

29-2-03

2003/2/29

**DATA HIDING IN WAVE MEDIA FILE  
BY USING  
STEGANOGRAPHY TECHNIQUES**

تعتمد كلية الدراسات العليا  
هذه النسخة من الرسالة  
التوقيع التاريخ

By  
**Doua A. Nassar**

Supervised by  
**Dr. Ahmed Aljaber**

**Submitted in Partial Fulfillment of the Requirement for the Degree of  
Master of Science in Computer Science**

**Faculty of Graduate Studies  
University of Jordan**

**May 2003**

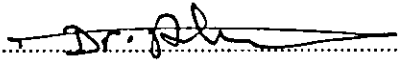
This thesis was successfully defended and approved

On:.....29/5/2003.

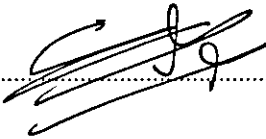
Examination Committee

عبد صفوان الرسالده  
Signature

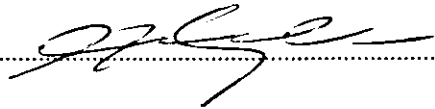
Dr. Ahmed Aljaber  
Assoc. Prof. Of Design Analysis of  
Computer Algorithms.

.....  


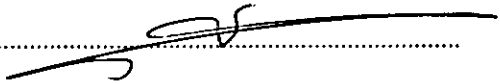
Dr. Moh'd Belal Al-Zoubi  
Assist. Prof. Of Graphics and Pattern  
Recognition.

.....  


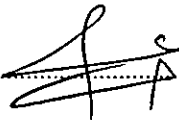
Dr. Riyadh Jabri  
Assoc. Prof. Of Compiler Design.

.....  


Dr. Emad Salah  
Assist. Prof. Of Networks and Complex  
Systems.

.....  


Prof. Dr. Saleh O'qeili  
Prof. Of Architecture.

.....  


## DEDICATION

*To soul of my father in law, Haj Abed Alrahman Hayajneh, grand father Haj Adel*

*Naif, and grand mother Hajeh Ameneh. May them souls rest in peace.*

*To my respected Parents, husband and loving kids*

## ACKNOWLEDGMENT

*In the completion of this thesis, I'm very grateful to Dr. Ahmed Al-Jaber Dean, King Abdullah II School for Information Technology, for an ever lasting memory during the course of my research. He guided me at every step and took the pains to go through my chapters more than once, although it was an arduous job yet Dr. Ahmed never disappointed me. Once again, I pay my heartfelt and sincere thanks to Dr. Ahmed Al-Jaber and his family.*

*Words stand helpless to thank those who delegated my life with roses and flowers, deeply, thanks to my parents and husband who scarifies a lot and gave a lot towards nothing, so what ever i give, still unable to express what they really deserve.*

*I'm also indebted to the Hashimeh University staff having granted me scholarship for pursuing my research work.*

*At the end, i would like to pay my thanks to all people help me in accomplishing this research especially, Dr. Lala for her big help and patient on me, colleagues in my office and all people forgeted to mentioned their names above.*

## TABLE OF CONTENTS

Subject	Page
Committee Decision	ii
Dedication	iii
Acknowledgement	iv
List of Contents	v
List of Tables	viii
List of Figures	ix
Abstract	xi
<b>1. GENERAL INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Literature Review	2
1.3 Aim of Thesis	4
1.4 Thesis Layout	4
<b>2. STEGANOGRAPHY TECHNIQUES</b>	<b>5</b>
2.1 Introduction	5
2.2 Steganography	5
2.3 The Terminology of Steganography	6
2.4 Types of Steganography	7
2.4.1 Pure Steganography	7
2.4.2 Secret Key Steganography	7
2.4.3 Public Key Steganography	8
2.5 Steganography in Digital Media	9

<b>Subject</b>	<b>Page</b>
2.5.1 Steganography in Text	9
2.5.2 Steganography in Image	11
2.5.3 Steganography in Audio	13
2.6 Steganography attackers.	15
2.7 Sound Fidelity Criteria Computation	17
3. AUDIO MEDIA FUNDAMENTAL AND ENVIRONMENT	19
3.1 Introduction	19
3.2 Mechanics of Sound	19
3.2.1 The Decible	21
3.3 Digital Audio and Sound Manipulation	21
3.4 Wave File Format	24
4. WAVE BASED STEGANOGRAPHY SYSTEM	28
4.1 Introduction	28
4.2 Wave Based Steganography System Theoretical Concept.	28
4.2.1 Discrete Cosine Transformation (DCT)	28
4.2.2 Differential Pulse Code Modulation	30
4.2.3 Mediator Amplitude Values Technique (MAV)	30
4.3 Wave Based Steganography System (WBSS)	30
4.3.1 Wave Differential Technique	31
4.3.1.1 Embedding Average Amplitude Algorithm	33
4.3.1.2 Extracting Average Amplitude Algorithm	35
4.3.2 Wave DCT Technique	37
4.3.2.1 DCT-Based Embedding Algorithm	39
4.3.2.2 DCT-Based Extracting Algorithm	42

<b>Subject</b>	<b>Page</b>
4.4 WBSS Experiment Results	44
4.4.1 Wave Differential Results	44
4.4.2 Wave DCT Results	54
5. DISCUSSION, AND FUTURE WORKS	65
5.2 Discussion	65
5.4 Future Work	70
6. REFERENCES	71
Appendix A	74
Appendix B	75
Abstract in Arabic	94

## LIST OF TABLES

<b>Subject</b>	<b>Page</b>
Table (3.1) Sources Frequencies.	20
Table (3.2) The Quantization (integer rounding) using Sign.	23
Table (3.3) Native Wave File Format.	25
Table (3.4) Contents of fmt Chunk for PCM data.	26
Table (4.1a) The content of Stego Key	32
Table (4.1b) The content of Stego Key	38
Table (4.1) The Wave Cover Media Properties.	44
Table (4.2) The Stego Key.	46
Table (4.3) The Embed message differences data.	47
Table (4.4) The comparison between proposed technique and LSB.	48
Table (4.5) The Wave Cover media properties.	49
Table (4.6) The Stego Key.	51
Table (4.7) The Embed message differences data.	52
Table (4.8) The comparison between proposed technique and LSB.	53
Table (4.9) The Wave Cover Media Properties.	54
Table (4.10) The Wave Embedded Media Properties (Speech).	56
Table (4.11) The Stego Key.	57
Table (4.12) The Wave Embedded Media properties (Normal).	59
Table (4.13) The Stego Key.	60
Table (4.14) The Wave Embedded media properties (Uniform).	62
Table (4.15) The Stego Key.	63
Table (4.16) Results of DCT technique	64



## LIST OF FIGURES

<b>Subject</b>	<b>Page</b>
Figure (2.1) The Hiding Model	6
Figure (3.1) A sin Wave	19
Figure (3.2) Conversion from analog signal to PCM digital code	22
Figure (3.3) Quantized PAM signal	22
Figure (3.4) PCM	23
Figure (3.5) Nesting of chunk in a RIFF file	24
Figure (3.6) RIFF Chunk format	25
Figure (3.7) Structure of 8-bit mono wave file	27
Figure (3.8) Structure of 16-bit mono file	27
Figure (3.9) Structure of 8-bit stereo wave file	27
Figure (3.10) Structure of 16-bit stereo wave file	27
Figure (4.1) The Wave Cover media	45
Figure (4.2) The Wave Cover Spectrum	46
Figure (4.3) The Stego Wave media	47
Figure (4.4) The Stego Wave media	48
Figure (4.5) The Wave Cover media	49
Figure (4.6) The Wave Cover Spectrum	50
Figure (4.7) The Embedded Image	51
Figure (4.8) Histogram of Embedded Image	51
Figure (4.9) The Stego Wave Media.	52

<b>Subject</b>	<b>Page</b>
Figure (4.10) The Spectrum of Selected Cover Wave Media	53
Figure (4.11) The Cover Wave Media	55
Figure (4.12) The Cover Wave Media Spectrum	55
Figure (4.13) The Embedded Wave Media	56
Figure (4.14) The Spectrum Analysis of Embedded Wave Media	57
Figure (4.15) The Stego Wave Media	58
Figure (4.16) The Spectrum of Stego Wave Media	58
Figure (4.17) The Embedded Wave Media	59
Figure (4.18) The spectrum of Embedded Wave Media	60
Figure (4.19) The Stego Wave Media	61
Figure (4.20) The Spectrum of Stego Wave Media	61
Figure (4.21) The Samples Distribution of Uniform Audio Media	62
Figure (4.22) The Spectrum of Uniform Audio Media	63
Figure (4.23) The Stego Wave Media	63
Figure (4.24) The Stego Wave Media	64
Figure (5.1) The Cover wave spectrum	66
Figure (5.2) The stego wave spectrum (jump=2)	66
Figure (5.3) The stego wave spectrum for distributed embedded media	66

## LIST OF ABBREVIATIONS

A/D	Analog-to-digital.
D/A	Digital-to-analog.
db	Decibel.
DCT	Discrete Cosine Transform.
DPCM	Differential Pulse Code Modulation.
HAS	Human Audio System.
IFF	Interchange File Format.
ISDN	Integrated Service Digital Network.
JPEG	Joint Photographic Experts Group.
LSB	Least Significant Bit.
MAV	Mediator Amplitude Values Technique.
PAM	Pulse Amplitude Modulation.
PCM	Pulse Code Modulation.
RIFF	Resources Interchange File Format.
WAV	Windows AudioVisual.
WBSS	Wave Based Steganography System.
WDT	Wave Differential Technique.

**DATA HIDING IN WAVE MEDIA FILE  
BY USING  
STEGANOGRAPHY TECHNIQUES**

**By  
Doua A. Nassar**

**Supervisor  
Dr. Ahmed Aljaber**

**ABSTRACT**

Transforming the information across the world in a secure form has become a challenge to many experts. This resulted in the development of different disciplines of information hiding, one of which is steganography. Steganography encompasses methods of transmitting secret messages through an innocuous cover in an effort to conceal the existence of secret data

In this thesis, a new approach, Wave based Steganography System (WBSS) is proposed. This system consists of two techniques. Firstly the Wave Differential Technique (WDT). It is used to decrease the limitation and complexity of embedding huge amount of data (Text, Image) into Wave cover media. The second technique is used to embed and extract wave into Wave cover media.

The proposed system gives good results for PSNR, MAE, MSE, and an amount of embedded data that exceeds half of Wave cover media size. In addition, the result of this system shows that the proposed system is applicable in practice with high reliability.

## 1. GENERAL INTRODUCTION

### *1.1 Introduction*

Hiding information is one of the most important ways to keep something invisible. This depends on finding the best place to hide an object in a way that can't be recognized by others. In the same time, it must not affect the place where the object is kept. That is called "Steganography".

Digital watermark and steganography are two areas related to information hiding. The last few years have seen rapidly growing interest in ways to hide information in order to support ownership protection, copy control, annotation and authentication.

Steganography is an old word derived from Greek, which means, "covered writing" and it keeps the secrecy like the cryptography. Cryptography differs from steganography because it changes the secret object to a new form that cannot be read by the others. Just for people who know the rules for re-changing the forms (characters) to the original ones. Also it differs from Watermarking. Digital watermarks involve information embedded techniques that convey some information about the carrier.

Since watermarks are embedded in more significant areas of digital media. Watermarking techniques may be applied without fear of image destruction. This is due to lossy compression. In some cases digital watermarking is primarily one of intent.

The goal of steganography is to avoid drawing suspicion to the transmission of a secret message. Kuhn said, "The goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present". (Cochran, J.T., 2000).

The amount of hidden information is another steganography problem. Therefore, new techniques are required to avoid this problem.

## ***1.2 Literature Review***

Throughout history, people have hidden information by multifarious methods. For example, ancient Greek text was written on wax covered tablets. In one story Demeratus wanted to notify Sparta that Xerxes intended to avoid Greece. To avoid capture, he scraped the wax off of the tablets and wrote a message on the underlying wood. He then covered the tablets with wax again. The tablets appeared to be blank and unused so they passed inspection by sentries without question (Neil, F.J. and et. al., 2001) (Neil, F.J., 1999).

Histiaeus did another ingenious method. He wished to inform his friends that it was the time to begin revolt against the Medes and the Persians. He shaved the head of the most trusted slave and tattooed a message on the head, waited till his hair grew back, and sent him along. The message would be undetected until the head was shaved again (Neil, F.J. 1999).

In Tudor England, when Mary Queen of Scots was imprisoned at Chartly Castle, she sent secret messages to the Catholics including the French Ambassador, by hiding the letters inside the empty beer barrels that left the castle (Neil, F.J., 1999).

Another common form of invisible writing is through the use of invisible inks. The Germans also developed covert communications by developing microdot technology. (Neil, F.J. and et. al., 2001).

T. Clelland et al (1999), proposed a new technique called " genomic steganography ". This technique is used to encode a hidden message in a strand of human DNA. Different combinations of amino bases or nucleotides represented the

letters of the message. Additional sequences of amino bases are added to this strand to serve as a "key" to finding the strand containing the embedded message.

C. Christian (1998), proposed an information-theoretic model for steganography with passive adversaries. The adversary's task of distinguishing between an innocent cover message  $C$  and a modified message  $S$  containing a secret part, is interpreted as a hypothesis-testing problem.

J. Cox et al (1996), presented an image watermarking method in which the mark is embedded in the most perceptually significant frequency components  $V=\{r_i\}_{i=1}$  of an image's DCT to provide greater robustness to JPEG compression.

S. Mitchell et al (1998), presented a robust audio watermarking procedure using perceptual masking. This procedure embeds copyright protection into digital audio by directly modifying the audio samples.

A. Westfeld and G. Wolf (1998), described a system that transmit messages in a lossy DCT-based video compression scheme over an ISDN line. This is used for video-conference. Up to 8 kilobits could be embedded without degrading the signal to the point that the secret communication becomes apparent.

D. Gruhl et al (1996), proposed a novel transform coding technique called "echo hiding". This technique relies on the fact that we cannot perceive short echoes (of the order of a millisecond). It embeds data into a cover audio signal by introducing two types of short echo with different delays to encode zeros and ones. These bits are encoded at location separated by spaces of pseudo random length.

In this thesis, we propose a system based on Discrete Cosine Transformation (DCT) and Differential Pulse Code Modulation (DPCM) Techniques. This system together with its results will be discussed in chapter four.

### ***1.3 Aim of Thesis***

The aim of this thesis is to design an information hiding system to embed a message (Text, Image, and Wave) into wave media file in such a way, that there would be no perceivable changes in the audio file after the message insertion. At the same time, if the message that is to be hidden were encrypted, the level of security would be raised to a satisfactory level.

Amount of hiding information is a steganography problem. Therefore, new techniques are proposed and implemented to embed a large amount of information. These techniques handle the amount of hiding information and the security by replacing some of bits in the cover wave media with new bits from hidden information.

### ***1.4 Thesis layout***

The remaining chapters of the present study will be planned as follows. Chapter Two represents the complete discussion to the information hiding, and the most commonly used definitions for steganography and digital watermark. It explores different types of steganography and the media that are most frequently used to hide data in (text, image, and audio). Chapter Three includes a general introduction to the audio media, its features, its environments, and a view for the structure of the windows audio visual format (.Wav) files.

Chapter Four is dedicated for the proposed hiding system for embedding data in wave media files, all ideas, and algorithm(s) used for hiding and extracting processes. Finally, Chapter Five deals with the suggestion for future work as a point represented by conclusions and recommendations.



## 2. STEGANOGRAPHY TECHNIQUES

### 2.1 Introduction

Steganography is an ancient art of hiding information. Digital technology gives us new ways to apply steganographic techniques.

Steganography can be classified into three types: pure, secret key, and public key steganography.

Today steganography is applied in most media that deal with computers, like text, image, audio, and other like unused area network packets.

### 2.2 Steganography

Steganography is a technique to make confidential messages imperceptible to human eyes by using some other data like an image. The data, which hides the secret message, is called "Vessel", "Carrier", "Container", or "Dummy data" of the secret message. It looks innocent, and attackers can not see anything to attack.

Therefore, steganography is more an "information imperceptualizing technique" than an "information hiding technique". Encryption of the embedded data would further improve security. This scenario is analogous to putting something in a very secure safe and then hiding this data in a way it's hard to find its places [Neil, F. J., 1999].

Other alternative definitions are given as follows:

- Steganography is the art of passing information in a manner that the very existence of the message is unknown [Johnson, N. F. and Jajodia S., 1998].
- Steganography encompasses methods of transmitting secret messages through innocuous cover carriers in such a manner that the very existence of the embedded message is undetectable [Johnson, N. F. and Jajodia S., 1998].

579071

### 2.3 The Terminology of Steganography

The general model for hiding information in other information can be described as in Figure (2.1):

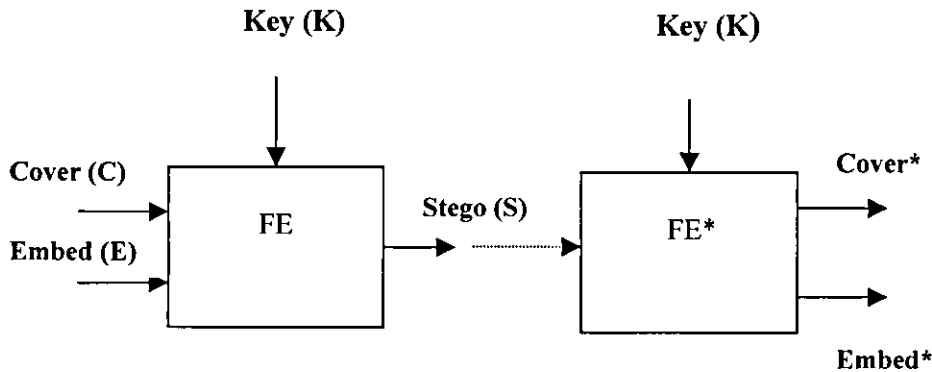


Fig. (2.1) *The Hiding Model*

Where:

FE: is the steganographic function "Embedding",

FE\*: Steganographic function "Extracting",

Cover (C): is the cover data in which "Embed" will be hidden.

Embed (E): is the message to be embedded.

Key (K): is the parameter of FE & FE.

Stego (S): is the cover data with embedded message.

The embedded data is the message that one wishes to send secretly. It is usually hidden in an innocuous message referred to as a cover-text, a cover-image or a cover-audio, producing the stego-text or other stego-object. A stego-key is used to control the hiding process so as to restrict detection and/or recovery of the embedded data to parties who know it [Petitcolas, F.A, and et. al, 1999].

## 2.4 Types of Steganography

Steganography is classified into three main types. These types are: Pure, secret key, and public and private steganography.

### 2.4.1 Pure Steganography

Pure steganography is a system, which does not require the preceding exchange of some private information (like a stego-key). The mapping of embedding and extracting process can be described as follows:

$$F_E: C \times E \rightarrow C$$

$$F_E^*: C \rightarrow E$$

The extracting and embedding algorithms should be private. The sender and receiver must have passage to these algorithms.

**Definition** [Smith, A., 1998]: -

*The quadruple  $\mathcal{S} = (C, E, F_E, F_E^*)$  where  $C$  is the set of possible covers,  $E$  is the set of the secret messages with  $|C| \geq |E|$ .*

*$F_E: C \times E \rightarrow C$  the embedding functions.*

*$F_E^*: C \rightarrow E$  extraction function,*

*With the property that  $F_E^*(F_E(c, e)) = e$ , for all  $e \in E$  and  $c \in C$  is called Pure Steganography.*

### 2.4.2 Secret key steganography

In this system the sender uses the secret key  $K$  to embed the secret message into elected cover  $C$ . If the key used in the embedding process is known to the receiver, he can reverse the process and extract the secret message. The cover  $C$  and the stego-object can be intellectually similar.

**Definition** [Smith, A., 1998]:

The quintuple  $\mathcal{S} = \langle C, E, K, F_{EK}, F_{EK}^* \rangle$  where  $C$  is the set of possible covers,  $E$  the set of secret messages with messages  $|C| \geq |E|$ ,  $K$  the set of secret Keys.

$$F_{EK}: C \times E \times K \longrightarrow C$$

$$F_{EK}^*: C \times K \longrightarrow E,$$

With the property that  $F_{EK}^*(F_{EK}(c, e, k), k) = e$  for all  $e \in E$ ,  $c \in C$  and  $k \in K$  is called a Secret Key Steganography.

### 2.4.3 Public key steganography

Public key steganography does not rely on the exchange of a secret key. Public key steganography system requires the use of two keys, one private and one public key. The public key is stored in a public database. The public key is used in the embedding process. The secret key is used to reconstruct the secret message [Smith, A., 1998].

Public key steganography handle the fact that the decoding function  $F_{EK}^*$  in a steganography system can be applied with any cover  $C$ , whether or not it already contains a secret message (recall that  $F_{EK}^*$  is a function on the entire set  $C$ ). A protocol, which allows public key steganography had been proposed by Anderson [Smith, A., 1996]. It relies on the fact that encrypted information is random and enough to "hide in plain sight".

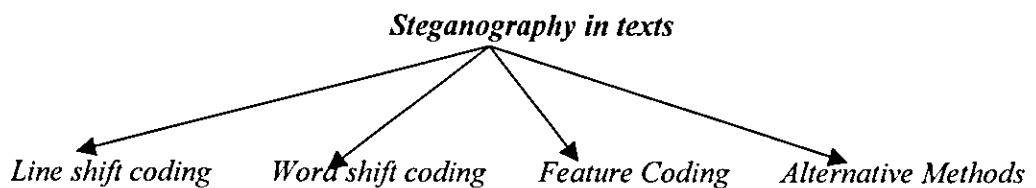
## ***2.5 Steganography in Digital Media***

The improvement of computer technology and the Internet have given a new life to steganography and creative methods with which it is employed computer-based steganographic techniques. It introduces changes to digital carriers to embed information foreign to the native carriers. Since 1995 interest in steganographic methods tools as applied to digital media exploded [Neil, F.J., and et. al. 2001].

Today steganography is applied in most media that dealing up with computer, like: text, image, audio, and unused area network packet. This will be discussed in the following points.

### ***2.5.1 Steganography in texts***

Data hiding in text is an exercise in the discovery of modifications that are not noticed by readers. Four major methods of encoding data will be considered in the following sections.



#### ***- Line-shift Coding***

In this method, text lines are vertically shifted to encode the document uniquely. Encoding and decoding can generally be applied either to the format file of a document, or the bitmap of a page image.

### ***- Word-shift Coding***

Codewords are coded into a document by shifting the horizontal locations of words within text lines, while maintaining a natural spacing appearance. This encoding can also be applied to either the format file or the page image bitmap.

### ***- Feature Coding***

A third method has been suggested by Brassil et al [Grul, B.W.;and et. al.,1996]. Certain text features are altered, or not altered, depending on the codeword. For example, one could encode bits into text by extending or shortening the upward, vertical end lines of letters such as b, d, h, etc. Generally, before encoding, feature randomization takes place. That is, character end line lengths would be randomly lengthened or shortened, then altered again to encode the specific data.

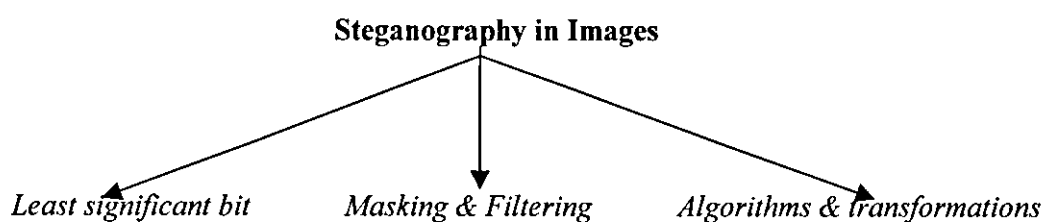
### ***-Alternative Methods***

Bender, et al., provides interesting text-coding methods alternative, in [Grul, B.W.;and et. al.,1996]. He suggested three major methods of encoding data:

1. Open space methods, similar to the ones suggested by Brassil.
2. Syntactic methods that utilize punctuation and contractions.
3. Semantic methods, that encode using manipulation of the words themselves.

## 2.5.2 Steganography in Images

Information can be hidden in many different ways in images. The most common approaches to information hiding in images will be considered in the following sections.



### - *Least Significant bit Insertion (LSB)*

It is the most well known image steganography technique. It is a common, simple approach to embedding information in a graphical image file, and it is extremely easy to attacks. When applying LSB techniques to each byte of a 24-bit image, three bits can be encoded into each pixel, as each pixel is represented by three bytes. For example, the letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

The binary value for the letter A is (10000011). Inserting the binary value of A into the three pixels, starting from the top left byte, would result in:

(00100111 11101000 11001000) (00100110 11001000 11101000)

(11001000 00100111 11101001)

### ***- Masking and filtering***

This technique hides information by marking an image in a manner similar to paper watermarks (applied without fear of differences in cover). It is done by using a cover or a mask of signal with another one to make the first non-perceptible. This uses the fact that the human visual system cannot detect changes in image.

Masking techniques are more suitable for use in lossy JPEG images than LSB insertion because of their relative immunity to image operations such as compression and cropping [[www.An Introduction to Steganography.com](http://www.An Introduction to Steganography.com)].

### ***- Algorithms and Transformations***

Transformation is a process, which takes information in one domain and expresses it in another. Images are in spatial domain and they transformed to frequency domain. Compression uses transformation, Fast Fourier transformation (FFT), and Discrete Cosine transformation are some of them.

JPEG images use the discrete cosine transform (DCT) to achieve compression. DCT is a lossy compression transform, because the cosine values cannot be calculated precisely. Rounding errors may be introduced as well. Variances between the original data and the recovered data depend on the values and methods used to calculate the DCT.

Images can also be processed using Fast Fourier transformation (FFT) and wavelet transformation. Other properties such as luminance can also be utilized.

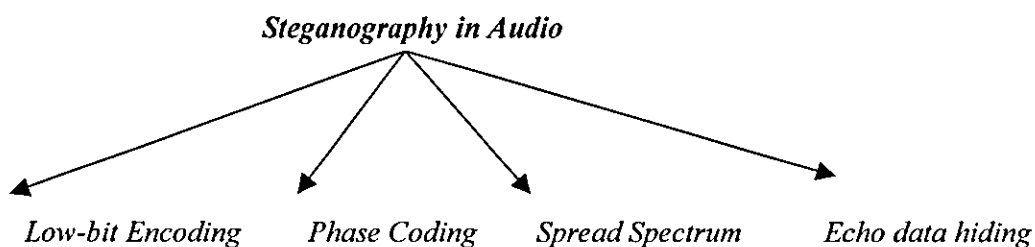


### 2.5.3 Steganography in Audio

Our hearing sense could not recognize all voices and noises that are accompanied with original wave media. Data hiding in audio signals is especial challenge, because the human auditory system (HAS) operates over a wide dynamic range. The HAS perceives over a range of power greater than one billion to one and a rang of frequencies greater than one thousand to one. Sensitivity to additive random noise is also acute [Grul, B.W.;and et. al.,1996].

When performing data hiding on audio, one must exploit the weaknesses of the HAS, while at the same time being aware of the extreme sensitivity of the human auditory system [Tanenbaum, A. S., 1996].

There are different methods of data hiding in audio, consider some of these methods in following sections.



#### **- Low-bit Encoding**

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

Crowd noise during a live event would mask low-bit encoded noise that would be audible in a string quartet performance.

The major disadvantage of this method is poor immunity to manipulation; channel noise, resembling, etc. can destroy the encoded information, unless it is encoded using redundancy techniques. In order to be robust, these techniques reduce the data rate, often by one to two orders of magnitude.

In practice, this method is useful only in physical storage and closed digital-to-digital environment [Grul, B.W.;and et. al.,1996].

### ***- Phase Coding***

The phase coding method works by substituting the phase of an initial audio segment with a reference phase that represents the data.

The first step in this coding is to divide sound into series of  $N$  short segment. Then construct a matrix of magnitude and phase by applying Discrete Fourier transform (DFT) on each segment. After that you have to calculate phase difference between each adjacent segment is calculated. For the first segment  $S_0$ , an artificial absolute phase  $P_0$  is created. For the rest of the segments, new phase frames are created. The new phase and the original magnitude are combined to get a new segment,  $S_n$ . Finally; the new segments are concatenated to create the encoded output.

For the decoding process, the synchronization of the sequence is done before the decoding. The length of the segment, the DFT coefficient, and the data interval must be known at the receiver. The value of the underlying phase of the first segment is detected as 0 or 1, which represents the coded binary string [Grul, B.W.;and et. al.,1996].

### **- Spread Spectrum**

Most communication channels try to concentrate audio data in as narrow a region of the frequency spectrum as possible in order to conserve bandwidth and power. When using a spread spectrum technique. However, the encoded data is spread across as much of the frequency spectrum as possible.

One particular method discussed in Bender et. al [Grul, B.W.;and et. al.,1996] is Direct Sequence Spread Spectrum (DSSS) encoding. It spreads the signal by multiplying it by a certain maximal length pseudo random sequence, known as a *chip*. The sampling rate of the host signal is used as the *chip rate* for coding. The calculation of the start and end quanta for phase locking purposes is taken care of by the discrete, sampled nature of the host signal. As a result, a higher chip rate and therefore a higher associated data rate, is possible. However, unlike phase coding, DSSS does introduce additive random noise to the sound [Grul, B.W.;and et. al.,1996].

### **- Echo data hiding**

Echo data hiding embeds data into a host signal by introducing an echo. Varying three parameters of the echo hides the data: initial amplitude, decay rate, and offset, or delay. As the offset between the original and the echo decreases, the two signals blend. At a certain point, the human ear cannot distinguish between the two signals, and the echo is merely heard as added resonance. This point depends on factors such as the quality of the original recording, the type of sound, and the listener.

By using two different delay times, both below the human ears perceptual level, we can encode a binary one or zero. The decay rate and initial amplitude can also be adjusted below the audible threshold of the ear. This is needed to ensure that the information is not perceivable. To encode more than one bit, the original signal is

divided into smaller portions, each of which can be echoed to encode the desired bit. The final encoded signal is then just the recombination of all independently encoded signal portions.

As a binary one is represented by a certain delay  $y$ , and a binary zero is represented by a certain delay  $x$ . Detection of the embedded signal then just involves the detection of spacing between the echoes. A process for doing this is described in Gruhl, et al.'s work. [Gruhl, B.W.;and et. al.,1996]

Echo hiding was found to work exceptionally well on sound files where there is no additional degradation, such as from line noise or lossy encoding. It is also used when there are no gaps of silence. Work to eliminate these drawbacks is being done [Gruhl, B.W.;and et. al.,1996].

## ***2.6 Steganography attackers***

Any steganography system needs three steps to be broken, detecting, extracting and disabling embedded information. If the attacker can prove the existence of a secret message the system will be unsecured.

In the world of attacks, there are three types of attackers, passive, active and malicious attacker.

- Passive attacker: attacker has to decide whenever a cover  $C$  sent contains secret information or not.
- Active attacker: attacker who are able to change the cover during the communication process and he is not able to extract or prove the existence of a secret message.
- Malicious attacker: attacker who is able to forge message if the embedding method is not dependent on some secret information shared by sender and receiver.

Mean absolute error (MAE) can measure the quality to the difference of a reconstructed image compared with an original image. The value of this measure is between 1 and 0, the actual value is good if the value near from zero.

$$MAE = \frac{\sum |f(i, j) - F(i, j)|}{N^2} \quad (2.3)$$

For our computations, the equations (2.1, 2.2 and 2.3) are used to compute the fidelity of wave files. However the wave signals are not different from the image signals.

### 3. AUDIO MEDIA FUNDAMENTAL AND ENVIRONMENT

#### 3.1 Introduction

By our Human Audio System (HAS), we can recognize all the different sounds around us, and most of the voices for our family and friends. This is because of the sensitivity of hearing received by the ear.

Sound is a continuous wave that travels through the air. The wave is made up of pressure differences. Sound is detected by measuring the pressure level at a location.

So to get audio into a computer, we must digitize it (convert it into a stream of numbers). In this chapter we will find how to convert the analog audio into digital, and the format of the Wave file which is the type of Audio we will use it.

#### 3.2 Mechanics of Sound

The simplest musical sound is elicited when a tone producing object vibrates backwards and forwards exhibiting what physicists call simple harmonic motion. When an object vibrates in this way it follows the path traced out. Such a motion is known as sinusoidal and the trace is known as a sine wave. [Brice, R., 1997].

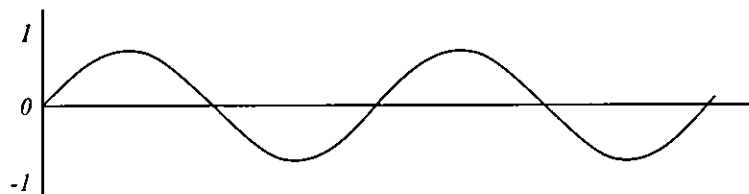


Figure (3.1) *A sine wave*

A sine wave is a periodic wave. This means that its first pulse is followed by an indefinite number of identical pulses. Periodic wave results in sounds that can be called tones, such as the tone of the guitar, the piano or the bell [Stolz, A, 1993].

The human ear is capable of perceiving sound between 20 Hz and 20000 Hz. However this range varies considerably from person to another, particularly the high frequency limits. The ability to hear these upper frequencies decreases with age. Table (3.1) shows the range of various sound sources [Stolz, A., 1993]

*Table (3.1) Sources frequencies*

<b>Instrument</b>	<b>Frequency range</b>
<b>Human voice</b>	70-2000 Hz
<b>Pipe organ</b>	16-4000 Hz
<b>Piano</b>	30-3500 Hz
<b>Violin</b>	200-3000 Hz
<b>Flute</b>	260-3000 Hz

In other words, the dynamic range of hearing is so wide as to be up to the fundamental physical limitation [Brice, R., 1997].

To create any wave sound :

$$y = A \sin 2\pi ft \quad \dots(3.1)$$

Where:

A = represent the wave amplitude values

f = represent the frequency, which is calculate as (1/duration).

t = represent the time.

The primary element of a wave is its strength or amplitude. The highest point along the curve of the sound wave determines the amplitude. The higher the amplitude, the louder the sound will be. The physical unit of loudness is the decibel (dB).

### 3.2.1 The Decibel (dB)

A decibel is an algorithmic unit of measuring specifying the degree of loudness of sound wave. Because of the vast dynamic range of the human ear, it is more convenient to measure sound intensity by 10 (logarithms) than by numbers on a linear scale [Alan, A.,C.,1989].

The equation (3.2) shows how we can convert linear scale wave data (amplitude) to decibel (dB):

$$decibel = 20 \log_{10} \left( \frac{Amp}{8^5} \right) \dots\dots\dots(3.2)$$

Where:

Amp = stands for the wave sample amplitude value.

$8^5$  = stands for the maximum possible amplitude value (32768) and referred to as 0 decibel, or 0 dB.

### 3.3 Digital audio and sound manipulation

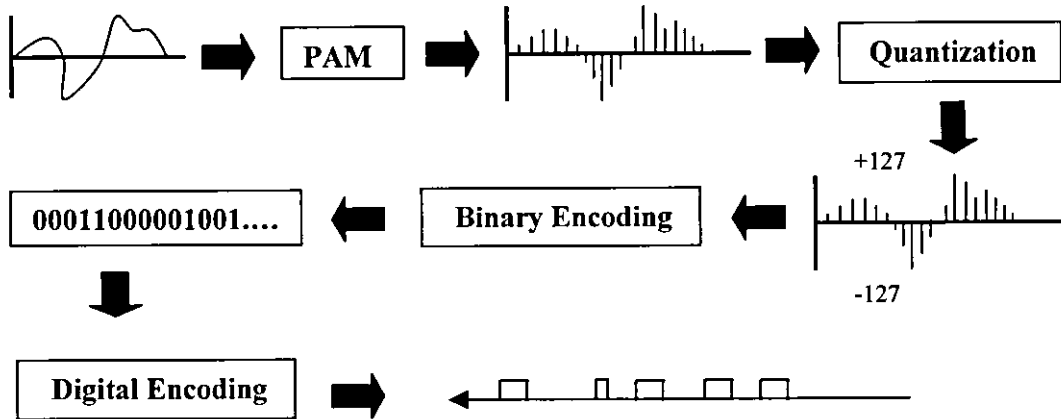
Sound is air pressure fluctuation. Digitized sound is a graph of the change in air pressure over time. Effected media like (microphone, instruments, machines, etc) generates several waves of sound in analog form. Analog-to-digital converter (ADC) changes the generated wave into digital form.

Analog-to-digital encoding (A/D) is the representation of analog information by a digital signal. To do so you need to reduce the potentially infinite number of values in an analog message so that they can be represented as a digital stream with a minimum loss of information.



There are several methods for analog-to-digital encoding. The most popular are the PAM, *Pulse Amplitude Modulation* & PCM (*Pulse Code Modulation*).

Figure (3.2) shows the conversion from analog to digital code.

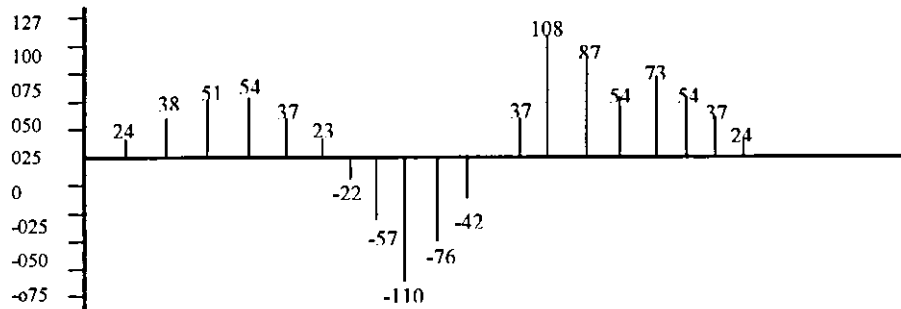


*Figure (3.2) Conversion from analog signal to PCM digital code*

The first step in analog-to-digital encoding is called pulse amplitude modulation (PAM). This technique takes analog information, sampling it, and generates a series of pulses these pulses are still of any amplitude. Thus we must modify them by using pulse code modulation (PCM).

PCM modifies the pulses created by PAM to create a completely digital signal, to do so, PCM first quantize the PAM pulses.

Quantization is a method of assigning integral values in specific range to sampled instances. Figure (3.3) shows the result of quantization is presented in.



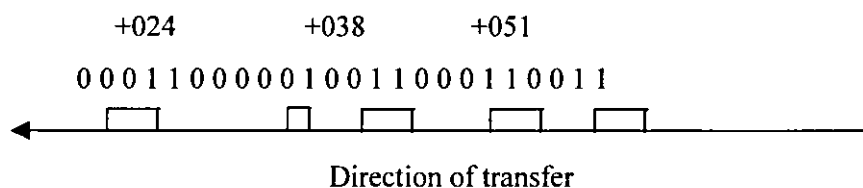
*Figure (3.3) Quantized PAM signal*

After finishing quantization, a simple method of assigning sign and magnitude values to quantization samples. Each values is translated into its seven-bit binary equivalent, the eighth bit indicates sign. Table (3.2) shows the quantization using sign.

**Table (3.2) The quantization (integer rounding) using sign**

Actual value	Rounded value	Binary representation
23.73	24	00011000
38.27	38	00100110
50.94	51	00110011
53.82	54	00110110

The binary digits are then transformed into a digital signal using one of the digital-to-digital encoding techniques.



**Figure (3.4) PCM**

Figure (3.4) shows the result of the pulse code modulation of the original signal encoded finally into a digital signal. The most popular format for representing samples of high-quality digital audio is a 16-bit linear quantization (e.g. Windows Audio-Visual WAV) and Audio Interchange File Format AIFF.

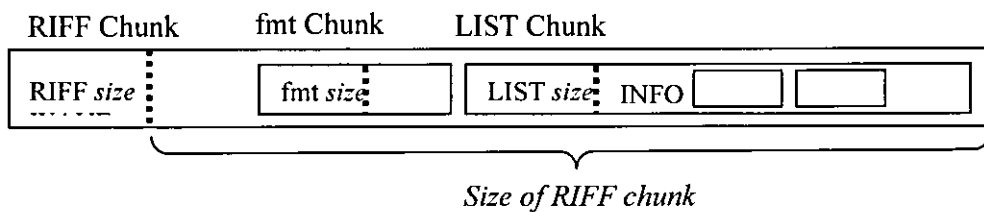
### 3.4 Wave file format

Wave file is the native sound format used by Microsoft Windows. It is sort of the *Interchange File Format* (IFF). IFF was originally developed by *Electronic Arts* to be used on the Commodor Amiga. These files contain a single image, formatted text, animation, sound, or any combination of different types of data. Microsoft defined a general file format called the *Resource Interchange File Format* (RIFF).

Because wave files are a special type of RIFF file, we will discuss the basics of RIFF.

A RIFF file consists of a collection of nested chunks, as illustrated in Figure (3.5), each chunk contains a four-character code (such as RIFF, fmt\_, or LIST; shorter codes are padded with space). Each code indicating the type of chunk, for example, an *fmt\_chunk* contains information about the format of a sound.

Following the chunk type is a 4-byte size value indicating the size of the data carried by the chunk, note, for instance, that the entire file illustrated in Figure (3.9) is a single RIFF chunk [Kientzle, T.,1998].



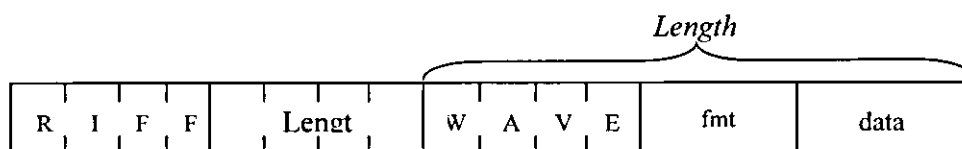
*Figure (3.5) Nesting of chunk in a RIFF file*

The size field in the RIFF chunk is exactly 8 bytes less than the total file size because the chunk type and size are not included in the count.

The data of a container chunk starts with a four-character code indicating the type of data contained within that chunk.

The internal chunk names may correspond to different things, depending on the surrounding container, because this fmt chunk is contained with a RIFF wave container, it might contain different information.

In wave files, the outermost chunk is a RIFF container with a wave container type. Most wave files contain both an fmt chunk and a data chunk, as shown in Figure (3.6) [Kientzle, T.,1998].



**Figure (3.6) RIFF Chunk format**

- **Header of Wave File**

Because So many wave files have this same basic structure, many native program treat wave file as having a fixed header with the format shown in table (3.3).

**Table (3.3) Native wave file format**

Size	Description
4	Chunk type :RIFF
4	Total file size minus 8
4	RIFF contains type:Wave
4	Chunk type: fmt_
4	Format chunk data length: usually 16
16	Format chunk data
4	Chunk type: data
4	Length of sound data (No. Of samples in file)
n	Actual sound samples

- **The *fmt* Chunk**

The *fmt* chunk contains the actual sound format information. The precise contents of the *fmt* chunk vary depending on the compression method. Table (3.4) shows the format used for plain PCM data [Kientzle, T.,1998].

**Table (3.4) Contents of *fmt* chunk for PCM data**

Size	Description
2	Compression code
2	Number of channels (Stereo or mono)
4	Samples per second (sample rate)
4	Average number of byte per second
2	Block alignment (8, or 16 bit sampling)
2	Number of bytes of additional information
n	Additional compressor-specific information

- **Sample rate:** number of samples per second, The most popular temporal sampling rates for audio records include 8 kHz, 11.024 kHz, 22.05 kHz, and 44.1 kHz.

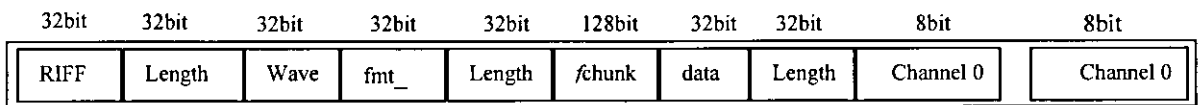
- **The *Data* Chunk**

The data chunk block starts after the 32-bit length value. The data chunk stores the actual compressed sound data [Kientzle, T.,1998]. The information from *fmt* chunk are needed for decoding the information contained in this chunk. Therefore, every wave file must always contain at least both of these chunks. However, other chunks can also be added to the file [Stolz, A,1993]. In particular, the single data chunk is sometimes replaced by a LIST container, which contains *Slant* chunks (indicating silent intervals) and data chunks with sound data.

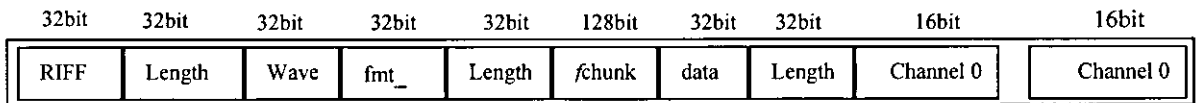
The Sound Blaster card uses 8 bits per sample. However, 12- and 16-bit 24-bit samplers are also available. The wave format supports only 8- and 16-bit samples. This

means that for samplers using 12 bits. The WAVE format will still record 16 bits per sample, to avoid unnecessary calculation [ Stolz, A,1993].

Figure (3.7) presents the structure of an 8-bit mono wave file, where the sampling data are arranged as a sequence of bytes. While in Figure (3.8) the sampling data of 16-bit mono wave files are arranged as a sequence of unsigned integers (2 bytes).

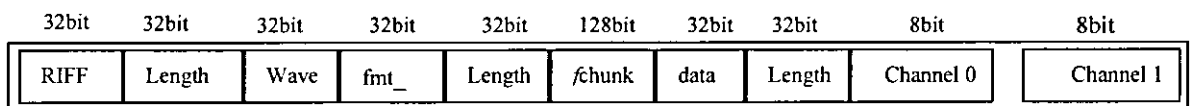


**Figure (3.7) Structure of 8-bit mono wave file**

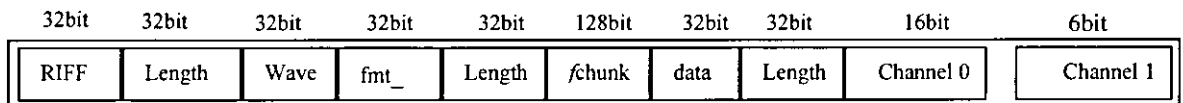


**Figure (3.8) Structure of 16-bit mono wave file**

In stereo mode using two channels (i.e., channel 0 and channel 1) generates the sampling data. Figures (3.9) and (3.10) shows the sampling data that arranged in alternating ways. The first word belongs to channel-0, the second word to channel-1, the third word is the second sample generated by channel-0, and so forth.



**Figure (3.9) Structure of 8 bit stereo wave file**



**Figure (3.10) Structure of 16 bit stereo wave file**

## **4. WAVE BASED STEGANOGRAPHY SYSTEM (WBSS)**

### ***4.1 Introduction***

Wave hiding information discipline was explored and evolved by several sophisticated systems.

In this chapter, new techniques are constructed and tested. The first one is called Wave Differential Technique. This technique deals with two kinds of messages, Text, and Images.

The second technique is based on Discrete Cosine Transformation and deal only with wave messages. Finally, possible results of the proposed techniques are fabricated and compared with traditional results of a well-known technique.

### ***4.2 Wave Based Steganography System (WBSS) Theoretical Concept***

Discrete Cosine Transformation (DCT), Differential Pulse Code Modulation (DPCM), and Mediator Amplitude Values Technique (MAV) are the most important concepts of WBSS. The following sections give a brief illustration of these notions.

#### ***4.2.1 Discrete Cosine Transformation (DCT)***

The DCT is a special case of a Discrete Fourier transform in which the sine components of the coefficients have been eliminated leaving a single number. In fact a DCT produces as many coefficients as its input samples. It is the primarily used in data reduction processing because it converts the input waveform into a form where redundancy can be easily detected and removed.

The DCT compute the unitary discrete cosine transform for an input vector or matrix. Mathematically, the unitary DCT of an input sequence  $x$  is:

$$y(k) = \sum_{n=1}^N w(n)x(n) \cos \frac{\pi(2n-1)(k-1)}{2N} \quad ,k=1, \dots, N \quad \dots\dots\dots(4.1)$$

Where:

$$w(n) = \begin{cases} \frac{1}{\sqrt{N}} & n=1 \\ \sqrt{\frac{2}{N}} & 2 \leq n \leq N \end{cases}$$

$N$  is the length of  $x$ , and  $x$  and  $y$  are of the same size.

The DCT is closely related to the discrete Fourier transform (DFT). However, it has better *energy compaction* properties, with just a few of the transform coefficients representing the majority of the energy in the sequence.

The energy compaction properties of the DCT make it useful in applications such as data communications and signal coding.

The equation bellow shows the inverse of Discrete Cosine Transform. This equation reconstructs a signal from a complete or partial set of DCT coefficients.

$$x(n) = \sum_{k=1}^N w(k)y(k) \cos \frac{\pi(2n-1)(k-1)}{2N} \quad ,n=1, \dots, N \quad \dots\dots\dots(4.2)$$

Where:

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k=1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases}$$

Because of the energy compaction mentioned above, it is possible to reconstruct a signal from only a fraction of its DCT coefficients [Jain, A.K.,1989] [Pennebaker, W.B., and Mitchell, J.L. ,1993].



### 4.2.2 Differential Pulse Code Modulation (DPCM)

As mentioned in chapter 3, PCM is a type of coding that is called waveform coding used to digitize the analog voice signals.

Differential PCM is a simple way to achieve modest compression. Instead of storing the samples directly, DPCM stores the differences among successive samples. If the sampling rate is fairly high, these differences will tend to be small. As a result, a fewer bits per sample is used to store just the differences [Kientzle, T., 1998].

### 4.2.3 Mediator Amplitude Values technique (MAV)

We introduce the following technique MAV that computes the average of odd successive samples of Cover wave media. Through the process, a new form of Cover wave media will be constructed. The constructed Cover wave media embed the secure information in average of the even samples.

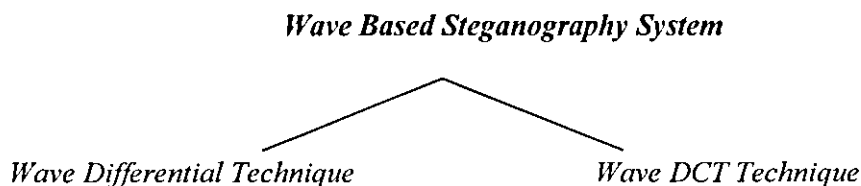
The equation (4.3) describes the MAV process.

$$W(t) = \left[ \frac{W(t-1) + W(t+1)}{2} \right] \dots \dots \dots \quad (4.3)$$

Where:  $t=2,4,6, \dots, N$

### 4.3 Wave Based Steganography System (WBSS) Techniques.

The WBSS consists of two new techniques to achieve the embedding and extracting process in wave-based steganography system. The following sections give more detail about proposed techniques.



### 4.3.1 Wave Differential Technique (WDT)

This technique consists of two algorithms the embedding and extracting algorithm which, dealing with two kinds of messages, Streams (Text) and Images (Gray scale image). The embedding algorithm processed through five steps. The *first step* is to check if the size of embedded message satisfies the equation (4.4):

$$E_{Size} * J < \left\lfloor \frac{C_{Size}}{2} \right\rfloor \dots\dots\dots (4.4)$$

Where:

$E_{Size}$  : stands for the embedded message size,

$J$  : stands for the number of jumps between Cover wave samples (assume it 2 worst case),

$C_{Size}$ : stands for the size of Cover wave media.

The *second step* is to compute the jump parameter. Jump parameter is defined as the starting embedded position. It can be calculated by dividing the size of cover over size of embed message, which differs every time we need to embed. This technique is called hide and jump technique.

The *third step* is to compute the differences between embedded message values by applying DPCM method. At the end of this stage we have to store the first value of embedded message in the key.

The *fourth step* is to take the Cover wave media samples and make the necessary initializations for the last step of embedding algorithm.

The *fifth step* is to embed the computed differences in the MAV of Cover wave media to generate Stego-wave media that holds the embedded message.

The second algorithm is the extracting algorithm. It is the reverse operation of embedding process, but without a stego-key this process cannot be accomplished.

The stego-key contains all embedded message restrictions (Size, Row, Column, etc) and it is generated through embedding process. Table (4.1a), shows stego-key fields.

Table (4.1a): the contents of the Stego Key

Field Description	Fist Byte in Embed Message	Message Row	Message Column	No. of Jumps	Message Type
No. of Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte

The stego-key message type field holds:

0: for gray scale images

1: for stream characters (Text)

The following are detailed embedding and extracting algorithms

#### 4.3.1.1 Embedding Average Amplitude Algorithm.

Beside the brief description illustrated in 4.3.1, the following is a full description of the embedding algorithm.

##### **INPUT**

C : represents Cover wave media

E : represents Embedded text or gray scale image media

##### **OUTPUT**

Stego : represents Stego-wave media that contains both Cover and Embedded wave media

Key : represents Stego - private Key

---

**Step 1:** Compute E size, the size of E must be at most half of C's size

If the  $\text{SizeOf}(E) * J_{\text{skip}} > \text{HalfOf}(C)$  then

Change C Cover media

Goto step 7

End If

**Step 2:** Compute the Jump Counter

$J_{\text{ski}} = \text{SizeOf}(C) / \text{SizeOf}(E)$

**Step 3:** Initialize matrix ( Dif ) to hold the difference between ( E ) values

Let  $\text{Key}_1 = \text{first byte of}(E)$

Let  $\text{Dif}_1 = \text{Key}_1$

For I = 1 to SizeOf (E)

    Compute the difference between the ( E ) values ,where

$$\text{Dif}_{I+1} = E_I - E_{I+1}$$

End For

**Step 4:** Initialize Jump counter ( N ) , where  $N = J_{\text{skip}}$  value

    Let Stego = C

    Compute the Average amplitude values of the Cover ( C ) ,where

    For j =N to SizeOf ( E )

$$\text{Stego}_N = ( C_{j-1} + C_{j+1} ) \text{ divided by } 2$$

    Increment N , where  $N = N + J_{\text{skip}}$

    End For

**Step 5:**Initialize Jump counter ( P ) , where  $P = J_{\text{skip}}$  value

    Adding the all elements of Dif matrix to the average values of Stego ,where

    For I = P to SizeOf ( E )

$$\text{Stego}_P = \text{Stego}_P + \text{Dif}_I$$

    Increment P , where  $P = P + J_{\text{skip}}$

    End For

**Step 6:**Store all necessary information {E width, E height, number of jumps, etc} in

    Key<sub>i</sub>, where  $i = 2 \ 0 \ 0 \ 05$

**Step 7:** Return Stego, Key

**Step 8:** End

### 4.3.1.2 Extracting Average Amplitude Algorithm

Beside the brief description illustrate in 4.3.1, the following is a full description of the extracting algorithm.

#### **INPUT**

Stego : represents Stego-wave media that contains both Cover and Embedded wave media

Key : represents Stego - private Key

#### **OUTPUT**

E : represents the extracting information (Embedded Media)

**Step 1:** Extract all information {E width, E height, number of Jumps, etc} in Key<sub>i</sub> ,  
where  $i = 2000005$ .

Set the first Key byte to Embed Media ( E ), where  $E_1 = \text{Key}_1$ .

Compute E size,

$$\text{SizE} = E_{\text{width}} * E_{\text{height}}$$

Let Dif be matrix of the same size of E.

**Step 2:** Initialize Jump counter ( N ), where N = number of Jumps value.

Compute the Stego values average and extract the differences between these values, where

For  $i = N$  to SizE

$\text{Tmp} = (\text{Stego}_{N-1} + \text{Stego}_{N+1})$  divided by 2

$\text{Dif}_i = \text{Stego}_N - \text{Tmp};$

Increment N , where  $N = N + J_{\text{skip}}$

End For

**Step 3:** Convert the difference of values ( Dif ) to the original Embedded

values, where

Let  $E_2 = E_1 - \text{Dif}_2$

For  $i = 2$  to  $\text{SizeOf}(\text{Dif})$

$E_{i+1} = E_i - \text{Dif}_{i+1}$

End For

**Step 4:** Return E

End

### 4.3.2 Wave DCT Technique

The WBSS supports another kind of hiding techniques, which is called Wave DCT technique. This technique is based on Discrete Cosine Transformation (DCT) method that is mentioned above. The DCT calculates and minimizes wave samples with some round-off error. Therefore it is applied only on wave messages: since the accuracy of extracted information must meet the originals. We can observe this in streams (Text) and images. Thus any little difference between the extracted and original Embedded wave message will not actually effect wave behavior and its clarity.

This technique consists of two algorithms, the embedding algorithm, and the extracting algorithm, which, deals with one kind of messages, which is Wave messages.

The embedding algorithm processed through five steps, the *first step* an embedded wave must exceed the size condition and match the (4.4) equation. In addition, to obtain best performance of proposed wave DCT technique, a shifted key was added to (4.4) equation and it becomes:

$$(E_{Size} * J) + S_{Max} < \left\lfloor \frac{C_{Size}}{2} \right\rfloor \dots\dots\dots (4.5)$$

Where:

$E_{Size}$  : stands for the embedded message size,

$J$  : stands for the number of jumps between Cover wave samples (assume it 2) ,

$C_{Size}$ : stands for the size of Cover wave media.

$S_{Max}$  : represent the maximum amplitude value of Cover wave media.



In this technique we add a Shifted key to increase security, and the process is to find the maximum value in the cover and start embedding from that position at the end we have to add the position of the largest value to the key.

The *second step* is to minimize the DCT round-off error, we must mutate all successive negative samples of Embedded wave media into positive samples by adding the maximum negative sample to those negative values.

Transforming the processed Embedded wave media into DCT form is the *third step* of the embedding process. The *fourth step* is to calculate the MAV technique and apply it on the samples of Cover wave media.

The *fifth step*, is to generate the Stego-wave media after hiding the Embedded wave message into Cover wave media.

The stego-key is the core of the extracting process. Table (4.1b) illustrates the stego key fields.

Table (4.1b): the contents of the Stego Key

Field Desc.	Embed minimum value	Embed Row	Embed Col.	DCT first Value	Shift	Embed Sample Rate	No. of Jumps	Cover max. value
No. of Byte	2 Byte	2 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Byte

The Shift field may contain:

- 0: no shift process support,
- 1: shift all embedded samples to appropriate position in Cover wave media

Reversing the embedded process leads to extracting the embedded message from Stego-wave media, and this procedure is called the Extracting process.

The following are detailed DCT based embedding and extracting algorithms.

### 4.3.2.1 DCT-Based Embedding Algorithm

Beside the brief description illustrated in 4.3.2, the following is a full description of embedding algorithm.

#### INPUT

- C** : represents Cover wave media
- E** : represents Embedded wave media
- J<sub>Skip</sub>** : represents the gaps between Cover wave values.
- Smp** : represents the Sample Rate of E wave media
- S<sub>disp</sub>** : represents the displacement of E media. This Variable takes (default value = 0 otherwise 1)

#### OUTPUT

- Stego** : represents Stego-wave media that contain both Cover and Embedded wave media
- Key** : represents Stego - private Key

**Step 1:** Compute E size, the size of E must be at most half of C's size.

If  $\text{SizeOf}(E) * J_{\text{Skip}} > \text{HalfOf}(C)$  then

Change C Cover media

Goto step 10

End If

**Step 2:** If  $S_{\text{disp}} = 1$  then

Shift the Embedded values according to the best match Cover position,

$B = \text{MaxOf}(C)$

Determine the size of E before shifting process,

$Siz = SizeOf ( E ) * J_{skip} + B$

If  $Siz > HalfOf ( C )$  then

Change C Cover media

Goto step 10

Else

Shifting E values,

$E = Shift ( E )$

End If

End If

**Step 3:** Substitute all negative amplitude values of E to positive number,

$M = MinOf ( E ),$

For  $i = 1$  to  $SizeOf ( E )$

$E_i = E_i + abs ( M )$

End For

**Step 4:** Normalize E values,

For  $i = 1$  to  $SizeOf ( E )$

$E_i = E_i$  divided by Smp

End For

**Step 5:** Calculate Discrete Cosine Transformation ( DCT ) to all E values,

$E_{det} = DCT ( E ).$

**Step 6:** Initialize Jump counter ( N ), where  $N = J_{skip}$  value

Let Stego = C

Compute the Average amplitude values of the Cover ( C ), where

579071

For  $j = 2$  to  $\text{SizeOf}(E_{\text{det}})$

$\text{Stego}_N = (C_{j-1} + C_{j+1})$  divided by 2

Increment  $N$ , where  $N = N + J_{\text{skip}}$

End For

**Step 7:** Initialize Jump counter ( $P$ ), where  $P = J_{\text{skip}}$  value

Adding the all elements of  $E_{\text{det}}$  matrix to the average values of Stego, where

For  $I = 2$  to  $\text{SizeOf}(E_{\text{det}})$

$\text{Stego}_P = \text{Stego}_P + E_{\text{det } I}$

Increment  $P$ , where  $P = P + J_{\text{skip}}$

End For

**Step 8:** Store all necessary information {E width, E height, number of jumps, etc} in

$\text{Key}_i$ , where  $i = 200008$

**Step 9:** Return Stego, Key

**Step 10:** End

### 4.3.2.2 DCT-Based Extracting Algorithm

In addition to the brief description illustrate in 4.3.2, the following is a full description of extracting algorithm.

#### INPUT

Stego : represents Stego-wave media that contain both Cover and Embedded wave media

Key : represent Stego - private Key

#### OUTPUT

E : represent the extracting information (Wave Embedded Media)

---

**Step 1:** Extract all information {E width, E height, number of Jumps, etc} in  $Key_i$ ,  
where  $i = 20000 \div 8$ .

Set the forth Key byte to Embed Media ( E ), where  $E_1 = Key_4$ .

**Step 2:** Determine the E values position on C Cover media,

If shift value in  $Key_5 = 1$  then

    Compute E size,

$$SizE = E_{width} * E_{height} + \text{Shifting Value}$$

Else

$$SizE = E_{width} * E_{height}$$

End If

Let Dif be matrix of the same size of E.

**Step 3:** Initialize Jump counter (  $N$  ), where  $N$  = number of Jumps value.

Compute the Stego values average and extract the differences between these values, where

For  $i = 2$  to  $\text{SizeE}$

$\text{Tmp} = (\text{Stego}_{N-1} + \text{Stego}_{N+1})$  divided by 2

$\text{Dif}_i = \text{Stego}_N - \text{Tmp}$ ;

Increment  $N$ , where  $N = N + J_{\text{skip}}$

End For

**Step 4:** Calculate Inverse Discrete Cosine Transformation ( IDCT ) to all  $\text{Dif}$  values,

$E_{\text{idct}} = \text{IDCT}(\text{Dif})$ .

**Step 5:** Un-Normalize  $E_{\text{idct}}$

For  $X = 1$  to  $\text{SizeOf}(E_{\text{idct}})$

$E_X = E_{\text{idct}_X}$  multiply by  $\text{Smp}$

End For

**Step 6:** Reshape the  $E$  elements by returning the old negative values,

For  $X = 1$  to  $\text{SizeOf}(E_{\text{idct}})$

$E_X = E_X - \text{abs}(\text{Minimum value of } E \text{ in Key}_1)$

End For

**Step 7:**Return  $E$

End

## 4.4 WBSS Experiment Results

### 4.4.1 Wave Differential results

#### Experiment #1:

In the first experiment text is embedded into wave. This experiment is applied on Cover wave media with the following properties:

Table (4.1) The Wave Cover media properties

The property	Value
File Size	55784 bytes
Number of Channels	2
Number of Bits per Sample	16
Average Bytes per Second	88200
Wave type	Stereo
Sample Rate	22050
Duration	0.315669 seconds

Table (4.1), notices that the wave we used as a cover media from stereo type, which contains two channels with sample rate 22050.

In order to see how the color of an image is distributed, we have to get its histogram. The amplitude distribution of the wave is determined through the histogram also.

Figure (4.2) shows the spectrum of cover wave file.

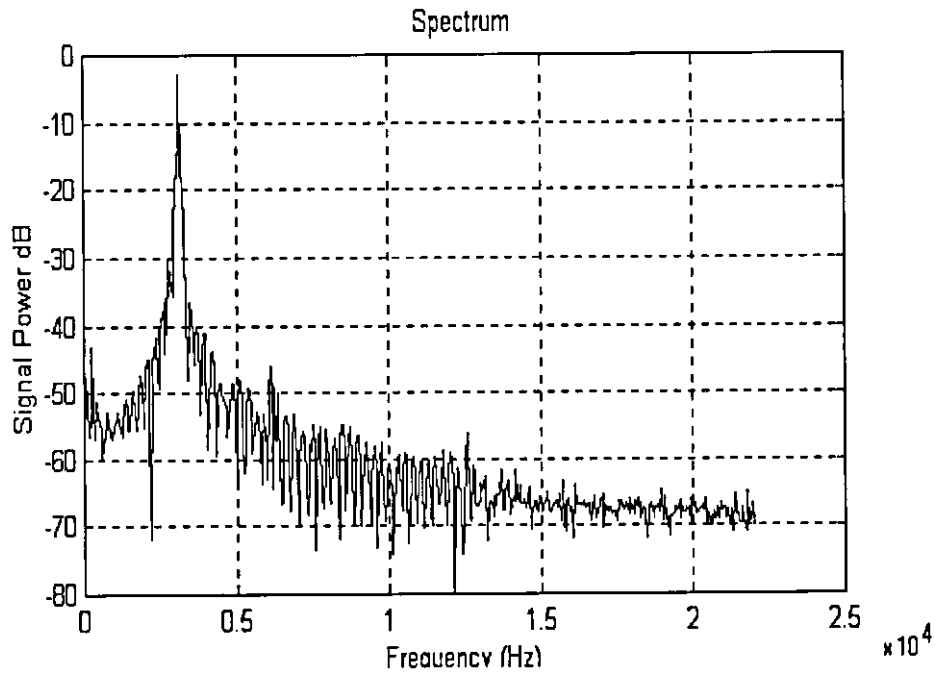


Figure (4.2) The wave Cover spectrum

- *Embedded message: steganography wave system.*

The Stego-wave media is generated after applying the wave differential technique. Table (4.2) shows the Stego-Key.

Table (4.2) The Stego-Key

Fist Byte in Embed Message	Message Row	Message Column	No. of Jumps	Message Type
83	1	25	1113	1

Table (4.2), notice that, the size of the embedded message is 25 characters; the jump is 1113, which indicates the position of the embedded message in the cover media.



Also, the first byte in embed message. Finally, the message type, if it contains 1 it means that the embedded message is text else it is image.

Table (4.3) shows differences data for embed message.

Table (4.3) The Embed message differences data

83	-1	15	-2	6	-13	-1	8	-11	17	-15	8	-17
25	-23	22	-21	17	5	-19	-6	6	-1	15	-8	

As mentioned in the beginning of this chapter, we have to calculate the difference between the embedded message element (Table (4.3)), which gives us a small element to be embedded in the cover.

Finally, the result of embedding text into wave file. In order to see, the result we have to compare the cover, the stego-wave media histogram and the frequency spectrum.

Figure (4.3) shows the stego wave media.

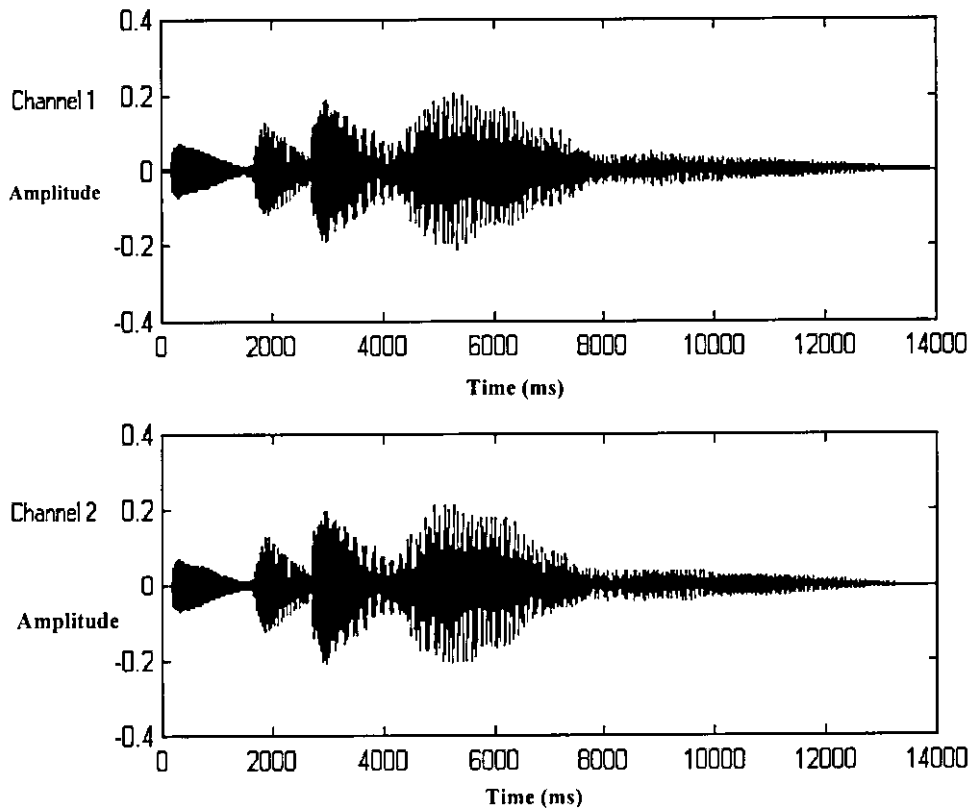
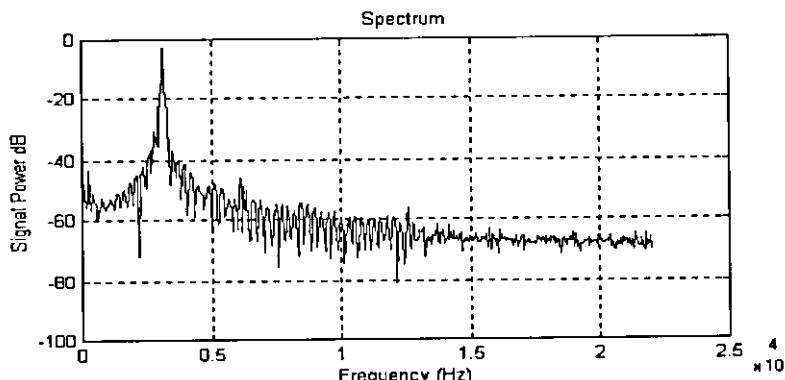


Figure (4.3) The Stego wave media



**Figure (4.4) The Stego wave spectrum**

Figure (4.3) and (4.4) show that there is a very slight difference between the cover and stego-wave media, which make it not noticeable.

To determine the quality of Stego-wave media after embedding message. The PSNR, MSE, and MAE were applied. Table (4.4) shows the results of embedding text in wave media file.

**Table (4.4) The comparison between proposed technique (wave differential technique) and least significant bit (LSB)**

Tech.	Length in Byte	Changed bytes in Cover	MSE	MAE	PSNR
Proposed system (WDT)	25	25	0.000047261049	0.000000873083	86.509932800724
LSB	25	200	0.000002606964	0.000000149857	111.67729789396

In our WDT (table (4.4)), the result of MSE, MAE became very low and close to zero, which is a good result. PSNR became good also because whenever the MSE, MAE have low values the PSNR must be high.

Comparing the result of our proposed system with LSB (table (4.4)), notice that, the amount of embedded information is greater than the one with LSB (reach half the cover), on the other hand, the result has been effected by this advantage with the difference in PSNR of both.

Also the changed bytes in proposed WDT is very small comparing it with LSB (Embedded = 25, proposed =25, LSB = 200).

**Experiment #2:** was to embed Image into wave; this experiment is applied on Cover wave media with the following properties:

Table (4.5) The Wave Cover media properties

The property	Value
File Size	313124 bytes
Number of Channels	2
Number of Bits per Sample	16
Average Bytes per Second	88200
Wave type	Stereo
Sample Rate	22050
Duration	1.77451 seconds

Table (4.5), notice that the wave is used as a stereo covers media. It contains two channels with sample rate 22050, and file size 313124 bytes.

Figure (4.5) shows the distribution of wave amplitude values which contains two channels, the range of the amplitude in this wave only from -1 to 1 in y-axis, and the times that amplitude repeated in that wave in x-axis .

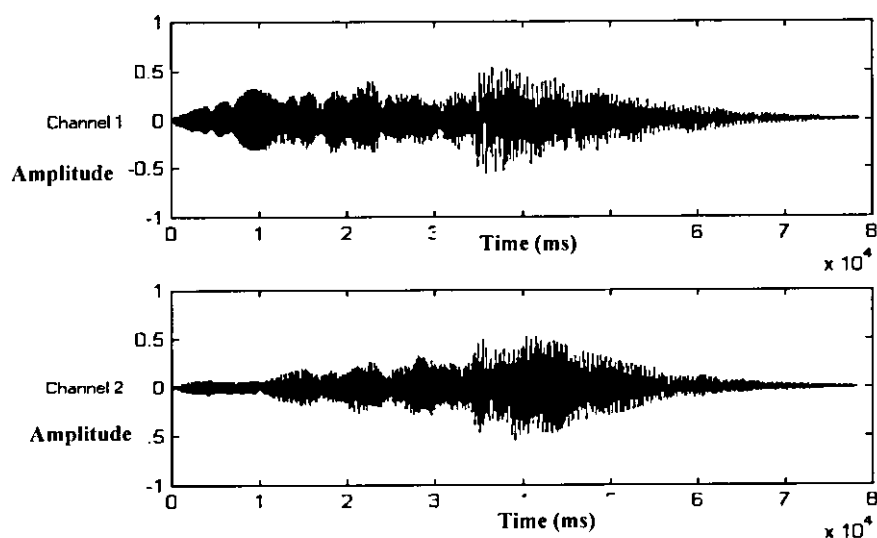


Figure (4.5) The wave Cover media

Sound is represented in terms of the amount of vibration at each individual frequency, which is usually presented as a graph of either power function of frequency or pressure function of frequency. This graph is called frequency spectrum. The power or pressure is measured in decibel and the frequency is measured in Hertz or kilohertz

Figure (4.6) shows the cover wave spectrum.

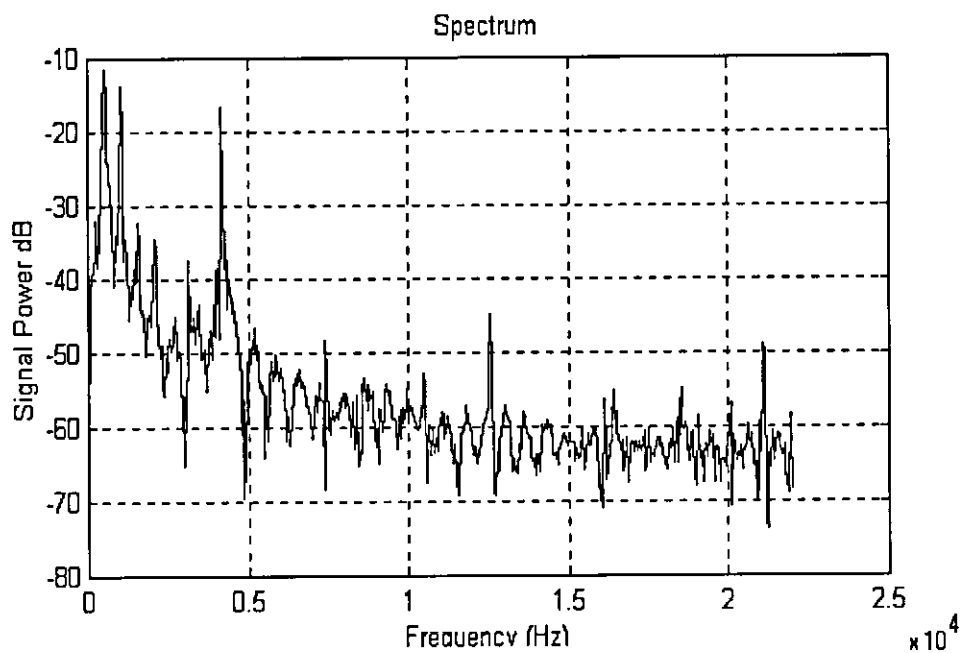


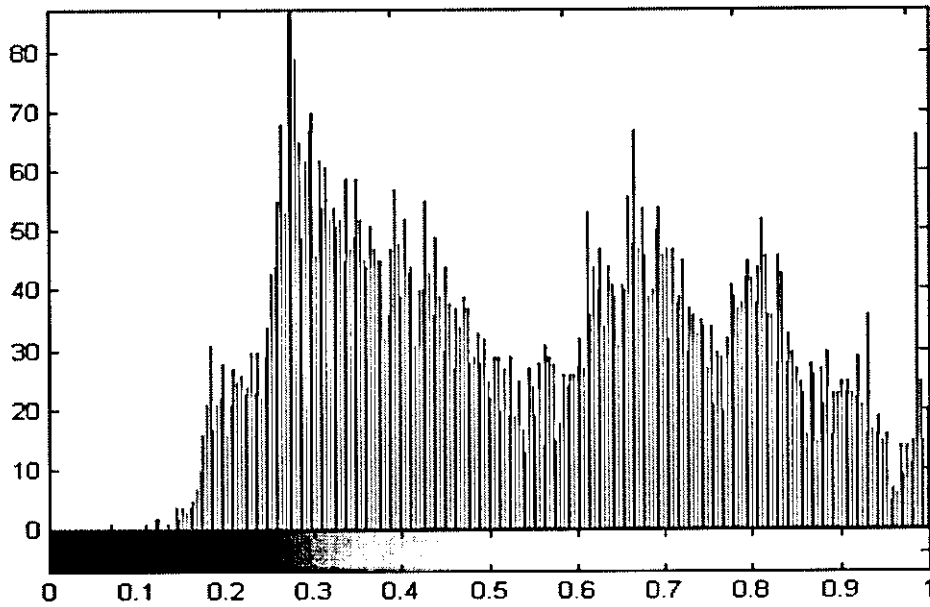
Figure (4.6) The wave Cover spectrum

- **Embedded media:** in the second experiment the embedded media is gray scale image.

Figure (4.7), illustrate the embedded image (Nature image) , while Figure (4.8) shows its histogram.



**Figure (4.7) the embedded image**



**Figure (4.8) the histogram of embedded image**

Table (4.6) shows the properties of stego-key, which is initialized after the embedding process. Stego-key is needed to extract the embedding message from stego.

**Table (4.6) The Stego-Key**

Fist Byte in Embed Image	Message Row	Message Column	No. of Jumps	Message Type
230	75	100	20	0

Table (4.7) describe the embedded image differences data.

Table (4.7) The Embed message differences data

230	-11	-10	-2	1	0	0	1	-1	0	0	0	0
0	0	0	-1	3	20	7	21	10	15	15	10	

At the end of embedding process we have to compare the results of both cover and stego after embedding the selected gray scale image, Figure (4.9) shows the histogram of the stego-wave, while Figure (4.10) describes the frequency spectrum of stego-wave.

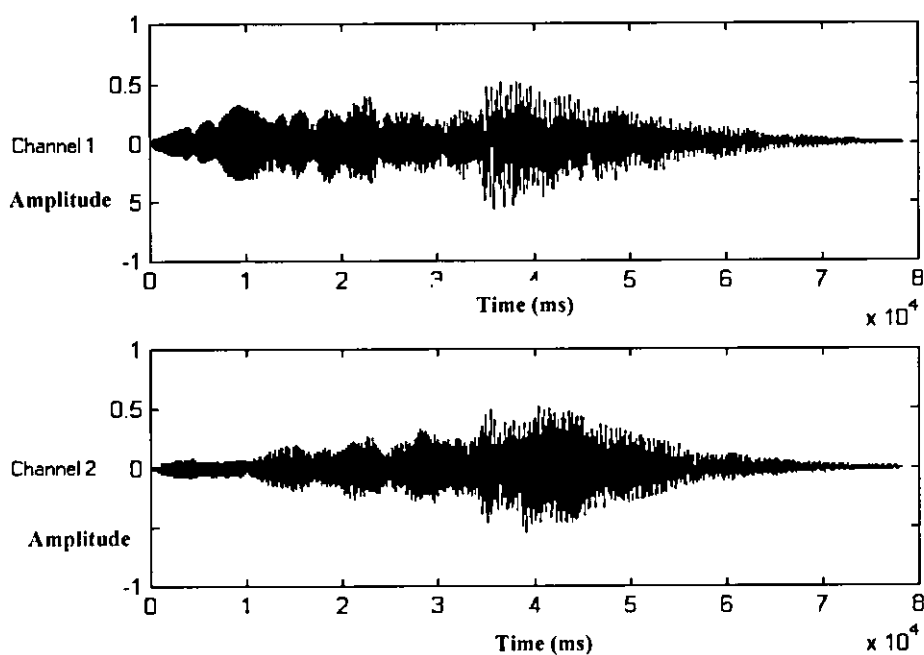
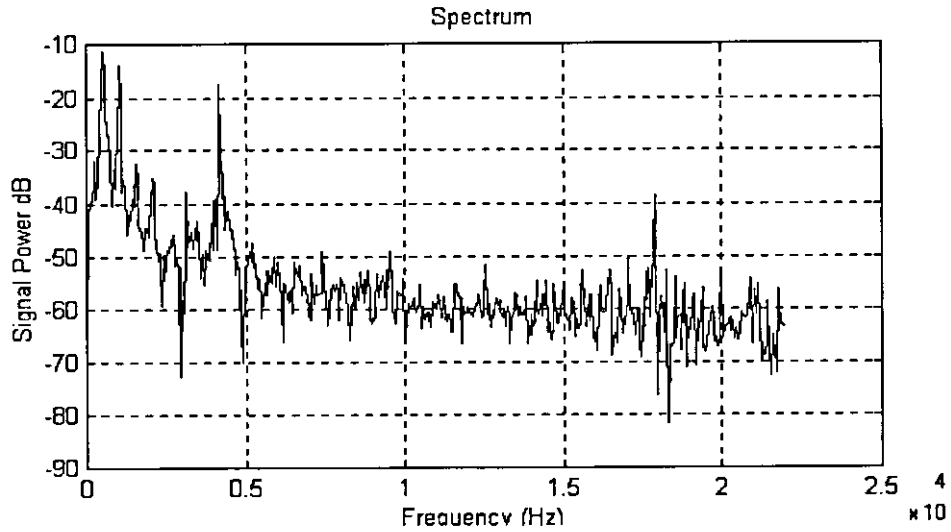


Figure (4.9) The Stego-wave Media



**Figure (4.10) the stego wave spectrum**

Finally, and after finishing the embedding process we have to compare results of the proposed WDT with LSB. Table (4.8) shows the PSNR, MSE and MAE results of the proposed and LSB.

**Table (4.8) The comparison between proposed technique (wave differential) and least significant bit (LSB) Technique.**

Tech.	Length in Byte	Changed bytes in Cover	MSE	MAE	PSNR
Proposed (WDT)	7500	7500	0.000402839899	0.00006.7126335	67.897350427251
LSB	7500	60000	0.000019835846	0.000008675810	94.050985274902

From the table above, notice that LSB and Proposed algorithm differ in the number of the changed bytes (60000 bytes in LSB, and 7500 bytes in the proposed). This difference effects MSE, MAE, and PSNR, which become better than the proposed WDT.

#### 4.4.2 Wave DCT results

DCT technique is used in the second approach to eliminate the Embedded wave media samples. The following are the results of three tests applied on different wave media type.

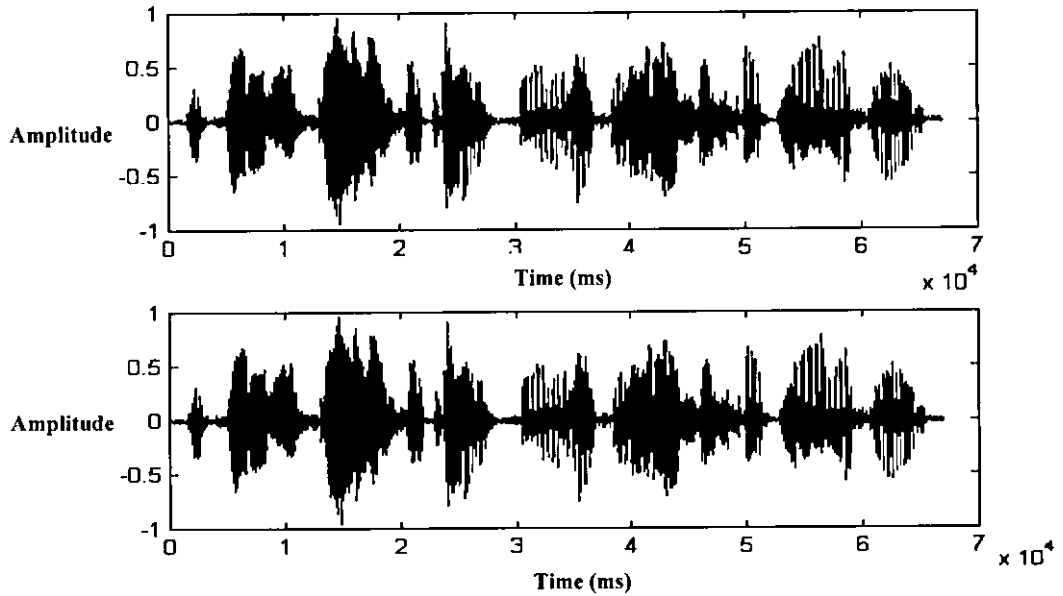
*Experiment #1:* The first experiment in this technique is applied on cover wave media with properties shown in Table (4.9).

Table (4.9) The Wave Cover media properties	
The property	value
File Size	134174
Number of Channels	2
Number of Bits per Sample	8
Average Bytes per Second	44100
Wave type	Stereo
Sample Rate	22050
Duration	1.52011 seconds
Wave contain	The King Abdullah II for Information Technology

From the table (4.9), notice that the wave that is used as cover media from stereo type, which contain two channels with sample rate 22050.

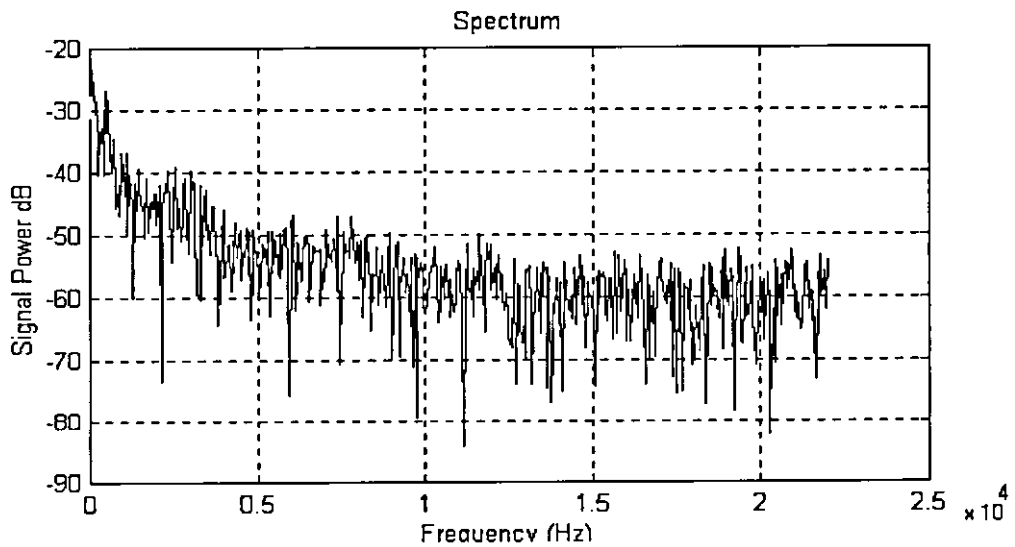
Figure (4.11) shows the distribution of wave amplitude values which contain two channels, range of amplitude in this wave is from -1 to 1 in y-axis, and the times that amplitude repeated in that wave in x-axis.





**Figure (4.11) The Cover wave media**

To see how sound is represented in terms of the amount of vibration at each individual frequency. The following Figure (4.12) shows the frequency spectrum of the cover wave media.



**Figure (4.12) The Cover wave media spectrum**

- **Embedded media #1:** The first embedded media was *Speech wave media*, which have the properties shown in table (4.10). The wave embed from Mono type, which contains only one channel with 8 bits per samples and sample rate 22050.

The property	Value
File Size	20116 bytes
Number of Channels	1
Number of Bits per Sample	8
Wave type	Mono
Sample Rate	22050
Duration	0.91229 seconds
Wave contain	I.T. College

Figure (4.13) shows the histogram of the embedded message which contains only one channel of data, but Figure (4.14) shows the frequency spectrum and the relation between the power and frequency in that graph.

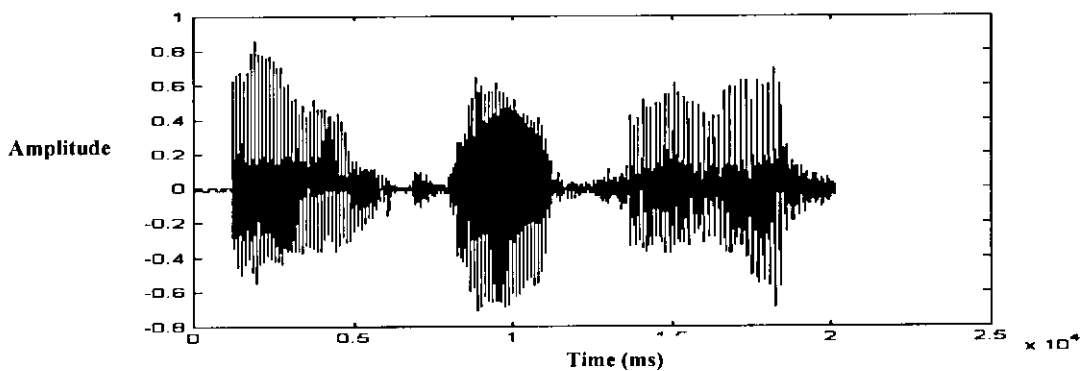
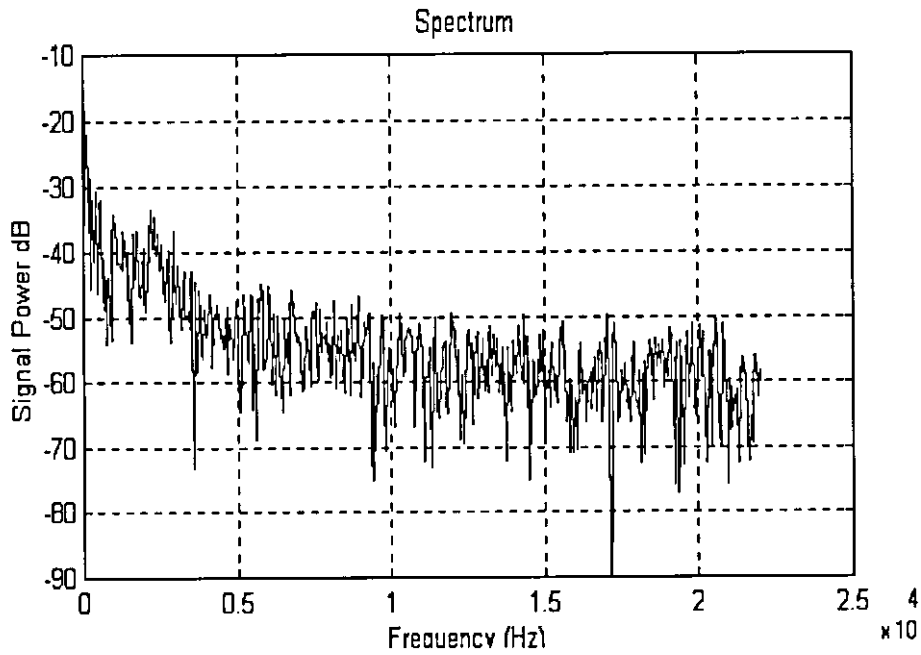


Figure (4.13) The Embedded wave media

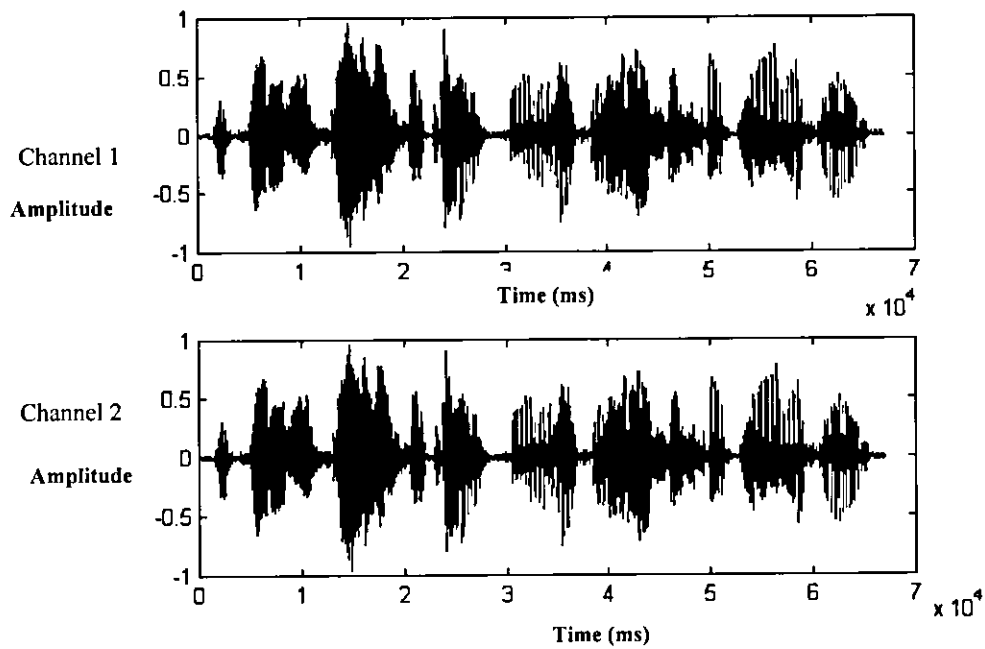


**Figure (4.14) the spectrum analysis of Embedded wave media**

The stego-key is generated after the embedding process end contains of the minimum embedded value, which is important to make the negative values in the embedded message process positive, DCT first value contains cover maximum value which we use for shifting process to increase security for the process.

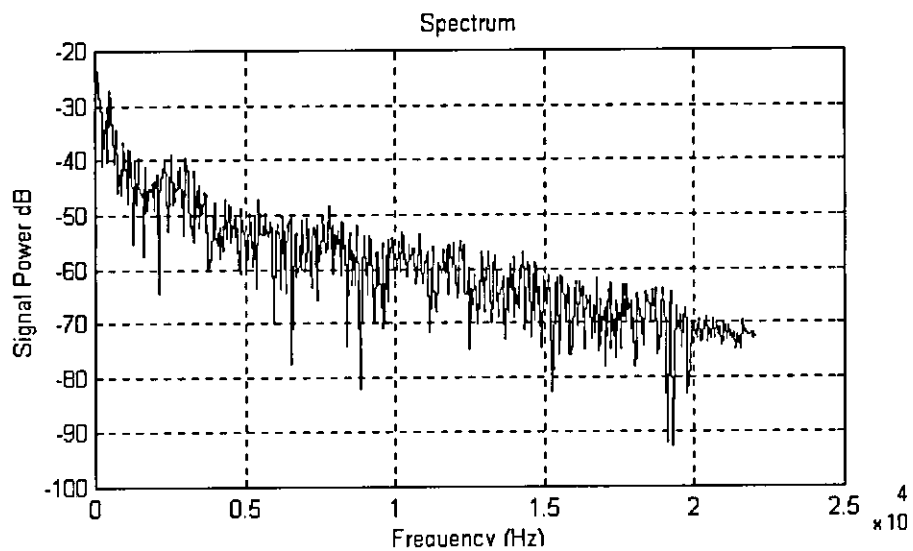
Embed minimum value	Embed Row	Embed Col.	DCT first Value	Shift	Embed Sample Rate	No. of Jumps	Cover maximum value
-15676	20116	1	99.7232	0	22050	2	14767

Finally, after achieving the end of embedding process, the stego wave is generated (Figure 4.15). This stego wave contains two channels of data. It also becomes nearer to the cover histogram. Beside, we have the stego frequency spectrum (Figure (4.16)), which becomes also nearer to the cover spectrum.



**Figure (4.15) The Stego-wave media**

Figure (4.16) shows the spectrum of generated stego-wave media.



**Figure (4.16) The spectrum of Stego-wave media**

- **Embedded media #2:** by using the same cover we embed another form of wave called *Normal audio media*. This wave has several properties that differ from speech wave media.

Table (4.12) shows the properties of this media. It is from the stereo type it contains two channels of 8 bit per sample, and sample rate of 22050.

The property	Value
File Size	28316 bytes
Number of Channels	2
Number of Bits per Sample	8
Average Bytes per Second	44100
Wave type	Stereo
Sample Rate	22050
Duration	0.319909 seconds
Wave contain	Part of song

Figure (4.25) shows the histogram of the wave file, which contains distribution of amplitude over the wave, the number of channels it contains (two channels), and the repeated amplitude all over the wave.

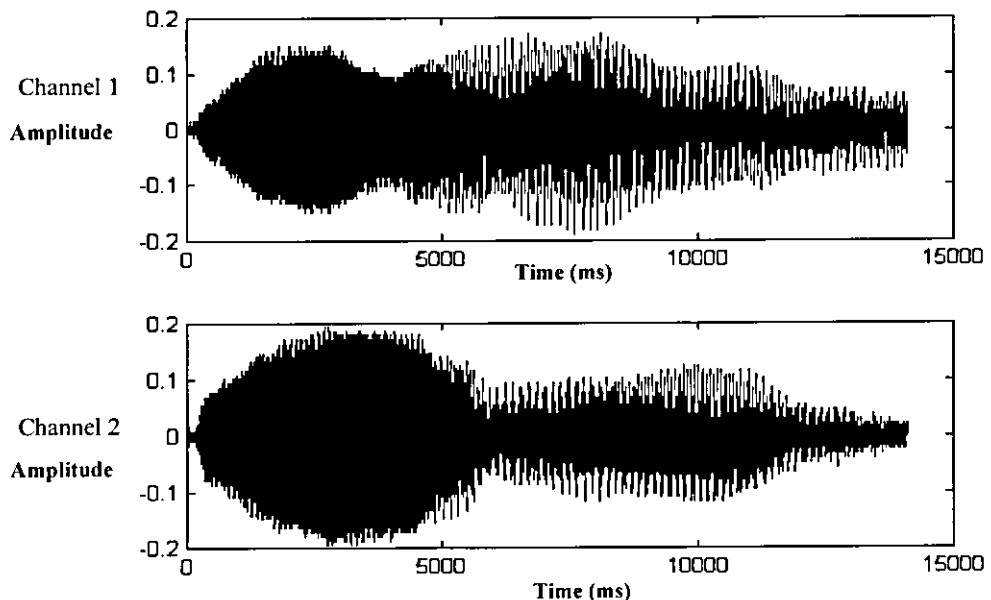
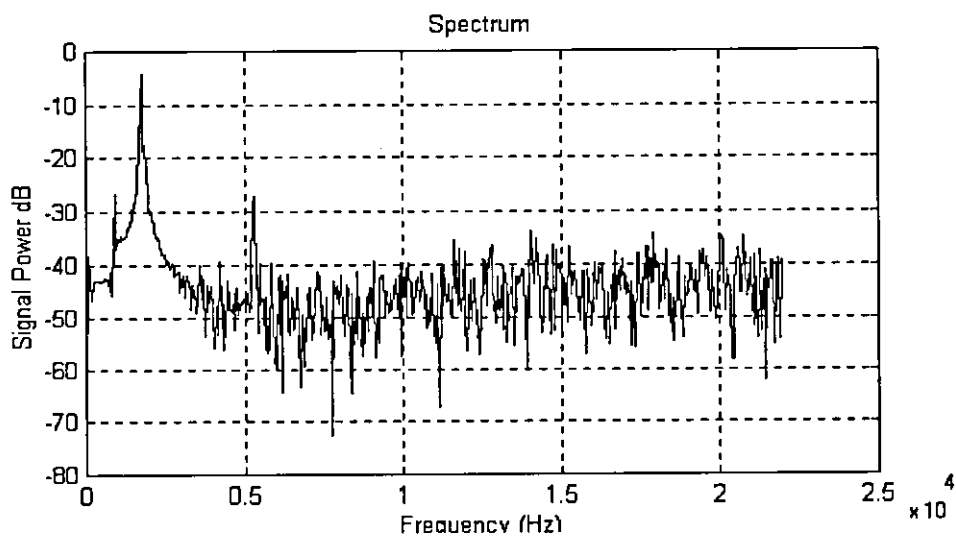


Figure (4.25) The Embedded wave media

Figure (4.26) shows the frequency spectrum of the embedded normal wave media.



**Figure (4.18) The spectrum of Embedded wave media**

After finishing the embedding process stego-key is generated, which is important for the extractor to get the embedded message. Table (4.13) shows the stego-key.

Embed minimum value	Embed Row	Embed Col.	DCT first Value	Shift	Embed Sample Rate	No. of Jumps	Cover maximum value
-4306	14108	2	32.7971	0	22050	2	14767

Figures (4.19), (4.20) show Stego-wave media and spectrum after embedding above wave media. From these figures, the stego-wave becomes nearer to the original one. The nearer the stego-wave is to the original one, the more difficult it becomes for the passive attacker to detect.

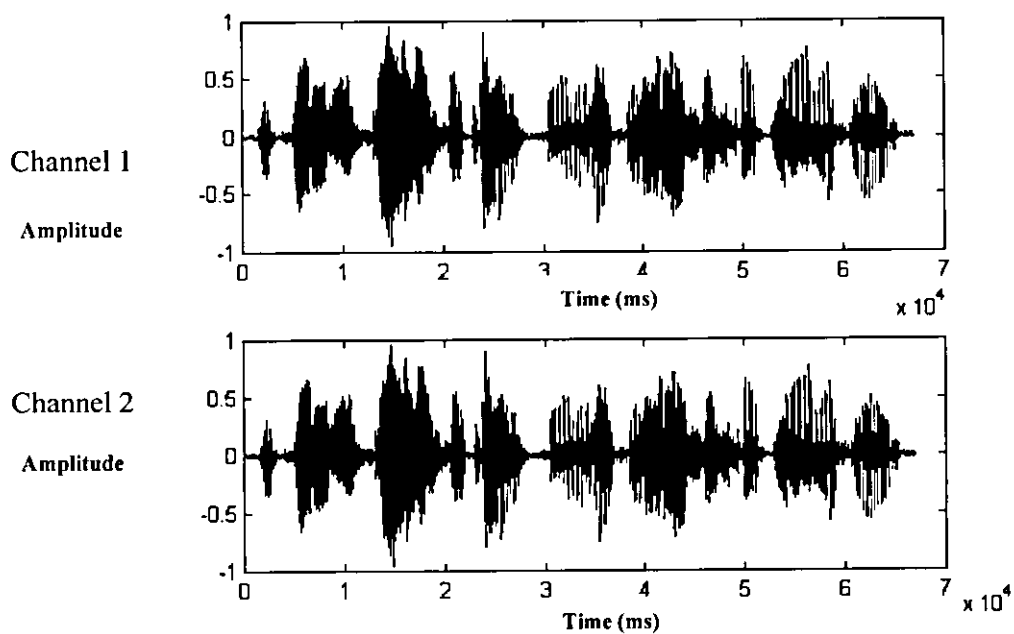


Figure (4.19) the spectrum of Stego-wave media

Figure (4.20) shows the spectrum of Stego-wave media.

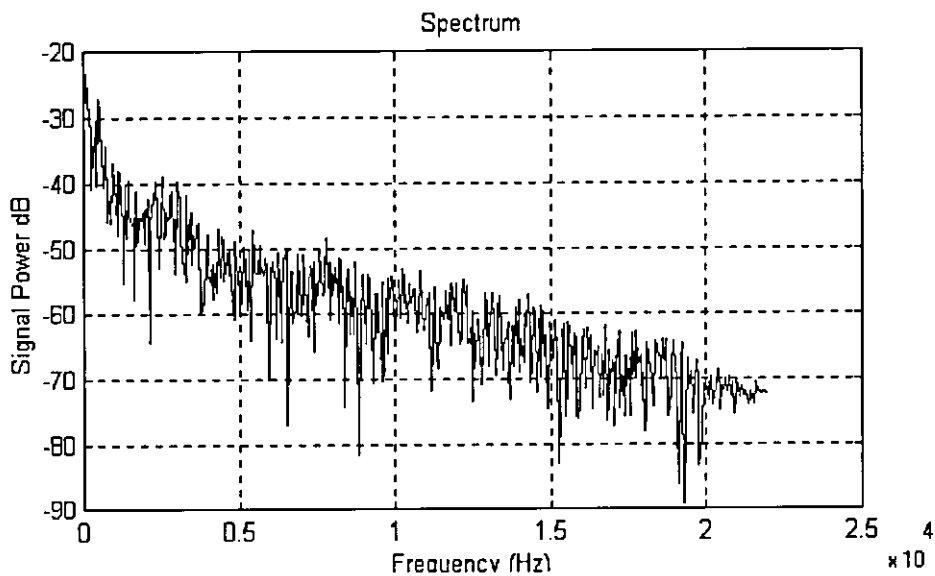
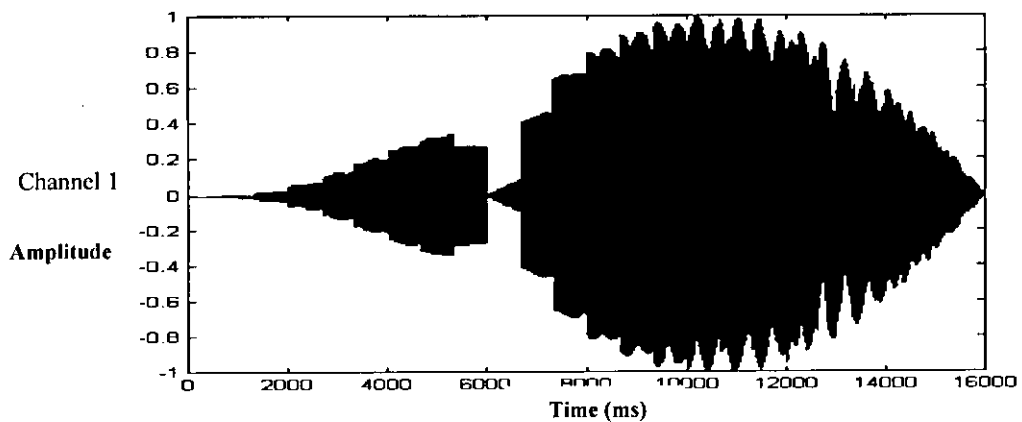


Figure (4.20) The spectrum of Stego-wave media

**Embedded media #3:** The final test for this technique has been done on the uniform audio media which is produced by using a special kind of sinusoidal functions. This result are the uniform samples. The properties of generated wave are shown in Table (4.14).

Table (4.14) The Wave embedded media properties (Uniform)	
The property	Value
File Size	16100 bytes
Number of Channels	1
Number of Bits per Sample	8
Average Bytes per Second	8000
Wave type	Mono
Sample Rate	8000
Duration	2 seconds

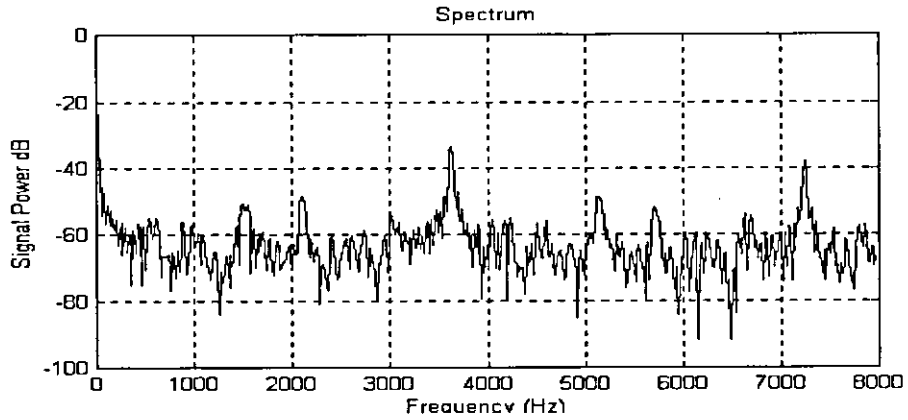
By looking to the Figure (4.21), notice the histogram for embedded wave with uniform behavior for samples with amplitude distribution from 1 to -1 and one data channel.



**Figure (4.21) The samples distribution of uniform audio media**



Figure (4.22) shows the frequency spectrum of the uniform audio media.

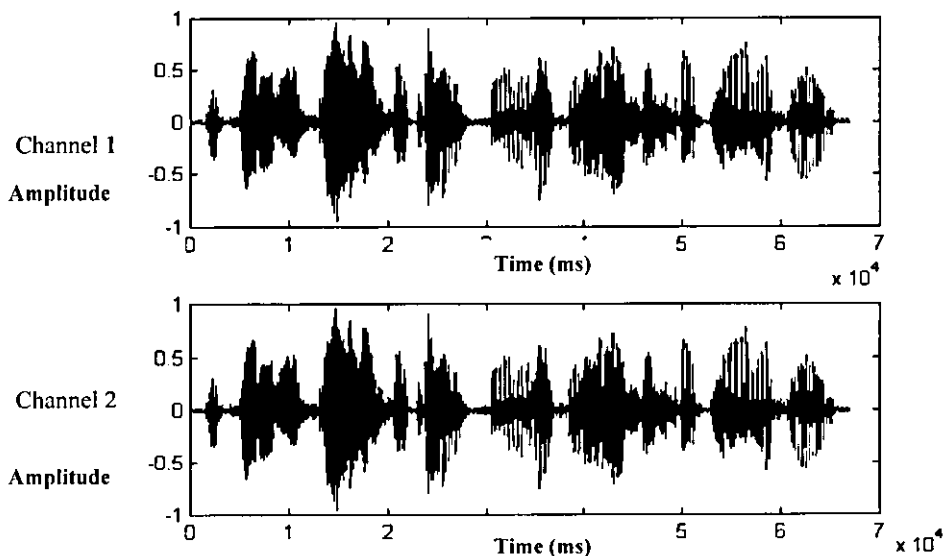


**Figure (4.22): The spectrum of uniform audio media**

The Stego-wave media is generated after embedding the Uniform wave media. Table (4.15) shows the Stego-key, while figure (4.23) shows the amplitude distribution of Stego-wave media.

Table (4.15) The Stego-Key							
Embed minimum value	Embed Row	Embed Col.	DCT first Value	Shift	Embed Sample Rate	No. of Jumps	Cover maximum value
-8000	16000	1	125.6854	0	8000	2	14767

The embedding process has been finished. Thus we have to compare both the histogram of stego with the cover to see how efficient the process was. Figures (4.23) and (4.24) shows the histogram of the stego and the frequency spectrum also.



**Figure (4.23) The Stego-wave media**

Figure (4.24) shows the Stego-wave Spectrum.

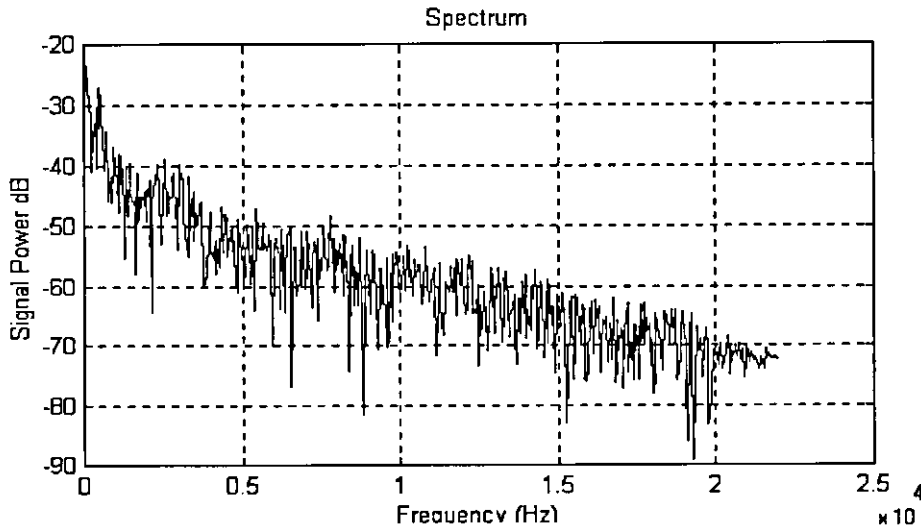


Figure (4.24) The Stego-wave media

Table (4.16), illustrates the result of wave DCT technique after applying three different types of embedded wave media.

Table (4.16) result of wave DCT technique

8-Bit Stereo Speech Wave File (134174 bytes)					
No	Message	Length in Byte	MSE	MAE	PSNR
1	Speech wave	20116	0.003338142851	0.000654690090	49.529901643869
2	Normal wave	28316	0.004943509054	0.001089397211	46.119293321505
3	Uniform wave	16100	0.003206302952	0.000554873746	49.879908899796

From the table above we notice that the uniform wave media gave the best result because the samples take a uniform behavior by using special sinusoidal functions. Next comes speech wave media, which contains silent gaps between the recorded speeches. In the last stage came normal wave media which samples behave different distribution.

Beside that, if we compare the results we can see that whenever the embedded data becomes larger the MSE, MAE, and PSNR are affected and become worse.

## 5. DISCUSSION, AND FUTURE WORKS

### *5.1 Discussion*

Steganography has its place in security. So hiding a message with steganography methods reduces the chance of a message being detected.

In this proposed system, two techniques are discussed as possible methods for embedding data in wave media file. The first one is called Wave Differential Technique, which deals with two kinds of messages, Text, and Images. The second technique is based on Discrete Cosine Transformation and deal only with wave messages.

The proposed system deals with three types of secret information, streams (Text), gray scale images and audio (wave) media.

Our proposed system is one of the substitution systems, and from secret setganography type, where we try to encode secret information by substituting insignificant parts of the cover by secret message bits. The receiver can extract the information if he knows the position where secret information has been embedded. By using hide and jump technique in our system, together with the secret key, the only one who can detect the existence of the embedded message is the holder of secret key.

In WDT (first technique), there is a primary step before embedding, which calculates the differences among samples instead of storing the samples directly. As a result, small difference and fewer bits per sample will be used to store just the differences. By using this method for hiding message. The level of security raised to quite a satisfactory level. Now, even if the hidden message were discovered. The person who is trying to get the message

would only be able to lay his hands on the encrypted message without being able to decrypt it.

The distribution of embedding message all over the cover makes a noticeable difference in the spectrum of the stego. Figure (5.1) shows spectrum (for exp.#1 embedding text) for cover media. Figure (5.2) shows how spectrum will be when we make the Jump constant (Jump=2). Finally after distribute the embedding all over the cover shown in Figure (5.3).

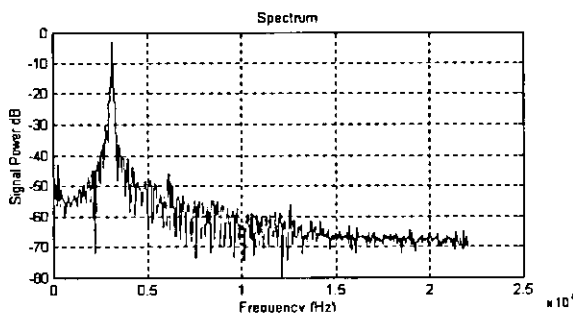


Figure (5.1) The Cover wave spectrum

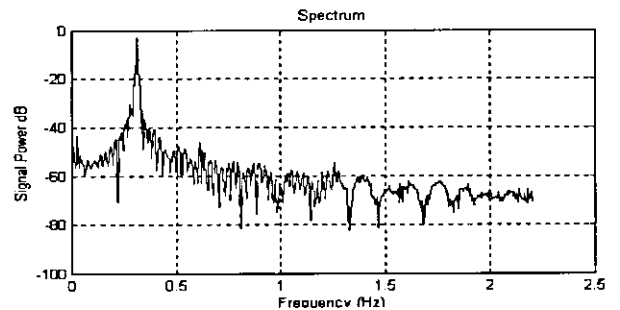


Figure (5.2) The Stego wave spectrum jump=2

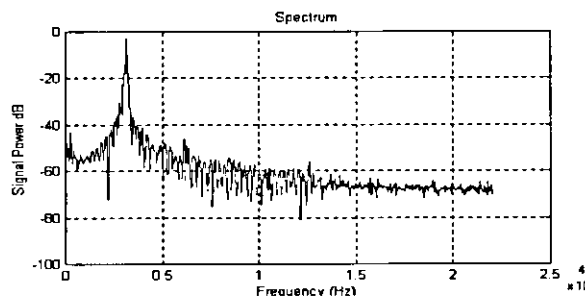


Figure (5.3) The Stego wave spectrum for distributed embedded media

A comparison between the results of the suggested technique (Wave Differential Technique) and the results of well-known method (LSB), indicates that the PSNR, MSE, and MAE (94.05, 0.000019, 0.0000867) of the LSB technique results are better than the wave differential technique (63.4700, 0.00067066, 0.00011968). Since the amount of embedded data increase, while changed bytes decrease which effect on the MSE, MAE,

and PSNR for the proposed technique.

The amount of embedded information of Wave Differential Technique is greater than the information that is embedded with LSB method, the size of selected embedding audio media approximately is equal to the half of the size of cover wave media, this makes it suitable for hiding a vast amount of information. In LSB we can only hide one bit by byte which increases the changed bytes in cover (Embed message =7500 byte, changed in LSB = 60000 byte, proposed = 7500 byte). This makes the proposed system better than LSB for embedding large amount of data.

The suggested technique (WDT) is found faster than LSB method. The proposed methods take  $O(N/2)$ , where LSB take  $O(N)$  in the worst case (if jump counter =2 ). Also the proposed methods are not causing significant degradation in PSNR.

In case of computing the average values of cover even samples, the number of embedded bits comes to 10 bits per word (16-bit wave file) and 6 bits per byte (8-bit wave file), while the LSB embed one bit per byte.

In wave DCT technique (second technique), to embed wave inside another, notice that using the DPCM technique is not efficient to embed the differences between samples. In this case, the Discrete Cosine Transformation (DCT) technique was used in data reduction processing because it converts the input waveform into a form where redundancy can be easily detected and removed.

Three forms of audio media are used in wave DCT technique, (speech, normal, and uniform). The comparison of the extracted PSNR, MSE, and MAE values these types. Notice that, the uniform wave media (PSNR=49.87990, MSE=0.003206, MAE=0.0005548) was the best result because of the samples takes a uniform behavior by using special sinusoidal functions. Next comes speech wave media (PSNR=49.5299, MSE=0.0033814, MAE=0.000654), which contains silent gaps between the recorded speech, while normal wave media whose samples behave in different distribution came last.

Using shift as a parameter in a secret key, gives the sender each time different key and position for embedding the secret message. This adds another level of security for the process.

The LSB method couldn't be used as opponent for DCT technique, because the size of selected embedding audio media is approximately equal to half of the size of the cover wave media and submitting to the equation (4.5).

- For example to embed an audio media of size 16100 bytes inside an audio of size 134174 bytes we have the following:

The number of bytes changed using the LSB method are :

$$(16100 * 2 * 8) = 257600, \text{ this number greater than } 134174.$$

The number of bytes changed using DCT technique using equation (4.5)

$$(16100 * 2) + 0 = 32200, \text{ this number is less than } (134174 / 2).$$

Form these calculations, it is shown that by using the DCT method only 50% of the total bytes will be changed while by using the LSB 100% of the total bytes will be changed

an attack..

Whenever embedded data increase, the result (MSE, MAE, PSNR) becomes worse than small data. So if we need non-noticeable hidden data we have to make it small as possible as we can.

## ***5.2 Future Works***

The following suggestions are put forward as materials for future research works:

1. Compose multi-layer-hiding technique to embed and scatter the embedded media bytes in the cover wave media.
2. Developing hiding techniques that are depending on different transformation methods like (Wavelet Transformation and Fractal Techniques).
3. Applying Dithering technique on cover wave media to reduce the precision of wave data. Also this technique provides plenty of opportunity to store information.

## 6. REFERENCES

- Alan, A. 1989. *Audio Technology Fundamentals*. Howard W. Sams & Company, USA.
- Beker, H., and Piper, F. 1982. *Cipher Systems the Protection of Communications*. Northwood Book, London.
- Brice, R. 1997. *Multimedia and Virtual Reality Engineering*. Newnes, Great Britain.
- Cachin, C. 1998. *An Information Theoretic Model for Steganography*. Lecture notes in computer science. Springer-Verlage, UK.
- David, R. and Tim, T. 1993. *The Audible PC*. Sybex Inc., USA.
- Durand, B. 1997. *3-D Sound for Virtual Reality and Multimedia*. McGraw Hill, USA.
- Fabien, P., Ross J., and Kuhn G. 1999. Information Hiding-Survey. *IEEE Trans. On computer*, 87(7):1062-1078.
- Forouzan, B., Coomb, C., and Fegan, C. 1998. *Introduction to Data Communication and Networking*. McGraw-Hill Co., USA.
- Gary, K. 2002. Hiding data in data. *Windows & .NET magazine*, 1:10-15 .
- Grul, W., Neel, M., and Lu, A. 1996. Techniques for Data Hiding. *IBM System Journal*, 35 (384):1-10.



- Jain, K. 1989. *Fundamentals of Digital Image Processing*. Englewood Cliffs, Prentice-Hall, USA.
- Johnson, F., and Jajodia S. 1998. Steganalysis of Image Created Using Current Steganography Software. *Information Hiding: Second International Workshop*, Lecture Notes in Computer Science, Springer-Verlag, 1525:273-289.
- Johnson, F., and Jajodia, S. 1998. Exploring Steganography: Seeing the Unseen. *IEEE Computer*. 87: 26-34.
- Ken, P. 1991. *Advanced Digital Audio*. SAMS, USA.
- Khalid, S. 2000. *Introduction to Data Compression*. Academic Press, USA.
- Kientzle, T. 1998. *A Programmers Guide To Sound*. Addison-Wesley Developers Press, USA.
- Lisa, M., Charles, B., and Charles, R. 1999. Spread Spectrum Image Steganography. *IEEE Transactions on Image Processing*, 8(8): 1075-1083.
- Michael, R. 1985. *Sound Production, Technical Notes for the Non-Technician*. UNESCO, UK.
- Mitchell, D., Ahmed H., and Launence B. 1998. Robust Audio Watermarking Using Perceptual Masking. *Signal Processing*. 66:337-355.
- Pennebaker, B., and Mitchell, L. 1993. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, USA.

Smith, A. 1998. *Information Hiding*. Lecture Notes in Computer Science, Springer-Verlage, UK.

Smith, A. 1996. *Information Hiding*. Lecture Notes in Computer Science, Springer-Verlage, UK.

Stefan, K., and Fabien, P. 2000. *Information Hiding techniques for steganography and digital watermarking*. Artech House Inc., UK.

Stolz, A.. 1993. *The Sound Blaster Book*, Abacus, USA

Tanenbaum, S. 1996. *Computer Networks*. Third Edition, A Simon & Schuster Company, USA.

## APPENDIX A

## Traditional Audio Tools

Software Products	Author / Company	Audio type(s) supported
Data Stash (Shareware)	Guan Inc.	Any binary
Hide4PGP (Freeware)	Heinz Repp	WAV and VOC
Invisible Secrets Pro (Shareware)	NeoByte Solutions	WAV
MP3Stego (Freeware)	Fabin Peticolas	WAV
Scramdisk (Freeware)	Sam Simpson	WAV
Steganos (Shareware)	Steganos GmbH	WAV and VOC
StegHide (Freeware)	Stefan Hetzl	WAV and AU
StegoWav (Freeware)	Peter Heist	WAV
S-Tools (Freeware)	Andrew Brown	WAV
SureSign (Shareware)	Signum Tech.	WAV

579071

## Appendix B Source Code

```

% INPUT
% Cover represent Wave sequence
% Embed represent Text or Image sequece (Image must be gray scale)
% Sr  represent Sample Rate
% Jump represent the Gap between Wave Values, default value is 2

% OUTPUT
% t  represent Stego-Wave
% Key represent Private Key

function [t,Key,DifEmbd]=embt(Cover,Embed,Sr)
t=0;
Key=zeros(1,3); %init. Private Key

message = 'The Embedded Size Greater than Cover Size, try to select another Cover
media';
mesg = 'Error in Function';
if nargin<3
    msgbox (strcat(mesg,' ,Function Argument must be 4'),'Error','Error');
    return
end

[Re,Ce]=size(Embed);
SizE=Re*Ce;

[Rc,Cc]=size(Cover);
SizC=Rc*Cc;
Jump=fix(SizC/SizE);

X_Cover=Av(Cover,Sr,Jump,SizE); %Compute The Avarage Values of Cover Wave
Media

if (SizE*2)>=fix(SizC /2) %Compare Between Embedded and Cover Size
    msgbox(message,'Embedding Process','Error');
    return
end
if ischar(Embed)
    disp('Embedding Text Media in Wave Media');
    Emb=Embed';
    Emb=upper(Emb);
    for i=1:SizE

```

```

    if Emb(i)=='
        Emb(i)='@';
    end
end
disp(Emb)
Emb=double(Emb); %Convert Char to Num.
disp(Emb)
Typ=1;
else
    disp('Embedding Image Media in Wave Media');
    Emb=Embed(:);
    Typ=0;
end

DifEmbd=DifWave(Emb); %Compute the diff. between the Embed Num.

Key(1)=DifEmbd(1);
Key(2)=Re;
Key(3)=Ce;
Key(4)=Jump;
Key(5)=Typ;

if Cc==2 %Conver 2 Channel into 1 Channel
    %X_Cover=Wav8bits(X_Cover);
    X_Cover=X_Cover(:);
end

t=X_Cover;
n=Jump;
for i=2:Size
    t(n)=t(n)+DifEmbd(i);
    n=n+Jump;
end

if Cc==2 %Conver 1 Channel into 2 Channel
    %t=Wav16bits(t);
    t=reshape(t,Rc,Cc);
end
return

```

```

% Extracting Function
% INPUT
% Stego represent Stegi Wave sequence
% Key represent Private Stego Key

% OUTPUT
% t_wave represent Cover Wave
% m_Embed represent Embedding Message (Text or Image)

function m_Embed =extt(Stego,Key)

message = 'Error in Function';
if nargin<2
    msgbox (strcat(message,' ,Function Argument must be 2'),'Error','Error');
    return
end
m_Embed(1)=Key(1); %First Byte in Private Key represent First Byte in Embedding
Message
Jump=Key(4);
SizE=Key(2)*Key(3);
n=Jump;
Cc=size(Stego,2);
if Cc==2 %Conver 2 Channel into 1 Channel
    Stego=Stego(:);
    %Stego=Wav8bits(Stego);
end

for i=2:SizE
    Avr=fix((Stego(n-1)+Stego(n+1))/2);
    m_Embed(i)=Stego(n)-Avr;
    n=n+Jump;
end
S=DifWave_Inv(m_Embed);
for i=1: SizE
    if S(i)==64
        S(i)=32;
    end
end
if Key(5)==1

    m_Embed=char(S');
else
    m_Embed=reshape(S,Key(2),Key(3));
end
return

```

```

function fig = dbP(dB,SF)
%load dbP;
H=pwd;

H=strcat(H,'\dbP.m');
h3 = figure('Units','points', ...
    'Color',[0.8 0.8 0.8], ...
    'FileName',H, ...
    'PaperPosition',[18 180 576 432], ...
    'PaperUnits','points', ...
    'Position',[88.5 159 420 226.5], ...
    'Tag','Fig1', ...
    'ToolBar','none','Name','decibel (dB)');

h1 = uicontrol('Parent',h3, ...
    'Units','points', ...
    'BackgroundColor',[0.752941176470588 0.752941176470588 0.752941176470588], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'ForegroundColor',[1 1 0.501960784313725], ...
    'ListboxTop',0, ...
    'Position',[0 0 80 13], ...
    'String','Quit', ...
    'Tag','Pushbutton1','CallBack','close','BackgroundColor',[ 1 0.501960784313725
0.752941176470588 ]);
%if nargout > 0, fig = h0; end
S=size(dB,1)/2;
plot((0:S-1)/S*SF,dB(1:S));
xlabel('Frequency (Hz)');
ylabel('Signal Power dB');
title('Spectrum');
grid on;
set(h3,'MenuBar','none');
set(h3,'Name','Frequency Analysis','NumberTitl','off');
set(h3,'Resize','off');

```

```
function d=diffx(G,f1,f2)
[R,C]=size(G);
d=G;
for i=1:R
    if (G(i)>=f1 & (G(i)<=f2))
        d(i)=0;
    end
end
return
```



```
function d=diffx(G,fc1,fe1,fc2,fe2)
[R,C]=size(G);
d=G;
for i=1:R
    if (G(i)<=fc1 & (G(i)>=fe1))
        d(i)=0;
    end
    if (G(i)>=fc2 & (G(i)<=fe2))
        d(i)=0;
    end
end
return
```

% this function try to subtract each position by next and previous position to extract original num.

% Ex: 23 2 -4 -> 23 25 21

```
function S=DifWave_Inv(Strm)
```

```
St=Strm(:);
```

```
R=size(St,1);
```

```
S(1)=St(1);
```

```
S(2)=S(1)-St(2);
```

```
for i=2:R-1
```

```
    S(i+1)=S(i)-St(i+1);
```

```
end
```

```
S=S';
```

```
return
```

%The Input of this function is Half of FFt of some array  
%The output is Complete FFt of some array  
%Ex: Ift=DoublFFt(Ft(1:51)) ->size of Ift = 100

```
function Ift=DoublFFt(Ft)
[R,C]=size(Ft);
Siz=size(Ft,(C>=R)+1); % this statm. select row or col size.
Ift=Ft;
for i=0:Siz-2
    Ift(100-i,1)=Ft(i+2,1);
end
return
```

```

% This function Embed selected SMALL wave file (E) in Long one (C).
% INPUT  C   = Long Wave file
%        E   = Embed wave file (C half size)
%        SmpRat= Sample Rate of E wave media
%        Shift = Find best match position to Embed the E media (1=True or 0=False)
%        Jump  = Number of Gap between C wave media
%
% OUTPUT Stego = Both C and E in One File
%        Key   = Private Key
%
%
% NOTE : C and E must be UN-NORMLIZED values
function [Stego,Key]=EmbedWaveNew(C,E,Jump,Shift,SmpRat)
DimC = size(C,2); %Check if C wave media
Key = zeros(8,1);
Stego = 0;
message = 'The Embedded Size Greater than Cover Size, try to select another Cover
media';
mesg = 'Error in Function';
if nargin<4
    msgbox (strcat(mesg,' ,Function Argument must be 5'),'Error','Error');
    return
end

[Key(2),Key(3)] = size(E); %Save the Size of E wave media
[C_S1,C_S2] = size(C);

C = C(:);
E = E(:);

[E_Size,tmp] = size(E);
[C_Size,tmp] = size(C);

if (E_Size*Jump) >= fix(C_Size /2) %Compare Between Embedded and Cover Size
    msgbox(message,'Embedding Process','Error');
    return
end

F_c = FindMax(C);

ShiftAccp = (E_Size*Jump)+F_c;
if (Shift==1)
    if ShiftAccp > fix(C_Size /2)
        msgbox(message,'Embedding Process','Error');
        return
    else

```

```

    E = ShiftM(E,F_c);
    [E_Size,tmp] = size(E);
end
end

MinE = min(E);
Key(1) = MinE;
Et=E+abs(MinE); % Shift All E elements (make all E elements Positive Numbers
Av_C = Av(C,1,Jump,E_Size); % Compute the Average value of C media

E_SRate = Normz(Et,SmpRat); % Normalize E wave media
dct_E = dct(E_SRate); %Compute the DCT for E wave media
Key(4) = dct_E(1);
Key(5) = Shift;
Key(6) = SmpRat;
Key(7) = Jump;
Key(8) = F_c;

n = Jump;
Stego = Av_C;
for i=2:E_Size
    Stego(n)=Stego(n)+dct_E(i);
    n=n+Jump;
end

if DimC == 2 %Conver 1 Channel into 2 Channel
    Stego=reshape(Stego,C_S1,C_S2);
end

return

```

```
function Rev=Filp(O,L)
y=1;
ii=1;
ix=1;
L=L+1;
for i=1:size(O,1)
    k(y,ix)=O(i);
    ii=ii+1;
    ix=ix+1;
    if ii==L
        y=y+1;
        ix=1;
        ii=1;
    end
end
Rev=k;
```

```
% this Function try to find the best match between the Embed(a) and cover(Long)
function B=FindMatch(a,Long,Th)
Sa=size(a,1);
SLong=size(Long,1);
X=0;
n=0;
Inc=fix(SLong/Sa);
B=0;
for i=1:Inc
    Tmp=Long(X+1:Sa+X);
    dif=abs(sum((Tmp-a)));
    if dif<=Th
        disp('Find Match');
        n=n+1;
        B(n)=i;
    end
    X=X+Sa;
end
return
```

```
function F=FindMax(H)
J=max(H);
F=find(H==J);
F=F(1);
return
```



```
function N=Normz(X,SmplRate)
N=X/SmplRate;
return
```

```
function O=odd(X)
if fix(X/2)==(X/2)
    O=0;
else
    O=1;
end
return
```

```
function PlayWave  
load a,b;  
  
wavplay(a,b,'sync');  
return
```

```
function UN=UNormz(X,SmplRate)
UN=fix(X*SmplRate);
return
```

```
%Extract Information and Play Wave File
```

```
function [a,b,c,d,dB,t]=WaveInf(Ss)
```

```
[a,b,c,d]=wavread(Ss);
```

```
save a,b;
```

```
a1=a(:,1);
```

```
S=char(Ss);
```

```
set(gcf,'MenuBar','none');
```

```
set(gcf,'Name','SteganoWave Appl.','NumberTitl','off');
```

```
set(gcf,'Resize','off');
```

```
h1 = uicontrol('Style','pushbutton', ...
```

```
    'Units','points', ...
```

```
    'BackgroundColor',[0 0.501960784313725 1 ], ...
```

```
    'ListboxTop',0, ...
```

```
    'FontSize',10, ...
```

```
    'FontWeight','bold', ...
```

```
    'Position',[25.25 0 125 18], ...
```

```
    'String','Play', ...
```

```
    'Callback','playwave');
```

```
h2 = uicontrol('Style','pushbutton', ...
```

```
    'Units','points', ...
```

```
    'BackgroundColor',[ 1 0.501960784313725 0.752941176470588 ], ...
```

```
    'FontSize',10, ...
```

```
    'FontWeight','bold', ...
```

```
    'ListboxTop',0, ...
```

```
    'Position',[213.25 0 120 18], ...
```

```
    'String','Quit', ...
```

```
    'Callback','close');
```

```
%dBt=10*log10(a/32.768); %Compute the dB values
```

```
%dBt=real(dBt);
```

```
%dB=dBt(dBt~=-Inf);
```

```
dB=abs(fft(a(:,1),1024)).^2/1001;% another way to compute dB and view the freq. of  
Wave File (Periodogram)
```

```
t=1/dB;
```

```
dB=10*log10(dB);
```

```
if d.fmt.nChannels==1
```

```
    plot(a1);
```

```
    Siz=size(a,1);
```

```
    WInfo(S,b,c,d,Siz*(d.fmt.nBitsPerSample/8)*d.fmt.nChannels+100); %100=file header
```

```
    dBP(dB,b);
```

```

    return
end
a2=a(:,2);
subplot(2,1,1);plot(a1);
subplot(2,1,2);plot(a2);

Siz=size(a,1);
WInfo(S,b,c,d,Siz*(d.fmt.nBitsPerSample/8)*d.fmt.nChannels+100);
dBP(dB,b);
return
%This Function Compute the PSNR,MSE,MAE, where PSNR represent the Signal-to-Noise
Ratio
%MSE (Mean Square Error) and MAE (Mean Absolute Error).
function [Psnr,Mse,Mae]=PSNR(A,B)

Psnr=0;
Mse=0;
Mae=0;
if A == B
    error('Waves are identical');
    return
end
ColA=size(A,2);
ColB=size(B,2);
if (ColA==ColB)+1==2
    A=A(:);
    B=B(:);
end

Mse = A - B;
Mae=mean(mean(abs(Mse)));
Mse=sqrt(mean(mean(Mse.^2)));
dB = 20*log10(1/(Mse)); % This Statement Compute Final PSNR;

Psnr=dB;

```

## إخفاء المعلومات في ملف صوتي باستخدام الستيجانوجرافي

إعداد

دعاء عبد اللطيف قاسم نصار

المشرف

الدكتور أحمد الجابر

### ملخص

إن عملية نقل المعلومات بصورة سرية عبر العالم أصبحت تحدي لدى الكثيرين من الباحثين، ولهذا فقد تم تطوير العديد من الطرق التقنية لإخفاء المعلومات، وأحد هذه الطرق الستيجانوجرافي. الستيجانوجرافي هو عملية إخفاء المعلومات بطريقة تكون غير مرئية.

في هذه الرسالة، طريقة جديدة للإخفاء تم استحداثها وتتكون من نظامين الأول يعمل على إخفاء النصوص والصور المرئية الغير ملونة داخل ملف صوتي، أما الآخر فيقوم بإخفاء ملف صوتي داخل آخر.

لقد حقق هذا النظام نتائج جيدة من حيث مقارنة الملف الأصلي مع الملف المحتوي على الإخفاء، بالإضافة إلى أن كمية المعلومات التي يتم إدخالها تصل إلى نصف حجم الملف الصوتي.