only able to see the secret message by using a special grille. The grille was just a slab of wood that would fit over the message. The slab had holes in it at the spots where the strategically placed letters would be. The letters would then spell out the secret message. This technique was effective due to the fact that the person trying to intercept the message would not be able to decode it unless they had the correct slab. This was also one of the major downfalls. Both parties needed to agree on the type of slab to use. To make it more secure, I am sure that one grille was not used very often. Not only did the parties have to agree on the right grille, but if the receiver's grille was lost or broken, the message again would be unreadable and thus useless.

World War II brought about the invention of two new steganographic techniques. The first one was the invention of the Microdot technology. The Microdot technology was invented by the Germans to convey secret messages to its allies. FBI director at the time, J. Edgar Hoover, referred to the microdot as "the enemy's masterpiece of espionage." He stated this after they realized just how much the Germans were using this technology and just how much information the microdot could hold. The microdot is basically a highly detailed picture shrunk to about the size of a period or dot. You were able to fit an entire page of text or an entire picture into the little microdot, which is what made the microdot so successful. You could then see the picture again by either blowing up the dot or just by simply using a microscope. The Germans would put their dots into their letters and they were almost undetectable to the naked eye. What gave them away was the fact that they were glossy, due to the fact that they were still on film. They stood out from the rest of the paper. which was very dull.

The other technique perfected during World War II was the use of open coded messages. This technique is very similar to the grille technique but this time no grille is needed to see the special letters. For open coded messages, certain letters of each word are used to spell out the secret message. Open coded messages use normal words and messages to write the buffer text that hides the message. Because they seem normal, they often pass security checks. Here is a frequently used example of open coded messages. Apparently this message was actually sent by a German Spy during World War II.

> **Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.**

By taking the second letter of each word, the secret message is revealed, which is: **Pershing sails from NY June 1**. This technique is effect because it can pass through filter devices but easily decoded by someone if they new a message was hidden in this text. It would not take long using the Brute Force technique to find the hidden message.

All the techniques mentioned earlier were good ways of hiding messages but even with the new twists given to them, they were still nothing compared to the types of applications developed with the invention of the Computer. Computer technology has made it all so much easier to hide messages and made it a lot harder to discover that message. The rest of the paper will focus on the new applications that have derived from the use of the computer.

2.4.3 Modern Age of Steganography

A majority of the messages hidden today are hidden inside digital images, audio files or video files. But even modern printers can hide messages with the way they print the text out. For example, laser printers are so precise; they can offset a letter by 1/300th of an inch. By doing this at certain points, they could send a binary message, which would be undetectable to the naked eye. They way it works is that a normal space would be considered a "0" while spaces that are offset by 1/300th of an inch would be considered the "1's". This is good for hiding messages in print form, but has not solved the problem of sending the hidden message from computer to computer. This is where the files come into play.

Hidden files or pictures can be hidden in picture files because pictures files are so complex. Pictures on a computer are represented by tons and tons of pixels. Each pixel consists of a variation of all three primary colors, red, green and blue. In a standard 24-bit bitmap, 8 bits will represent each of the three colors. 8 times 3 is 24. That means there are 256 different variations of each color in every pixel that makes up a picture. So, to represent the color white, the code would look like 11111111 11111111 11111111. Now, the human eye cannot distinguish the difference between too many colors and so the color 11111110

11111110 11111110 would look exactly the same as white. Because of this, the last digit in every bit in every pixel could be changed. This is the basis of the Least Significant Bit Insertion technique.

Now to show how this becomes useful. You only need 8 bits to represent Ascii text and there are three extra in every pixel of a picture. Therefore, with every three pixels, you could form one letter of ASCII text. This may not seem like a lot, but when the standard image size is 640 x 480 pixels, that add up to a lot in a hurry. In order to make this practical to the user, a computer program would be needed. After you type in your secret message and determine a cover message (the picture you want to hide you message in) the program would go through every pixel and change the last digit to represent each letter of the message you wrote. You would then send the picture to the correct recipient who would then use his program to go through every pixel and take off the last digit and use that to form the message.

The problem of using steganography over digital communications has been solved. Also, the great thing about LSB (Least Significant Bit Insertion) is that the message is not lost if the file is compressed. Anyone who uses online pictures knows that bitmap files hold a lot of information and so are generally large in size. But because the secret message is encoded into the color bits, the message is never lost when compressed. The one problem with this approach is that it does not work for every picture type. LSB works mainly with Bitmaps because of the way bitmaps are compressed. JPEG's, on the other hand, are compressed using sophisticated algorithms and so a lot of the original information is lost.

Because information could so easily be lost with certain compression programs, other techniques were developed. One technique is called the Masking and Filtering technique. This technique is very similar to watermarking. The image is marked with the secret message or image and then cannot be seen unless the luminosity level is changed to an exact amount. This worked better because the text/image was now actually part of the picture and no longer in the coding part. Another technique developed used the way certain pictures are compressed to its advantage. As stated earlier, JPEG's are compressed using sophisticated algorithms and because of this, a lot of the original information of the picture is lost. So, basically, what this last technique does is, it determines how the picture is going to be compressed with all the algorithms. It then changes the information of the picture accordingly to the secret message. It changes the information in a way that when decompressed, it will look similar to the LSB approach. This way, when the picture is viewed, it still looks the same but the secret message could be determined by taking the last bit of each pixel just like the LSB approach.

Today, the Internet is filled with tons of programs that uses steganography to hide secret messages. A majority of the programs use a variation of the algorithm approach. When looking closely at a program to determine how it works, you soon discover that it is really complicated, or at least, seems to be. But in reality, they are just the using the algorithm approach plus a few minor twists. Steganography is in such wide use today that it has been reported that even the terrorist group Al Quida uses it to deliver messages. Apparently they were using nude pictures on the Internet to hide their messages. I have not been able to find an example of this though.

Steganography is also being used everyday life for practical needs. Odds are, you encounter the use at least once a week and do not even know it. One of the biggest uses today are with copyrighted materials like DVDs. DVDs are actually encoded with certain watermarks that the DVD player recognizes. The watermark has numerous functions. First it tells where the DVD came from so if someone makes copies of their DVDs, the original copy could always be determined. Secondly, the watermark determines if the DVD could actually be copied or not.

Finally, the watermark tells the DVD player if it could play the DVD or not. Unknown to a lot of people, but DVDs are made in certain "Regions" and they only work in that region. For example, Asia and North America are considered different regions and so a DVD from Asia will not play on a DVD player that has a North America region code.

Seeing how complex steganography is today, it is hard to imagine what the future could hold. But with the way technology is growing exponentially, the bounds for steganography seem limitless. One day, hiding a message inside someone's brain without the person even knowing it, Johnny Mnemonic style, may become a reality.

CHAPTER 3

LITERATURE REVIEW

Mohammad Pooyan, Ahmad Delforouzi : In this paper we present a novel method for digital audio steganography where encrypted covert data is embedded into the wavelet coefficients of host audio signal. To avoid extraction error we use lifting wavelet LSB transform. For using the maximum capacity of audio signals, we calculate hearing threshold in wavelet domain. Then according to this threshold data bits are embedded in the least significant bits of lifting wavelet coefficients. Inverse lifting wavelet transform is applied to modified coefficients to construct stego signal in time domain. Experimental results show that proposed method has large payload, high audio quality and full recovery.

Nedeljko Cvejic, Tapio Seppben MediaTeam Oulu: Conventionally, a perceptual limit of three hits per sample is imposed to the basic LSB audio steganography method. In this paper, we present a novel modification to standard LSB algorithm that is able to embed four bits per sample, thus improving the capacity of data hiding channel by 33%. The proposed algorithm makes use of minimum error replacement method for LSB adjustment and modified error diffusion method for decreasing *SNR* value. Objective test showed the algorithm succeeds in this task, while keeping SNR value close to the level of SNR obtained by standard LSB embedding with three bits per sample capacity. Subjective listening test proved that high perceptual transparency is accomplished even if four LSBs of host audio signal are used for data hiding.

Mitchell D. Swanson, Ahmed H. Tewfik: In this paper, the recent developments in transparent data embedding and watermarking for audio, image, and video were reviewed. Data-embedding and watermarking algorithms embed text, binary streams, audio, image, or video in a host audio, image, or video signal. The embedded data are perceptually inaudible or invisible to maintain the quality of the source data. The embedded data can add features to the host multimedia signal. e.g., multilingual soundtracks in a movie, or provide copyright protection. The reliability of data-embedding procedures and their ability to deliver new services such as viewing a movie in a given rated version from a single multicast stream were discussed. We also discuss the issues and problems associated with copy and copyright protections and assess the viability of current watermarking algorithms as a means for protecting copyrighted data.

Xin Li and Hong Heather Yu: In this paper, we propose a novel data hiding scheme for audio signals in cepstrum domain. Cepstrum representation of audio can be shown to be very robust to a wide range of attacks including most challenging time-scaling and pitch-shifting warping. In cepstrum domain, we propose to embed data by manipulating statistical mean of selected cepstrum coefficients. An intuitive psychoacoustic model is employed to control the audibility of introduced distortion. Our experiment results have shown that the novel audio data hiding scheme in cepstrum domain can achieve transparent and robust data hiding at the capacity region of above 2Obps.

Sang-Kwang Lee and Yo-Sung Ho: In this paper, we propose a: digital audio watermarking technique in the cepstrum domain. We insert a digital watermark into the cepstral components of the audio signal using a technique analogous to spread spectrum communications, hiding a narrow hand signal in a wideband channel. In our method, we use pseudo-random sequences to watermark the audio signal. The watermark is then weighted in the cepstrum domain according to the distribution of cepstral coefficients and the frequency masking characteristics of human auditory system. Watermark embedding minimizes the audibility of the watermark signal. The embedded watermark is robust to multiple watermarks. MPEG: audio coding and additive noise.

Chin-Chen Chang *Tung-Shou Chen Hsien-Chu Hsia: In this paper, a new scheme for image steganography will be presented. The proposed scheme has the ability to hide secret message in a digital image. We call the scheme the pattern-based image steganography (PBIS). First, PBIS does discrete wavelet transformation (DWT) on the digital image; separates the transformation result into non overlapping blocks, and classifies the wavelet coefficients of these blocks into several patterns. The secret message is embedded in to the image by changing the coefficients' patterns. The performance and reliability of PBIS are shown in the experimental results. The quality of the stego image of the proposed scheme is very close to that of the original one. Moreover, our scheme can be survived under JPEG lossy compression.

Kaliappan Gopalan: A method of embedding information in the cepstral domain of a cover audio signal is described for audio steganography application. The proposed technique combines the commonly employed psycho acoustical masking property of human auditory system with the decorrelation property of speech cepstrum, and achieves imperceptible embedding, large payload, and accurate data retrieval. Results of embedding using a clean and a noisy hot utterance show the embedded information is robust to additive noise and band pass filtering.

Sos S. Agaian, David Akopian, Okan Caglayan and Sunil A. D'Souza: This paper presents a lossless adaptive digital audio steganographic technique based on reversible two and higher dimensional integer transform. The adaptive technique is used to choose the best blocks for embedding perceptually inaudible stego information, and to select the best block sizes to maximize the number of blocks/capacity. The stego information is embedded in the integer domain by bit manipulation. In addition, we introduce a capacity measure to select audio carriers that introduce minimum distortion after embedding. The above technique is also applicable to compression based audio formats, such as MPEG audio layer-3 (mp3).

Hosei Matsuoka: This paper presents an improvement of spread spectrum audio data hiding methods. We introduce phase shifting in audio signals to reduce the correlation with PN signal per each sub-band. It allows easy detection of the embedded data signal from audio when de-spreading the compound signal. The paper reports the subjective test results and the measurements of noise resiliency. The proposed method generates the quality degradation at the same level of NMR +3dB, but excess +6dB noise, therefore, the method has 3dB benefits.

Sajad Shirali-Shahreza ,M. T. Manzuri-Shalmani: Embedding a secret message into a cover media without attracting attention, which is known as steganography, is desirable in some security applications. One of the medias which can be used as a cover media is audio signal. In this paper we introduce an adaptive wavelet domain steganography with high capacity and low error rate. We use lifting scheme to create perfect reconstruction filter banks which are Int2Int and hide data in Least Significant Bits (LSB) of details coefficient in an adaptive way to reduce the error rate. Our method have zero error rate for hiding capacity below 100 kilo bits-per-second (kbps) and 0.3% error for 200 kbps, in comparison to 0.9% error of normal wavelet domain LSB steganography. Signal to Noise Ratio (SNR) values and listening tests results show that the stegano audio is imperceptible from original audio even with hiding capacity up to 200 kbps.

Mazdak Zamani, Azizah Bt Abdul Manaf: Steganography is a technique used to transmit hidden information by modifying an audio signal in an imperceptible manner. The transmission must be possible inspite of subsequent imperceptible alterations (attacks) of the modified signal. We propose a novel approach of substitution technique of audio steganography. Using genetic algorithm, message bits are embedded into multiple, vague and higher LSB layers, resulting in increased robustness. The robustness specially would be increased against those intentional attacks which try to reveal the hidden message and also some unintentional attacks like noise addition as well.

NedeGko Cvejic, Tapio Seppanen: In this work, we increased the capacity of the classic LSB insertion method by performing the embedding process in the wavelet domain. The algorithm uses perfect reconstruction filter banks and embeds additional information inside wavelet domain of audio signal by modifying LSB values of wavelet coefficients. Objective and subjective tests show large advantage of the proposed method over time insertion method. For the same SNR values, proposed algorithm outperforms classical LSB algorithm by 150-200 kbps of hidden data. Subjective experiments showed that wavelet domain information hiding scheme is acoustically more transparent as well.

Ahmad Delforouzi, Mohammad Pooyan: In this paper a novel method for digital audio steganography is presented where encrypted covert data is embedded into the coefficients of host audio (cover signal) in integer wavelet domain. The hearing threshold is calculated in the integer domain and this threshold is employed as embedding threshold. The inverse integer wavelet transform is applied to the modified coefficients to form new audio sequence (stego signal). The characteristics of this method are large payload, high audio quality and full recovery.

Parul Shah, Pranali Choudhari and Suresh Sivaraman: In this paper we present a novel method for digital audio steganography where encrypted covert data is embedded by adaptively modifying wavelet packet coefficients of host audio signal. The major contribution of the proposed scheme is the technique introduced for adaptively modifying the host audio to embed the covert data. The modification of host audio is done by imposing a constraint which forces the modified value to be in the same range as its neighborhood. Due to this constraint the noise introduced due to embedding is very low compared to existing methods. The main advantage with proposed embedding scheme is superior Signal to Noise Ratio (SNR) values, with good hiding capacity and speed. Listening test results also show that distortions in the stego audio is imperceptible from the original audio even with highest hiding capacity. Our method also has zero bit error in recovered data which is one of the most desired features of any steganography technique.

Kaliappan Gopalan, Qidong Shi: Audio steganography using bit modification of time domain audio samples is a simple technique for multimedia data embedding with potential for large payload. Depending on the index of the bit used to modify the samples in accordance with the data to be hidden, the resulting stego audio signal may become perceptible and/or susceptible to incorrect retrieval of the hidden data. This paper presents some results of the tradeoff between the conflicting requirements of data robustness, payload and imperceptibility. Experimental results on both clean and noisy host audio signals indicate that while the payload can be as high as over 3000 bits/s–much higher rate than common audio data embedding techniques–notice ability of embedding is decreased and noise tolerance increased by using higher bit indices than the traditional least significant bit. BER of below one percent were observed for data retrieved from noise-added stego audio signals with 39 dB SNR for embedded payload of over 10 Kbits in a 3.3 s host audio.

Mazdak Zamani1, Azizah Bt Abdul Manaf1, Rabiah Bt Ahmad1, Farhang Jaryani1: A wide range of steganography techniques has been described in this paper. Beside the evaluation of embedding parameters for the existing techniques, two problems - weaknesses- of substitution techniques are investigated which if they could be solved, the large capacity - strength- of substitution techniques would be practical. Furthermore, a novel, principled approach to resolve the problems is presented. Using the proposed genetic algorithm, message bits are embedded into multiple, vague and higher LSB layers, resulting in increased robustness.

József LENTI: In this paper we analyze and test several steganographic techniques on still images. We show that embedding a large amount of data into the picture it can modify its visible properties. We compare the RSA and the elliptic curve (ECC) based digital signatures, and we analyze their advantages and disadvantages in steganography. In steganography it is important that the embedded data size should be minimized. Using ECC minimization of the embedded information is possible, because the minimal block size is smaller than in the case of RSA.

CHAPTER 4

STEGANOGRAPHY METHODS & TECHNIQUES

4.1 METHODS OF STEGANOGRAPHY

4.1.1 Steganography in Text

Encoding secret messages in text can be a very challenging task. This is because text files have a very small amount of redundant data to replace with a secret message. Another drawback is the ease of which text based Steganography can be altered by an unwanted parties by just changing the text itself or reformatting the text to some other form (from .TXT to .PDF, etc.). There are numerous methods by which to accomplish text based Steganography. I will introduce a few of the more popular encoding methods below.

Line-shift encoding involves actually shifting each line of text vertically up or down by as little as 3 centimeters. Depending on whether the line was up or down from the stationary line would equate to a value that would or could be encoded into a secret message. Word-shift encoding works in much the same way that line-shift encoding works, only we use the horizontal spaces between words to equate a value for the hidden message. This method of encoding is less visible than line-shift encoding but requires that the text format support variable spacing Feature specific encoding involves encoding secret messages into formatted text by changing certain text attributes such as vertical and horizontal length of letters such as b, d, T, etc. This is by far the hardest text encoding method to intercept as each type of formatted text has a large amount of features that can be used for encoding the secret message. All three of these text based encoding methods require either the original file or the knowledge of the original files formatting to be able to decode the secret message.

4.1.2 Steganography in Images

Coding secret messages in digital images is by far the most widely used of all methods in the digital world of today. This is because it can take advantage of the limited power of the human visual system (HVS). Almost any plain text, cipher text, image and any other media that can be encoded into a bit stream can be hidden in a digital image. With the continued growth of strong graphics power in computers and the research being put into image based Steganography, this field will continue to grow at a very rapid pace.

Image architecture and digital image compression techniques should be explained. As Duncan Sellars explains "To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. These pixels make up the images raster data." When dealing with digital images for use with Steganography, 8-bit and 24-bit per pixel image files are typical. Both have advantages and disadvantages, as we will explain below .8-bit images are a great format to use because of their relatively small size. The drawback is that only 256 possible colors can be used which can be a potential problem during encoding.

Usually a gray scale color palette e is used when dealing with 8-bit images such as (.GIF) because its gradual change in color will be harder to detect after the image has been encoded with the secret message. 24-bit images offer much more flexibility when used for Steganography. The large numbers of colors (over 16 million) that can be used go well beyond the human visual system (HVS), which makes it very hard to detect once a secret message, has been encoded. The other benefit is that a much larger amount of hidden data can be encoded into a 24-bit digital image a opposed to an 8-bit digital image. The one major drawback to 24-bit digital images is their large size (usually in MB) makes them more suspect than the much smaller 8-bit digital images (usually in KB) when sent over an open system such as the Internet.

Digital image compression is a good solution to large digital images such as the 24-bit images mentioned earlier. There are two types of compression used in digital images, lossy and lossless. Lossy compression such as (.JPEG) greatly reduces the size of a digital image by removing excess image data and calculating a close approximation of the original image. Lossy compression is usually used with 24-bit digital images to reduce its size, but it does carry one major drawback. Lossy compression techniques increase the possibility that the uncompressed secret message will lose parts of its contents because of the fact that lossy compression removes what it sees as excess image data. Lossless compression techniques, as the name suggests, keeps the original digital image in tact without the chance of loss. It is for this reason that it is the compression technique of choice for steganographic uses. Examples of lossless compression techniques are (.GIF and .BMP). The only drawback to lossless image compression is that it doesn't do a very good job at compressing the size of the image data.

4.1.3 Steganography in Audio

Encoding secret messages in audio is the most challenging technique to use when dealing with Steganography. This is because the human auditory system (HAS) has such a dynamic range that it can listen over. To put t his in perspective. the (HAS) perceives over a range of power greater than one million to one and a range of frequencies greater than one thousand to one making it extremely hard to add or remove data from the original data structure. The only weakness in the (HAS) comes at trying to differentiate sounds (loud sounds drown out quiet sounds) and this is what must be exploited to encode secret messages in audio without being detected.

There are two concepts to consider before choosing an encoding technique for audio. They are the digital format of the audio and the transmission medium of the audio. There are three main digital audio formats typically in use. They are Sample Quantization. Temporal Sampling Rate and Perceptual Sampling .

Sample Quantization which is a 16-bit linear sampling architecture used by popular audio formats such as (.WAV and. AIFF). Temporal Sampling Rate uses selectable frequencies (in the KHz) to sample the audio. Generally. the higher the sampling rate is. the higher the usable data space gets. The last audio format is Perceptual Sampling. This format changes the statistics of the audio drastically by encoding only the parts the listener perceives. thus maintaining the sound but changing the signal. This format is used by the most popular digital audio on the Internet today in ISO MPEG (MP3).Transmission medium (path the audio taken from sender to receiver) must also be considered when encoding secret messages in audio. W. Bender introduces four possible transmission mediums:

1) Digital end to end - from machine to machine without modification.

2) Increased/decreased re sampling - the sample rate is modified but remains digital.

3) Analog and re sampled - signal is changed to analog and re sampled at a different rate.

4) Over the air - signal is transmitted into radio frequencies and resembled from a microphone.

4.1.4 Steganography in Video

When information is hidden inside video the program or person hiding the information will usually use the DCT (Discrete Cosine Transform) method. DCT works by slightly changing the each of the images in the video. only so much though so it's isn't noticeable by the human eye. To be more precise about how DCT works, DCT alters values of certain parts of the images, it usually rounds them up. For example if part of an image has a value of 6.667 it will round it up to 7. Steganography in Videos is similar to that of Steganography in Images, apart from information is hidden in each frame of video. When only a small amount of information is hidden inside of video it generally isn't noticeable at all, however the more information that is hidden the more noticeable it will become.

4.1.5 Steganography in Documents

Steganography can be used in documents? Yes it's true! The use of Steganography in documents works by simply adding white space and tabs to the ends of the lines of a document. This type of Steganography is extremely effective, because the use white space and tabs is not visible to the human eye at all, at least in most text/document editors. White space and tabs occur naturally in documents, so there isn't really any possible way using this method of Steganography would cause someone to be suspicious. The most popular piece of software used to perform this type of Steganography is a piece of software called **SNOW.**

4.2 STEGANOGRAPHY TECHNIQUES

4.2.1 Embedding Using Stego key

Steganography embeds a secret message in a cover message, this process is usually parameterized by a stego-key, and the detection or reading of an embedded information is possible only having this key.Figl. 4.1 shows this process.

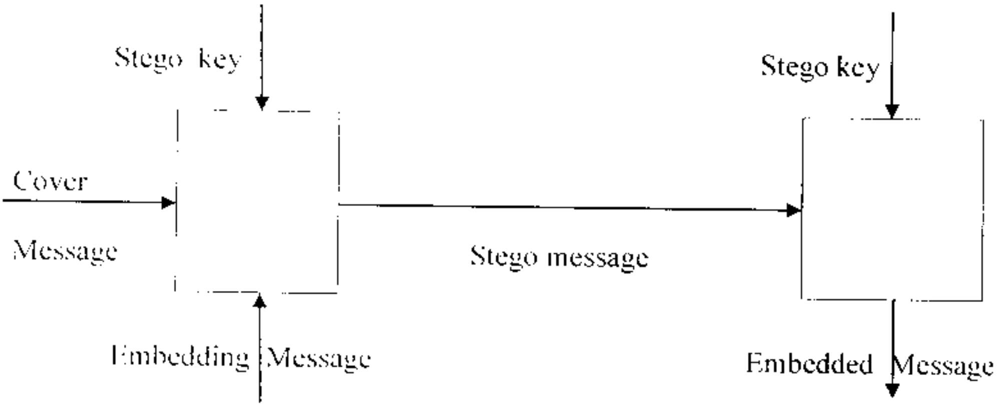


Figure 4.1 Embedding Technique of Steganography

4.2.2 Fingerprinting and Watermarking

Nowadays steganography is more and more important in publishing and broadcasting industries, where the embedding of copyright marks or serial numbers is needed in digital films, photos and other multimedia products. Some steganographic applications are able to scan the Internet, and to detect a copy of a specific image, or the modified image is published so an illegal usage of a copyrighted image can be detected. In the case of audio materials, the automatic monitoring of radio advertisements is also possible, the advertiser can automatically count how many times a specific advertisement was transmitted by a given radio station. Another possible application in the case of still images is to embed captions and other information into the picture so that one does not have to store distinctly the images, and connected information.

When the purpose is the protection of intellectual property, we can make a distinction between fingerprinting and watermarking. In the case of watermarking, copyright information is embedded in a digital media, and this media is transmitted to users. Fingerprinting embeds separate mark in the copies of digital media, this embedded information serves as a serial number, it can be detected who supplied this media to third parties.

Usually 24-bit or 8-bit files are used to store digital images. The former one provides more space for information hiding, however, it can be quite large. The colored representations of the pixels are derived from three primary colors: red, green and blue. 24-bit images use 3 bytes for each pixel, where each primary color is represented by 1 byte. Using 24-bit images each pixel can represent 16,777,216 color values. We can use the lower two bits of these color channels to hide data, then the maximum color change in a pixel could be of 64-color values, but this causes so little change that is undetectable for the human vision system. This simple method is known as Least Significant Bit insertion . Using this method it is possible to embed a significant amount of information with no visible degradation of the cover image.

Several versions of LSB insertion exist. It is possible to use a random number generator initialized with a stego-key and its output is combined with the input data, and this is embedded to a cover image. For example in the presence of an active warden it is not enough to embed a message in a known place (or in a known sequence of bits) because the warden is able to modify these bits, even if he can't decide whether there is a secret message or not, or he can't read it because it is encrypted. The usage of a stego-key is important, because the security of a protection system should not be based on the secrecy of the algorithm itself, instead of the choice of a secret key.

4.2.3 Public key steganographic Technique

As another possible way the algorithm requires the pre-existence of a shared secret key to designate pixels which should be tweaked. In this case both the sender and the receiver must have this secret. Suppose that the communicating parties do not have the opportunity to agree a secret key, but one of them (e.g. Bob) has a private/public key pair, and his partner knows the public key. In the case of a passive warden Alice knowing Bob's public key encrypts her message with this key, embeds it in a known channel (known position in the cover media), and sends it to Bob. Bob cannot be sure whether the channel contains a hidden message, but he can try to decrypt the random-looking string-sequence with his private key, and check whether it is a message or not.

Another approach is the cover image escrow scheme (or source extraction), where the extractor is required with the original cover image, and the cover image is subtracted from the stego image before the extraction of the embedded information. In this scheme, the user cannot read the embedded data, it is only possible to have the original unmodified picture, but these types of algorithms are characterized as robust against signal distortions.

4.2.4 Transform Domain Based Steganography

The destination extraction algorithms can be divided into two groups: spatial/time domain and transform domain techniques. In the former case information is embedded in the spatial domain in the case of images, and in time domain in the case of audio materials. The transform domain methods operate in the Discrete Cosine Transform, Fourier or wavelet transform domains of the host signal.

The Patchwork algorithm (developed at the MIT) selects random pairs of pixels, and increases the brightness of the brighter pixel and decreases the brightness of the other. This algorithm shows a high resistance to most non-geometric image modifications. If it is important to provide a protection against filtering attacks, then the information hiding capacity is limited.

High color quality images are compressed usually using a lossy compression method as, for example, in the case of Jpeg images. In Jpeg algorithm the pixels are first transformed into a luminance-chrominance space. The chrominance is then downsampled – it is possible because the HVS (Human Vision System) is less sensitive to chrominance changes than to luminance changes – so the volume of the data is reduced. Discrete Cosine Transform is then applied on the groups of 8 × 8 pixels. The next step causes the most loss in the case of JPEG, where the coefficients are scalar quantized (it is possible because if we reduce the coefficients of higher frequencies to zero, the changes to the original image will cause only small changes that a human viewer could not detect under normal circumstances).The final steps are lossless, when these reduced coefficients are also compressed and a header is added to the JPEG image. Steganographic applications usually operate after the quantization step, for example Jpeg-Jsteg, and SysCoP. SysCoP uses a position sequence generator. The inputs of the generator are the image data and user key, the output is a position sequence for selecting blocks where the code is embedded.

The block consists in this case of 8 × 8 pixels, it can be contiguous – the block is a square in the image – or distributed, where the pixels are randomly selected. A label bit is embedded through setting specific relationship among three quantized elements of a block, and the algorithm contains a checking mechanism to test whether the actual block is capable or not to store this information, how big modification is needed to store one bit information among these pixels.

4.3 TECHNIQUES FOR AUDIO STEGANOGRAPHY

In audio steganography, secret message is embedded into digitized audio signal which result slight altering of binary sequence of the corresponding audio file. There are several methods are available for audio steganography. Some of them are as follows :

4.3.1 LSB Coding

Sampling technique followed by Quantization converts analog audio signal to digital binary sequence.

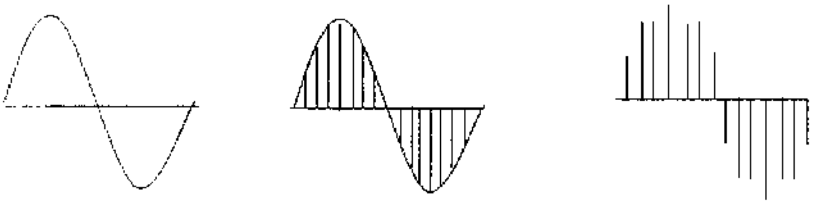


Figure 4.2 Sampling of the Sine Wave followed by Quantization process

In this technique LSB of binary sequence of each sample of digitized audio file is replaced with binary equivalent of secret message. For example if we want to hide the letter 'A' (binary equivalent **01100101**) to an digitized audio file where each sample is represented with 16 bits, then LSB of 8 consecutive samples (each of 16 bit size) is replaced with each bit of binary equivalent of the letter 'A'.

Sampled Audio Stream (16 bit)	'A' in binary	Audio stream with encoded message
1001 1000 0011 1100	0	1001 1000 0011 110**0**
1101 1011 0011 1000	1	1101 1011 0011 100**1**
1011 1100 0011 1101	1	1011 1100 0011 110**1**
1011 1111 0011 1100	0	1011 1111 0011 110**0**
1011 1010 0111 1111	0	1011 1010 0111 111**0**
1111 1000 0011 1100	1	1111 1000 0011 110**1**
1101 1100 0111 1000	0	1101 1100 0111 100**0**
1000 1000 0001 1111	1	1000 1000 0001 111**1**

Table 4.1 Audio Bit Steam Representation

4.3.2 Phase Coding

Human Auditory System (HAS) can't recognize the phase change in audio signal as easy it can recognize noise in the signal. The phase coding method exploits this fact. This technique encodes the secret message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to- noise ratio.

4.3.3 Spread Spectrum

There are two approaches are used in this technique: the direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS). Direct-sequence spread spectrum (DSSS) is a modulation technique used in telecommunication. As with other spread spectrum technologies, the transmitted signal takes up more bandwidth than the information signal that is being modulated.

Direct-sequence spread-spectrum transmissions multiply the data being transmitted by a "noise" signal. This noise signal is a pseudorandom sequence of 1 and −1 values, at a frequency much higher than that of the original signal, thereby spreading the energy of the original signal into a much wider band.

The resulting signal resembles white noise. However, this noise-like signal can be used to exactly reconstruct the original data at the receiving end, by multiplying it by the same pseudorandom sequence (because 1 × 1 = 1, and −1 × −1 = 1). This process, known as "de-spreading", mathematically constitutes a correlation of the transmitted Pseudorandom Noise (PN) sequence with the receiver's assumed sequence. For de-spreading to work correctly, the transmit and receive sequences must be synchronized. This requires the receiver to synchronize its sequence with the transmitter's sequence via some sort of timing search process.

In contrast, frequency-hopping spread spectrum pseudo-randomly retunes the carrier, instead of adding pseudo-random noise to the data, which results in a uniform frequency distribution whose width is determined by the output range of the pseudo-random number generator.

4.3.4 Echo Hiding

In this method the secret message is embedded into cover audio signal as an echo. Three parameters of the echo of the cover signal namely amplitude, decay rate and offset from original signal are varied to represent encoded secret binary message. They are set below to the threshold of Human Auditory System (HAS) so that echo can't be easily resolved.

Video files are generally consists of images and sounds, so most of the relevant techniques for hiding data into images and audio are also applicable to video media. In the case of Video steganography sender sends the secret message to the recipient using a video sequence as cover media. Optional secret key 'K' can also be used during embedding the secret message to the cover media to produce 'stego-video'. After that the stego-video is communicated over public channel to the receiver. At the receiving end, receiver uses the secret key along with the extracting algorithm to extract the secret message from the stego-object. The original cover video consists of frames represented by Ck(m, n) where 1 £ k £ N. 'N' is the total number of frame and m, n are the row and column indices of the pixels.

respectively. The binary secret message denoted by Mk(m, n) is embedded into the cover video media by modulating it into a signal. Mk(m, n) is defined over the same domain as the host Ck(m, n).The stego-video signal is represented by the equation

$$Sk(m, n) = Ck(m, n) + ak\ (m, n)\ Mk(m, n)\ ,\ k = 1, 2, 3\ .\ .\ .N$$

where ak (m, n) is a scaling factor. For simplicity ak (m, n) can be considered to be constant over all the pixels and frames. So the equation becomes:

$$k(m, n) = Ck(m, n)+a\ (m, n)\ Mk(m, n)\ ,\ k = 1, 2, 3\ .\ .\ .N$$

CHAPTER 5

GENETIC ALGORITHM

5.1 INTRODUCTION

A genetic algorithm is a problem solving technique which mimics the process of natural selection. Potential solutions to a problem are viewed as biological life forms which can reproduce. The process begins with a random population of these life forms which are tested to see how close they are to being solutions. The better solutions are deemed more *fit* than other members of the population and are allowed to mate to produce offspring which will compose a new population. This process is repeated until an acceptable solution to the problem is found. Ordered sets arise naturally and frequently in all areas of general algebra. Ordered sets consisting of sub algebras, normal subgroups, and ideals are some common examples.

The idea of employing an "evolutionary strategy" in problem solving was introduced in a 1973 paper of **I. Rechenberg.** His ideas lead to the development of the concept of a genetic algorithm in 1975 by John Holland and his students present a restricted model of a genetic algorithm here as a basic introduction to the idea.

John Holland. from the University of Michigan began his work on genetic algorithms at the beginning of the 60s. A first achievement was the publication of *Adaptation in Natural and Artificial System* in 1975.

Holland had a double aim : to improve the understanding of natural adaptation process. and to design artificial systems having properties similar to natural systems.

The basic idea is as follows: the genetic pool of a given population potentially contains the solution, or a better solution, to a given adaptive problem. This solution is not "active" because the genetic combination on which it relies is split between several subjects. Only the association of different genomes can lead to the solution. Simplistically speaking, we could by example consider that the shortening of the paw and the extension of the fingers of our basilosaurus are controlled by 2 "genes". No subject has such a genome, but during reproduction and crossover, new genetic combination occur and, finally, a subject can inherit a "good gene" from both parents

Holland method is especially effective because he not only considered the role of mutation (mutations improve very seldom the algorithms), but he also utilized genetic recombination, (crossover) : these recombination. the crossover of partial solutions greatly improve the capability of the algorithm to approach, and eventually find, the optimum.

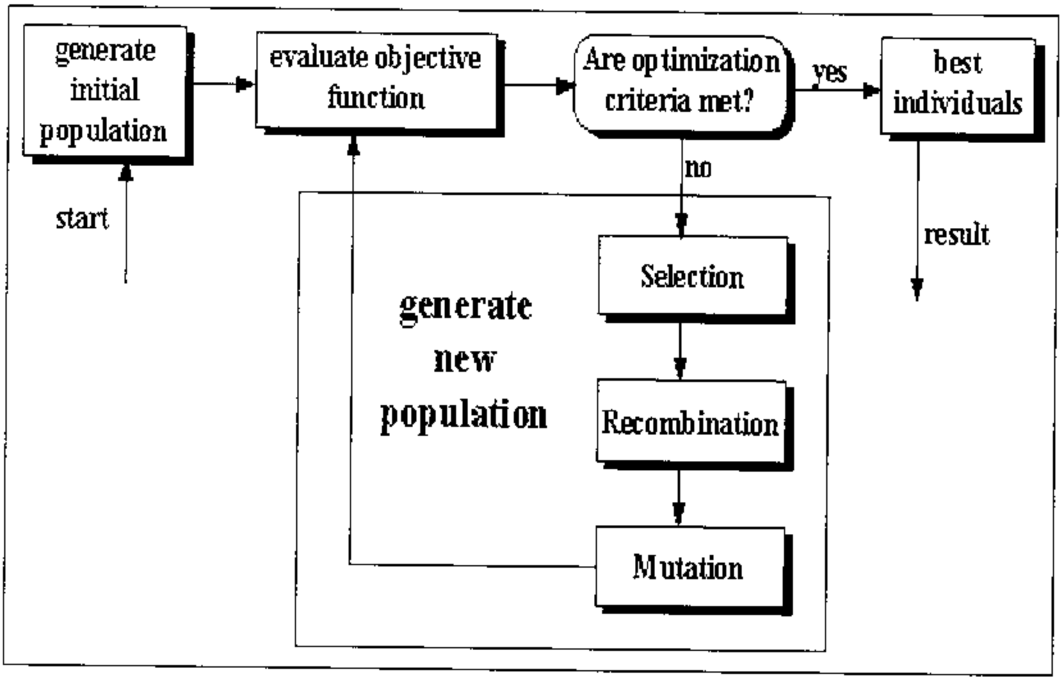


Figure 5.1 Structure Of a Simple Genetic Algorithm

5.2 BIOLOGICAL INSPIRATION

The characteristics of a biological life form are determined by genes (block of DNA) linked together in strings called *chromosomes*. When two life forms reproduce, corresponding chromosomes from each parent are twisted together in a process called *recombination* or *crossover* to form a chromosome in the offspring. This process occurs in the following way. A chromosome from Parent A is paired with a chromosome from Parent B. A random crossover point along the chromosome is chosen, and the two parent chromosomes are cut at this point. The genetic material from Parent A's chromosome before the crossover point is glued together with the material from Parent B's chromosome after the crossover point to form a new chromosome for the offspring .Recombination has the property that if both parents share a particular pattern in their chromosomes then this pattern will carry over to the offspring. Therefore, good (and bad) qualities which are shared by the parents can be passed on to the offspring.

The process of copying genes from the parents to offspring is imperfect. There are sometimes errors in copying which we call mutation. This mutation introduces a certain randomness into the genetic material of the offspring. This randomness might be negative or positive. It may even have no noticeable effects on the offspring.

If an organism is strong and survives long enough to reproduce, then some of its genetic material is passed on to the next generation. If an organism is too weak to survive long enough to reproduce, then its genetic material is removed from the population. Genetic algorithms attempt to mimic this situation. Possible solutions to a problem are envisioned as organisms each with a chromosome that contains the genetic material which encodes its individual traits. This chromosome is simply a list of symbols – usually 0's and 1's. At the simplest level, recombination works as described above. A crossover point is chosen at random. The two parents' chromosomes are cut at this crossover point. The first segment of one is glued to the second segment of the other to form the chromosome of the offspring. The offspring's chromosome is then mutated by randomly changing some of the symbols in it.

The genetic algorithm employs a fitness function which determines how good of a solution a particular life form is to the problem being addressed. Those organisms which display a greater fitness are given a greater chance to reproduce. First, a random population of organisms is generated, and the entire population is tested for fitness. Then some of the members of the population are selected for reproduction with the most fit organisms being more likely to reproduce. Some of the genetic material in the offspring is mutated.

Then the new generation of offspring replaces the previous generation. To insure that the new generation is at least as fit as the previous generation, some of the most fit members of the parent generation may be included with the offspring. This is called *elitism.* The process is repeated until an acceptable solution is found. Here is an outline of a typical genetic algorithm:

(1) **Initial Population**: An initial population of organisms is randomly generated.

(2) **Fitness Testing**: The fitness function is applied to each member of the population.

(3) **Solution**: If an acceptable solution is found in the population during testing then the algorithm terminates.

(4) **Reproduction**:

4.1 **Selection**: Pairs of organisms are selected from the population for mating with probability based on fitness.

4.2 **Recombination**: A crossover or recombination operation is applied to the chromosomes of each pair selected for reproduction to produce a new organism.

4.3 **Mutation**: The chromosome of the new organism is mutated with some small probability.

(5) **Next Generation**: Some or all of the offspring produced in the previous step are chosen to form the new population. This population may also include the most elite members of the previous population.

(6) **Repeat**: Go to step 2.

5.3 ENCODING AND FITNESS

The structure of a genetic algorithm is quite general. The means of selection, recombination, and mutation are often not tied to the problem at hand. The specific problem affects most directly how organisms are encoded and how fitness is computed. The most popular means of encoding a chromosome is as a sequence of 0's and 1's as most data can usually be represented in this way. However, genetic algorithms have been applied with chromosomes that are strings of integers, floating point numbers, and arbitrary symbols. John Koza has even applied the ideas behind genetic algorithms to "breed" computer programs in genetic programming. Recombination may be more complicated in these more general chromosomes, so we isolate our attention to strings of 0's and 1's.

By far the most difficult and most important step in building a genetic algorithm is constructing the fitness function. This is also the most problem-specific step. As we will demonstrate with our genetic algorithm for drawing ordered sets below, the fitness function can be influenced by a variety of parameters. The fitness function must be fast because the fitness of every organism in a population must be tested every generation in a genetic algorithm.

5.4 CROSSOVER

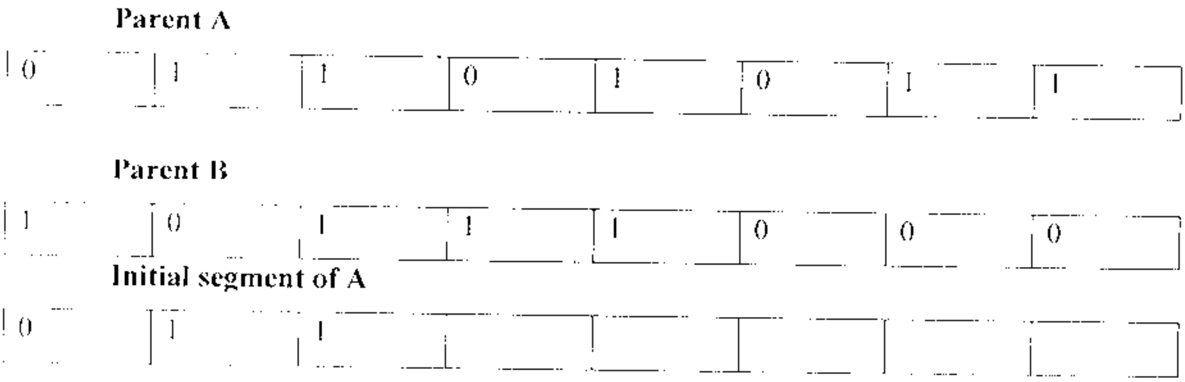
A crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation. In reproduction, good strings in a population are probabilistic-ally assigned a larger number of copies and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool.

The two strings participating in the crossover operation are known as parent strings and the resulting strings are known as children strings. It is intuitive from this construction that good sub-strings from parent strings can be combined to form a better child string, if an appropriate site is chosen. With a random site, the children strings produced may or may not have a combination of good sub-strings from parent strings, depending on whether or not the crossing site falls in the appropriate place. But this is not a matter of serious concern, because if good strings are created by crossover, there will be more copies of them in the next mating pool generated by crossover. It is clear from this discussion that the effect of cross over may be detrimental or beneficial. Thus, in order to preserve some of the good strings that are already present in the mating pool, all strings in the mating pool are not used in crossover. When a crossover probability, defined here as pc is used, only 100pc per cent strings in the population are used in the crossover operation and 100(1pc) per cent of the population remains as they are in the current population. A crossover operator is mainly responsible for the search of new strings even though mutation operator is also used for this purpose sparingly.

Many crossover operators exist in the GA literature. One site crossover and two site crossover are the most common ones adopted. In most crossover operators, two strings are picked from the mating pool at random and some portion of the strings are exchanged between the strings. Crossover operation is done at string level by randomly selecting two strings for crossover operations. A one site crossover operator is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the crossing site

5.4.1 One Point Crossover:

The simplest type of recombination is one point crossover. Chromosomes are strings of symbols (here, 0's and 1's). We begin with two chromosomes, one from Parent A and one from Parent B, both the same length. A crossover point is selected randomly. The two chromosomes are cut at this point, and a new chromosome is formed by using the chromosome from Parent A before the crossover point and from Parent B after the crossover point. This is depicted in Figure 5.2.



Final segment of B

			1	1	0	0	0

Offspring

0	1	1	1	1	0	0	0

Figure 5.2 One Point Crossover

5.4.2 Two Point Crossover

A first generalization of one point crossover is two point crossover. In two point crossover. two crossover points are randomly chosen. Genetic material is copied from A before the first crossover point. Between the crossover points, material is taken from Parent B. After the second crossover point. material is again taken from A. This is depicted in Figure 5.3.

Parent A

0	1	1	0	1	0	1	1

Parent B

1	0	1	1	1	0	0	0

Initial & Final segment of A

0	1				0	1	1

Middle segment of B

		1	1	1			

Offspring

0	1	1	1	1	0	1	1

Figure 5.3 Two Point Crossover

From two point crossover, one can imagine three point crossover, four point. five point, and so on. The logical conclusion of this is uniform crossover. In uniform crossover, each symbol in the offspring's chromosome is chosen randomly to be equal to the corresponding symbol of the chromosome of either Parent A or Parent B. This is depicted in Figure 5.4

Parent A	0	1	1	0	1	0	1	1
Parent B	1	0	1	1	1	0	0	0
Offspring	0	0	1	0	1	0	0	1

Figure 5.4 Uniform Crossover

One observation that should be made about crossover in any of these forms is that if a pattern appears in the chromosomes of both parents, then recombination will preserve that pattern. For example, both parents above have 1's as the third and fifth symbol in their chromosomes and 0 as the sixth. You can see that in one point, two point, and uniform crossover the offspring has the same pattern.

5.5 MUTATION

Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level; when the bits are being copied from the current string to the new string, there is probability that each bit may become mutated. This probability is usually a quite small value, called as mutation probability pm. A coin toss mechanism is employed; if random number between zero and one is less than the mutation probability, then the bit is inverted, so that zero becomes one and one becomes zero. This helps in introducing a bit of diversity to the population by scattering the occasional points. This random scattering would result in a better optima, or even modify a part of genetic code that will be beneficial in later operations. On the other hand, it might produce a weak individual that will never be selected for further operations.

Mutation is usually applied after the process of recombination. The simplest manner in which this is done is to select a small number of symbols in the offspring's chromosome and replace them with random symbols. If the only symbols are 0 and 1, then this amounts to selecting a few random symbols and toggling them between 0 and 1. In most circumstances, mutation should only be applied in a very limited way. If a population possesses a few very fit members then it may only take a few generations before the entire population resembles these members – even if they are not acceptable solutions. Mutation helps to avoid this *premature convergence.* In so doing, mutation helps to provide a deeper gene pool so that the genetic algorithm may have more chances to find a good solution. The need for mutation is to create a point in the neighborhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population. For example, the following population having four eight bit strings may be considered:

01101011

00111101

00010110

01111100

It can be noticed that all four strings have a 0 in the left most bit position. If the true optimum solution requires 1 in that position, then neither reproduction nor crossover operator described above will be able to create 1 in that position. The inclusion of mutation introduces probability of turning 0 into 1. These three operators are simple and straightforward. The reproduction operator selects good strings and the crossover operator recombines good sub-strings from good strings together, hopefully, to create a better sub-string. The mutation operator alters a string locally expecting a better string. Even though none of these claims are guaranteed and/or tested while creating a string, it is expected that if bad strings are created they will be eliminated by the reproduction operator in the next generation and if good strings are created, they will be increasingly emphasized. Further insight into these operators, different ways of implementations and some mathematical foundations of genetic algorithms can be obtained from GA literature.

Application of these operators on the current population creates a new population. This new population is used to generate subsequent populations and so on, yielding solutions that are closer to the optimum solution. The values of the objective function of the individuals of the new population are again determined by decoding the strings. These values express the fitness of the solutions of the new generations. This completes one cycle of genetic algorithm called a generation. In each generation if the solution is improved, it is stored as the best solution. This is repeated till convergence.

5.6 FITNESS FUNCTION

GAs mimics the survival-of-the-fittest principle of nature to make a search process. Therefore, GAs is naturally suitable for solving maximization problems. Maximization problems are usually transformed into maximization problem by suitable transformation. In general, a fitness function F(i) is first derived from the objective function and used in successive genetic operations. Fitness in biological sense is a quality value which is a measure of the reproductive efficiency of chromosomes.

In genetic algorithm, fitness is used to allocate reproductive traits to the individuals in the population and thus act as some measure of goodness to be maximized. This means that individuals with higher fitness value will have higher probability of being selected as candidates for further examination. Certain genetic operators require that the fitness function be non-negative, although certain operators need not have this requirement. For maximization problems, the fitness function can be considered to be the same as the objective function or $F(i) = O(i)$. For minimization problems, to generate non-negative values in all the cases and to reflect the relative fitness of individual string, it is necessary to map the underlying natural objective function to fitness function form. A number of such transformations is possible.

CHAPTER 6

ANALYSIS OF AUDIO STEGANOGRAPHY USING GENETIC ALGORITHM

6.1 INTRODUCTION

The process of embedding secret information (text data) into the selected audio file (wav) using "Genetic Algorithm" is performed and analyzed in this chapter. The three main properties of audio steganography is analyzed for the stego audio file. The properties of audio steganography are:

- ✓ **Capacity**
- ✓ **Robustness**
- ✓ **Transparency**

6.1.1 Capacity

Capacity of an information hiding scheme refers to the amount of information that a data hiding scheme can successfully embed without introducing perceptual distortion in the marked media. In the case of audio, it evaluates the amount of possible embedding information into the audio signal. The embedding capacity is the all included embedding capacity (not the payload) and can be measured in percent (%), bits per second or frame and bits per mega byte or kilo byte audio signal. In the other words, the bit rate of the message is the number of the embedded bits within a unit of time and is usually given in bits per second (bps). Some audio

Steganography applications, such as copy control, require the insertion of a serial number or author ID, with the average bit rate of up to 0.5 bps. For a broadcast monitoring watermark, the bit rate is higher, caused by the necessity of the embedding of an ID signature of a commercial within the first second at the start of the broadcast clip, with an average bit rate up to 15 bps. In some envisioned applications, e.g. hiding speech in audio or compressed audio stream in audio, algorithms have to be able to embed message with the bit rate that is a significant fraction of the host audio bit rate, up to 150 kbps.

6.1.2 Robustness

Robustness measures the ability of embedded data or watermark to withstand against intentional and unintentional attacks. Unintentional attacks generally include common data manipulations such as lossy compression, digital-to-analog conversion, re-sampling, re-quantization, etc. whereas intentional attacks cover a broad range of media degradations which include addition white and colored noise, rescaling, rotation (for image and video steganography schemes), resizing, cropping, random chopping, and filtering attacks .

Also, the robustness of the algorithm is defined as an ability of the data detector to extract the embedded message after common signal processing manipulations. Applications usually require robustness in the presence of a predefined set of signal processing modifications, so that message can be reliably extracted at the detection side. For example, in radio broadcast monitoring, embedded message need only to survive distortions caused by the transmission process, including dynamic compression and low pass filtering, because the data detection is done directly from the broadcast signal. On the other hand, in some algorithms robustness is completely undesirable and those algorithms are labelled fragile audio steganography algorithm.

6.1.3 Transparency

Transparency evaluates the audible distortion due to signal modifications like message embedding or attacking. In most of the applications, the steganography algorithm has to insert additional data without affecting the perceptual quality of the audio host signal. The fidelity of the steganography algorithm is usually defined as a perceptual similarity between the original and stego audio sequence. However, the quality of the stego audio is usually degraded, either intentionally by an adversary or unintentionally in the transmission process, before a person perceives it. In that case, it is more adequate to define the fidelity of a steganography algorithm as a perceptual similarity between the stego audio and the original host audio at the point at which they are presented to a consumer. In order to meet fidelity constraint of the embedded information, the perceptual distortion introduced due to embedding should be below the masking threshold estimated based on the HAS/HVS and the host media.

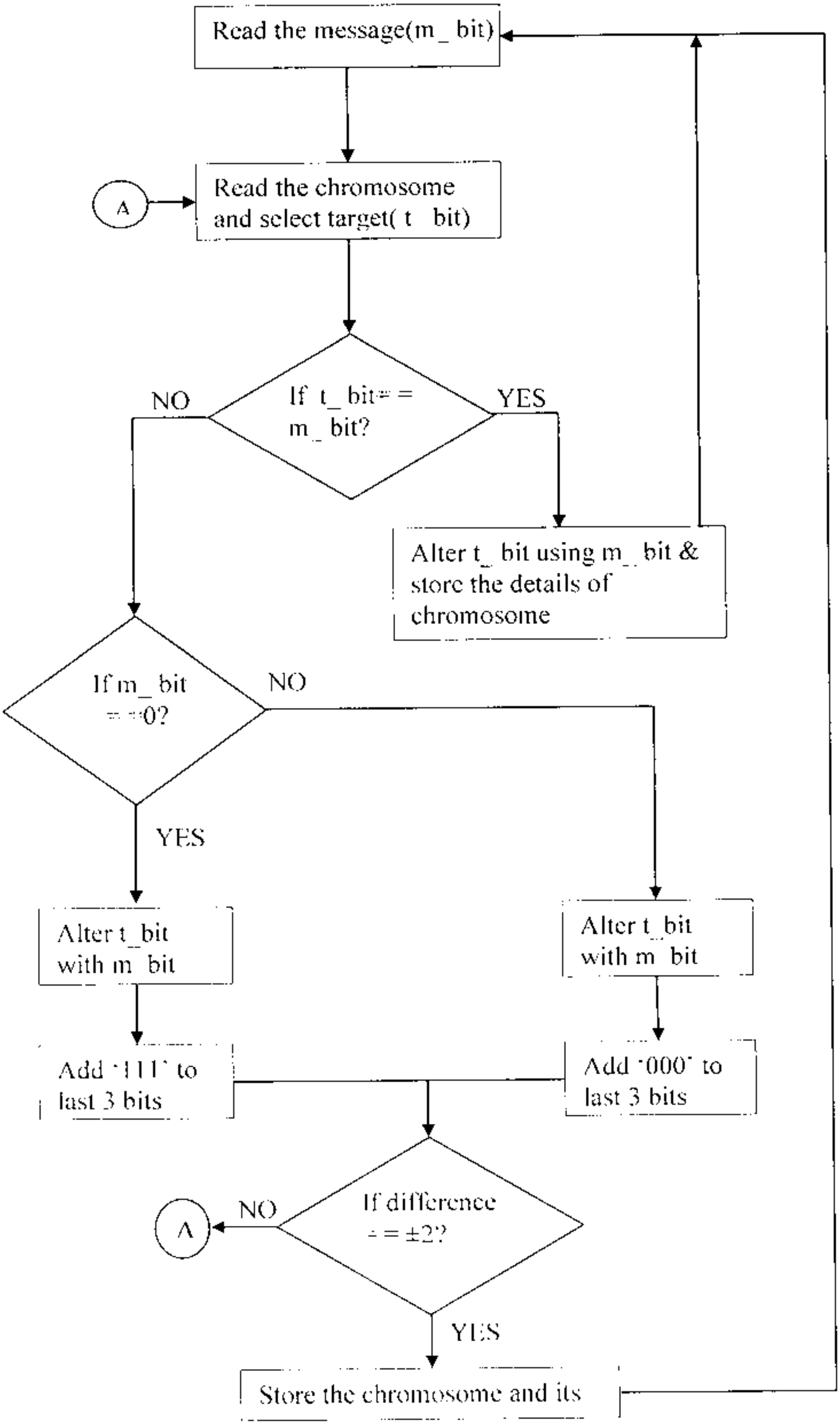


Figure 6.1 Data Embedding Process

6.2 DATA EMBEDDING PROCESS

There are three steps in the data embedding process namely alteration, modification and verification.

6.2.1 Alteration

In this process the message bit of the secret data is substituted with the target bit in the host audio file. Target bit in the host audio is decided by the user. In this alteration process the target bit is simply replaced by the secret data bit.

6.2.2 Modification

The modification process is the important and essential part of the algorithm. The entire result of the algorithm depends on the modification process. This process tries to reduce error due to alteration of secret data bits into the host audio signal and improves the transparency. This can be better understood from figure 6.1

The initial step in the modification process is to read the secret data bit (m_ bit) and the host audio signal. Select a chromosome from the host audio file and select the target bit(t_ bit) for embedding the secret data bit(m_ bit). Check if the t_ bit and the m_ bit are of equal value (t_ bit= =m_ bit). If t_ bit and m_b it are equal, then alter the t_ bit with m_ bit and store the chromosome details.

If the t_ bit and m_ bit are not equal, then check if the m_ bit is equal to '0'(m_ bit= '0). If the condition are satisfied then replaces the target bit by message bit and modify the last three bits by '111'. Calculate the difference between the original audio file and the modified audio file. If the difference is satisfactory then the next secret data bit is considered for embedding process.

If the t_ bit and m_ bit are not equal, then check if the m_ bit is equal to '1'(m_ bit= '1). If the condition is satisfied then replace the target bit by message bit and modify the last three bits by '000'. Calculate the difference between the original audio file and the modified audio file. If the difference is satisfactory then the next secret data bit is considered for embedding process.

If the difference calculated is not satisfactory, then the corresponding chromosome is discarded and the new chromosome sample is considered for embedding of the secret data bit. This process is repeated until a perfect target bit location is obtained for embedding the secret data bit.

6.2.3 Verification

This step is considered as a testing process. In this stage the original host audio signal is compared with the modified audio signal and checked for error. If the obtained error difference is minimum and does not affect the quality of the audio signal. then the new stego audio file is constructed. The modified samples are made to undergo further alteration process. if the error obtained during verification process is large and degrades the audio signal quality. The new stego signal has the embedding details of the secret data bit into the host audio file.

The proposed method can be explained by:

Step1: Read the audio file and the secret text data file into binary.

Step2: Select the secret data bit.

Step3: Select a chromosome from the audio file &choose the target bit for embedding.

Step4: check for chromosome location presence in the location array.

Step5: If the location is already present ,go to step3. else go to step6.

Step6: If the secret bit & target bit are equal. then replace the target bit by secret bit , store the location & go to step9.

Step7: Check secret bit = 0. if so embed the target bit by secret bit & perform crossover with the selected chromosome and the predefined chromosome ('00000111')& go to step9.

Step8: Check secret bit = 1. if so embed the target bit by secret bit & perform crossover with the selected chromosome and the predefined chromosome ('00001000')& go to step9.

Step9: Calculate the error between the original chromosome and the modified chromosome.

Step10: If the error obtained is < ±2. then the modified chromosome sample is termed as fittest chromosome sample.

Step11: The fittest chromosome sample is replaced in the original chromosome sample location & store the chromosome sample location.

Step12: Select the next secret bit and go to step3.

Step13: If the condition in step6,7&8 fails. go to step3.

- ✓ The above steps allow us to embed the secret data bit into the audio file.

6.3 DATA DECODING PROCESS

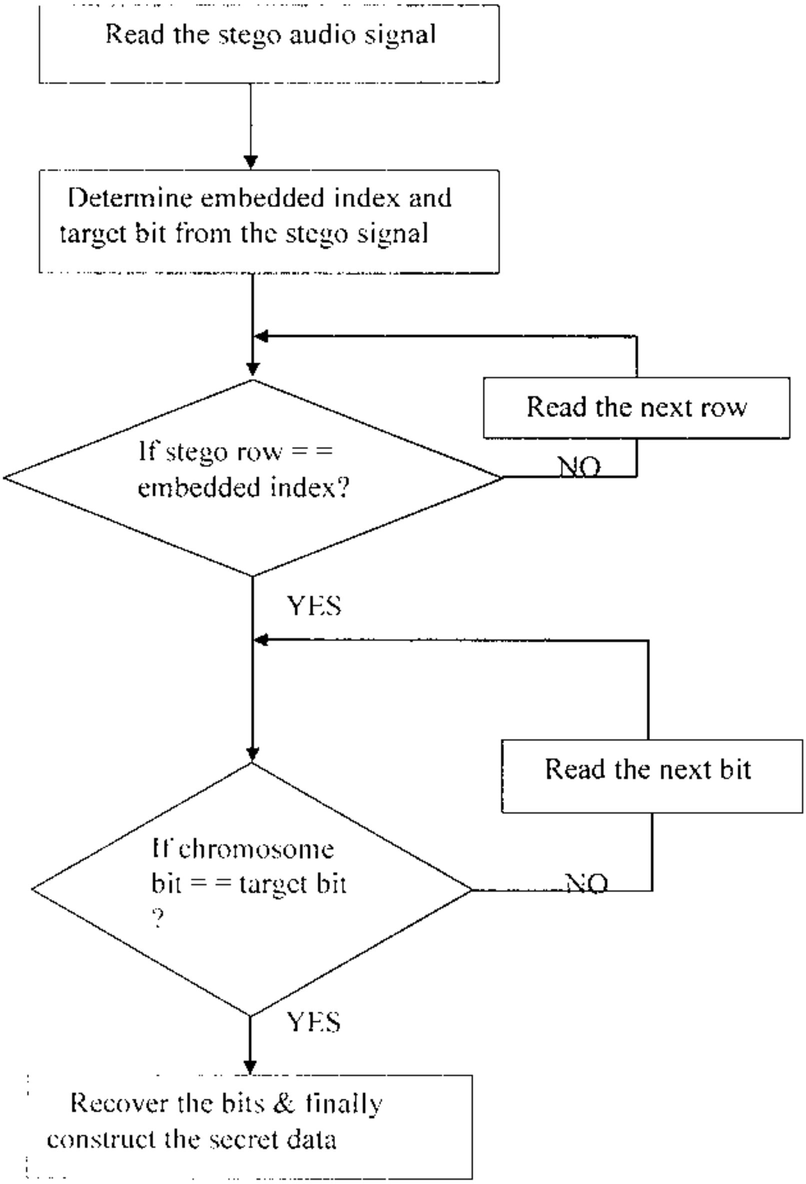

Read the stego audio signal
Determine embedded index and target bit from the stego signal
If stego row = = embedded index?
Read the next row
NO
YES
If chromosome bit = = target bit ?
Read the next bit
NO
YES
Recover the bits & finally construct the secret data

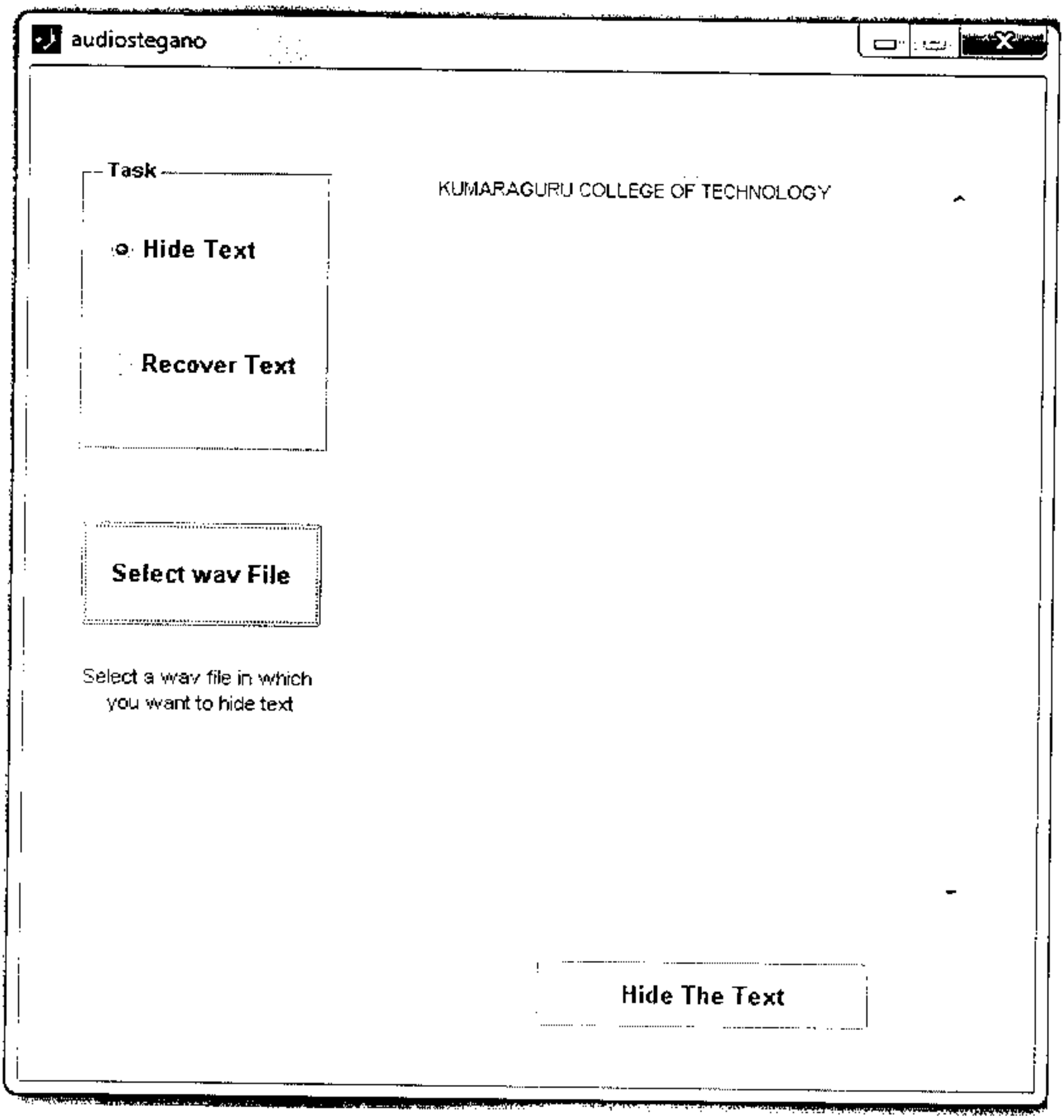
Figure 6.2 Data Decoding Process

The stego audio file which has the secret data bit is read using wave read function which is an in build function in the MATLAB. The embedded bit index location and the target bit location which is embedded in the stego audio file along with the secret data bit is retrieved. After the retrieval of the embedded bit index location, the stego file is now converted into binary data. Now the embedded bit index location is searched in the stego file, if the location is determined, then the secret data bit is retrieved from the target bit specified. After searching the entire embedded index location, the retrieved secret data bits are converted into ASCII format and the ASCII format data is the hidden secret message.

CHAPTER 7

SIMULATION RESULTS

7.1 ENTER THE TEXT TO HIDE:

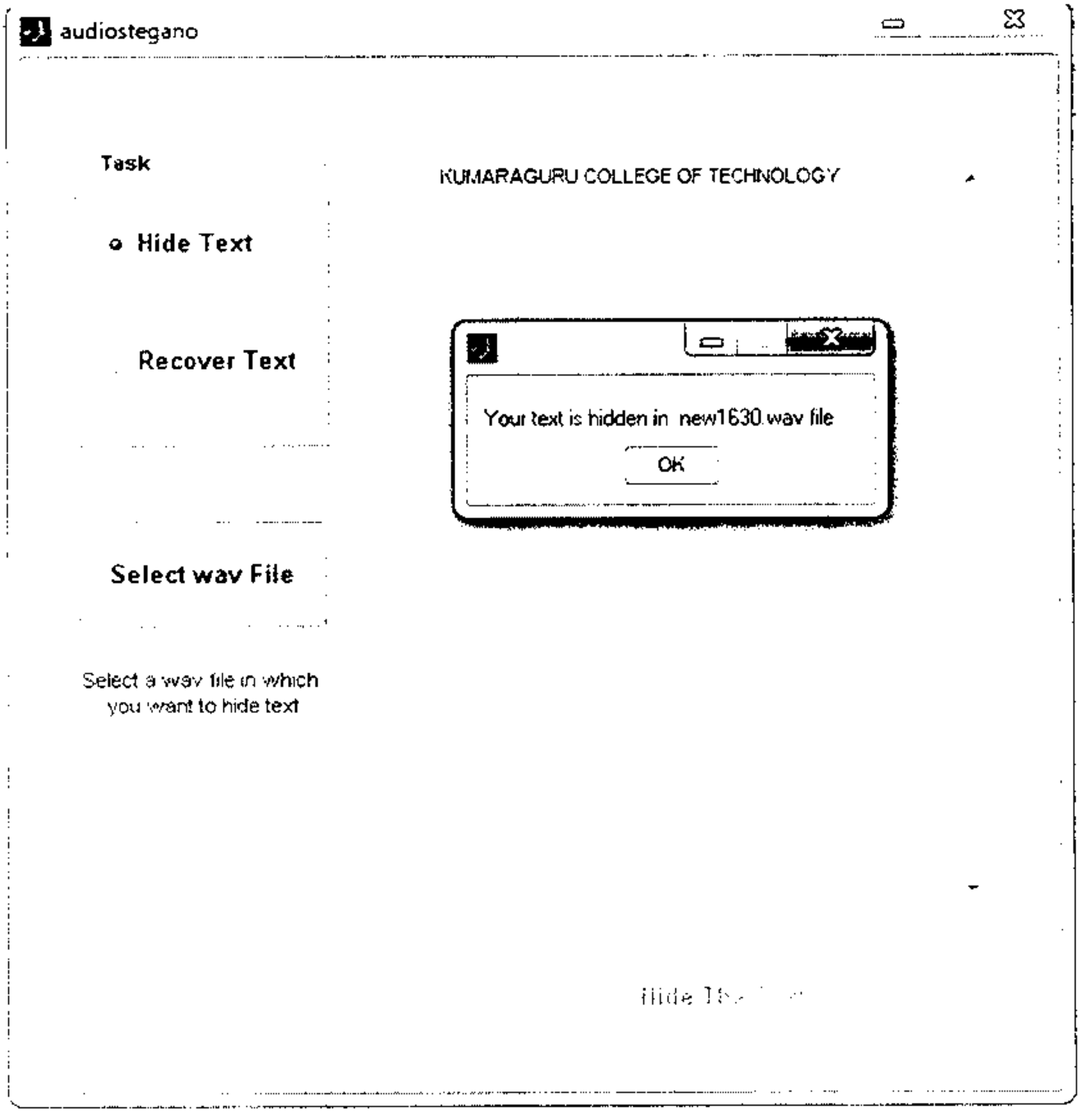


KUMARAGURU COLLEGE OF TECHNOLOGY

Select a wav file in which you want to hide text

Figure 7.1 Data Encoding Process

7.2 STEGO AUDIO FILE GENERATION:

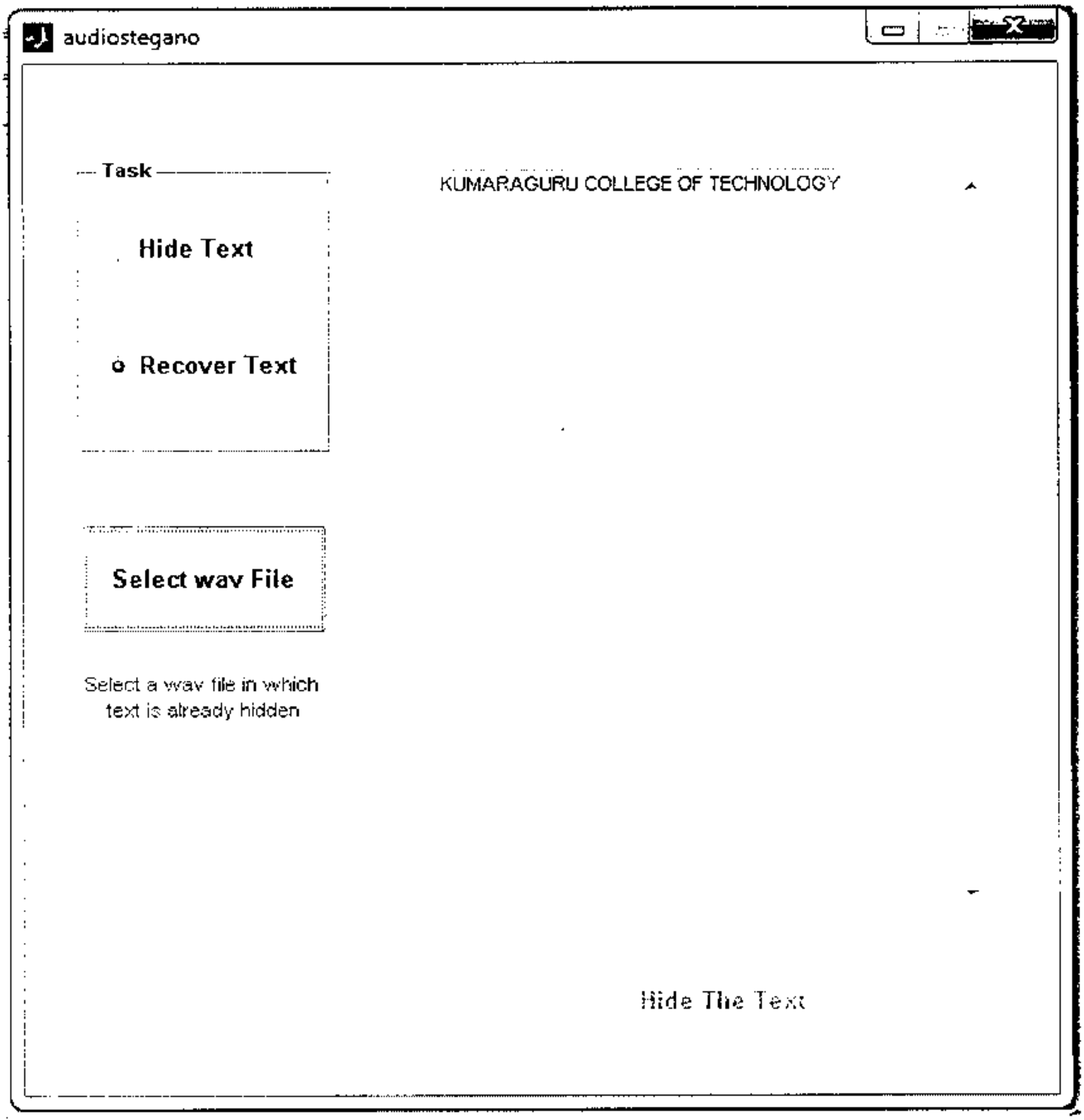


Select a wav file in which you want to hide text

Your text is hidden in new1630.wav file

Figure 7.2 Stego Audio File Generation

7.3 DECODED TEXT DATA FROM STEGO AUDIO FILE:



Select a wav file in which text is already hidden

KUMARAGURU COLLEGE OF TECHNOLOGY

Figure 7.3 Decoding Secret Data

7.4 SIMULATION OF ORIGINAL AND STEGO AUDIO FILE:

7.4.1 Audio File Selected For Hiding Text Data:

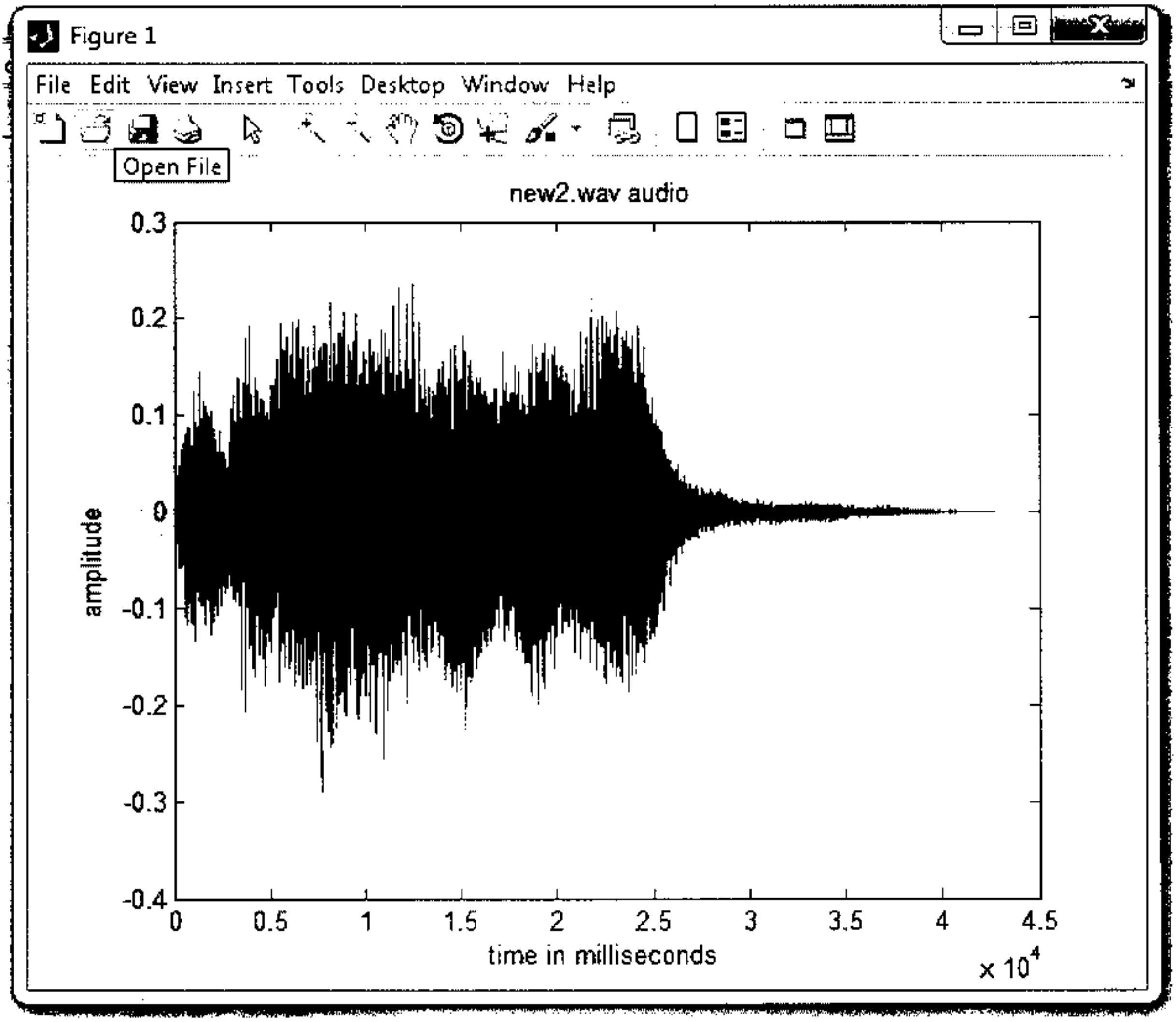


Figure 7.4 Simulation of Cover Audio File

7.4.2 Stego Audio File

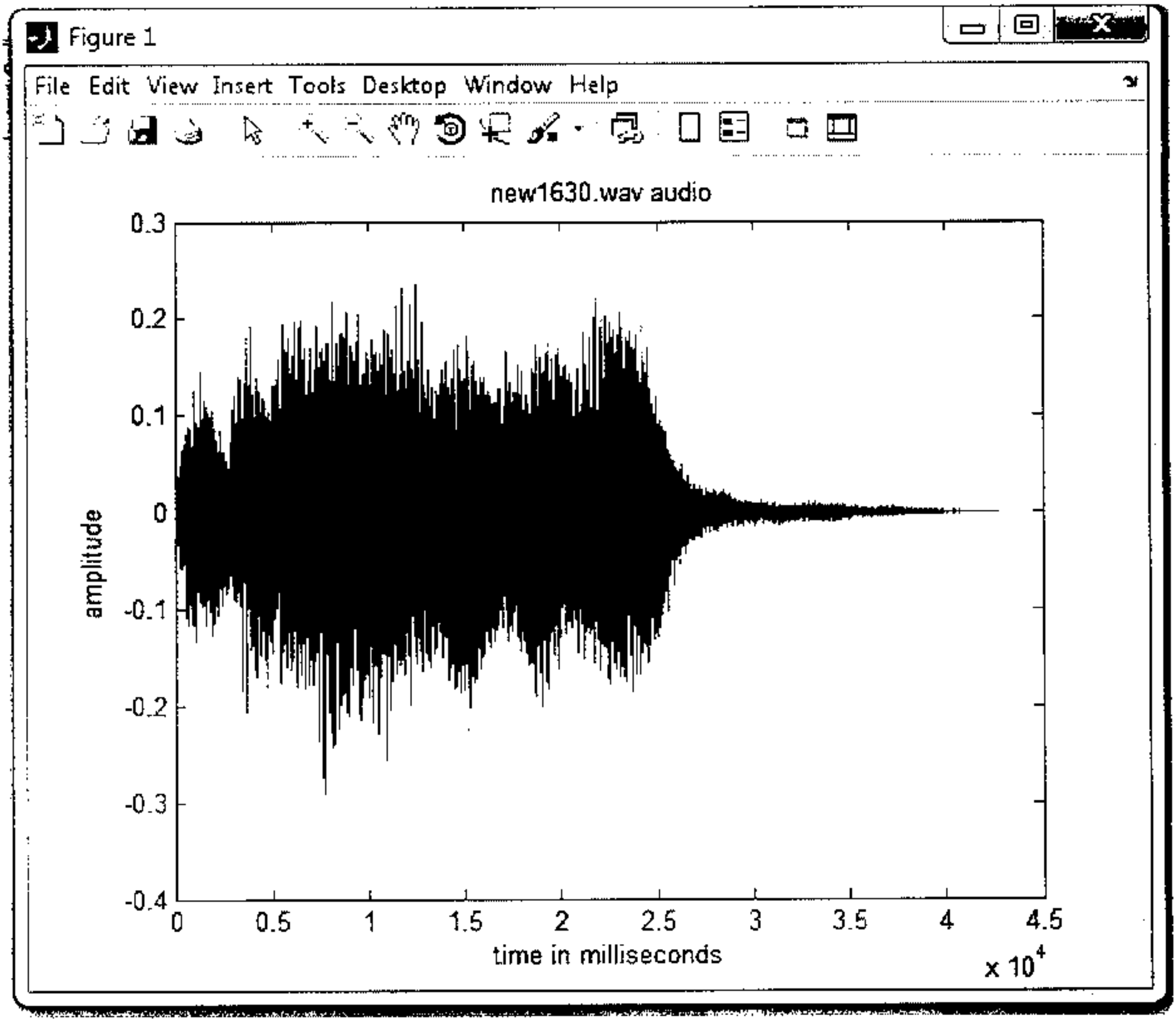


Figure 7.5 Simulation of Stego Audio File

Table 7.1 Peak Signal to Noise (PSNR) Analysis: new2 .wav

S.NO	Audio File (.wav)	No of text characters	Embedding time(min)	PSNR
1	new2	25	0.20	78.21
2	new2	50	0.45	59.68
3	new2	75	1.44	47.90
4	new2	100	3.23	39.47
5	new2	110	4.25	36.81
6	new2	120	5.39	34.25
7	new2	130	7.22	31.81
8	new2	140	8.30	29.74

Table 7.2 Peak Signal to Noise (PSNR) Analysis: xp .wav

S.NO	Audio File (.wav)	No of text characters	Embedding time(min)	PSNR
1	xp	25	0.19	86.34
2	xp	50	0.42	65.58
3	xp	75	1.38	53.42
4	xp	100	3.16	44.86
5	xp	110	4.07	42.18
6	xp	120	5.18	39.53
7	xp	130	6.38	37.10
8	xp	140	8.13	34.93

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The project describes the efficient method to embed the secret information(text data) into cover message(audio), without degrading the quality aspects of the cover audio file. With the help of Genetic algorithm it is possible to embed text data of 140 characters into the cover audio file. It is seen that the peak signal to noise ratio(PSNR) is improved as the text data characters selected for embedding reaches its maximum (140 characters). The time taken for embedding 140 characters into the cover audio file (new2.wav&xp.wav) is around 8 minutes and similarly the time taken for embedding 25 characters into the cover audio file (new2.wav&xp.wav) is around 20 seconds. The peak signal to noise ratio(PSNR) obtained for embedding 140 characters of secret text data into the new2.wav audio file is 29.74. The peak signal to noise ratio(PSNR) obtained for embedding 140 characters of secret text data into the xp.wav audio file is 34.93. It is found that the peak signal to noise ratio is inversely proportional to the embedding time. From the above results it is possible to perform secured communication between authorized users.

8.2 FUTURE SCOPE

The future scope of this project is to embed two bit in a chromosome sample and to increase the weight of the embedding bit. This technique can be tested in mp3 audio file also.

BIBLIOGRAPHY

[1]Robust audio steganography via genetic algorithm Zamani, M.; Taherdoost, H.; Manaf, A.A.; Ahmad, R.B.; Zeki, A.M.; Information and Communication Technologies, 2009. ICICT '09. International Conference on Digital Object Identifier Publication Year: 2009 , Page(s): 149 - 153

[2]Andersen, R.J., Petitcolas, F.A.P., On the limits of steganography. IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection 16 No.4 (1998) 47448 1.

[3] N. F. Johnson and S. Jajodia, "Steganalysis: The investigation of hidden information," in Proc. IEEE Inform. Technol. Conf., Syracuse, NY, 1998.

[4] P. Bmsb.I. Pitas, N. Nikoolaidis," Robust audio watermarking in the time domain," IEEE Transactions on Multimedia, vol3,2, June 2001

[5] Cvejic N. and Seppänen T. "Increasing the capacity of LSB based audio steganography", Proc. 5th IEEE International Workshop on Multimedia Signal Processing, St. Thomas, VI, December 2002, pp. 336-338.

[6] Pal S.K., Saxena P. K. and Mutto S.K. "The Future of Audio Steganography". Pacific Rim Workshop on Digital Steganography,Japan, 2002.

[7] Westfeld A. and Pitzmann A. "Attacks on Steganographic Systems". Lecture Notes in Computer Science, vol. 1768, Springer-Verlag, Berlin, pp. 61-75, 2000

[8] K. Gopalan, "Audio steganography using bit modification", Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Vol. 2, pp. 421- 424, April 2003.

[9] J. Zollner, H. Federrath, H. Klimant, et al., "Modelling the Security of Steganographic Systems", in 2nd Workshop on Information Hiding, Portland, April 1998, pp. 345-355.

[10] Matsuoka, H., "Spread Spectrum Audio Steganography using Sub – band Phase Shifting", Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'06), IEEE, 2006.

[11] An Introduction to Genetic Algorithms -Melanie Mitchell

[12]Genetic Algorithms in Search, Optimization, and Machine Learning David E. Goldberg

New heights old traditions

TAMILNADU COLLEGE OF ENGINEERING

COIMBATORE-641659

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEE

CERTIFICATE

This is to certify that Dr./Mrs./Mr.........MADHAN...RAJ....K.....;......P.G...

.KUMARAGURU...COLLEGE...OF...TECHNOLOGY.,..COIMBATORE.......... **has presented**

titled.STRENGTHENING...AUDIO..STENOGRAPHY...USING....GENETIC....ALGORITHM.............**aut**

............MADHAN....RAJ...:..K......AND........Ms....AMSAVENI...A..........................

the two day National Conference on Emerging Trends in Computer Comm

and Informatics - ETCCI 2011, technically co-sponsered by CIIT Internat

Journal held on 10th and 11th MARCH 2011.

Mrs.S.Latha Shanmuga Vadivu
Asst. Prof. ECE
CO-ORDINATOR

V. K
Mr.V.Karthikeyan
Lect. ECE
CO-ORDINATOR

Dr.M.Karthikeyan
HOD

Dr

[redacted]

Department of Electronics and Media Technology

NATIONAL CONFERENCE

on

Signals, Systems & Technologies in Media

This is to certify that

........................K. Madhan Raj...

presented a Paper titled ......Robustness....of...Audio...Steganog

.......Using.......Genetic.....Algorithms..

in the National conference held on 25th and 26th of Feb 2011.

G. L. Bala.

Dr.(Mrs) G.Joesmin Bala
CONVENER & HOD, EMT

Dr. Anne Mary Fernandez
REGISTRAR

Paul P. Ap

Dr. Paul P. A
VICE CHAN