Republic of Iraq
Al-Nahrain University
College of Science

Audio Steganagraphy
In Wavelet Transform Domain

A THESIS
SUBMITTED TO THE
COLLEGE OF SCIENCE, Al-NAHRAIN UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE IN
COMPUTER SCIENCE

By

Muntasir Jaber Jawad

(B.Sc. 2002)
Email: muntasirjaber2000@yahoo.com

SUPERVISORS

Dr. Loay A. George **Dr. Ban N. Al-kallak**

Junuery 2005 **Thi-Alhija 1425**

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

وَيَسْـَٔلُونَكَ عَنِ ٱلرُّوحِ ۖ قُلِ ٱلرُّوحُ مِنْ أَمْرِ رَبِّى

وَمَآ أُوتِيتُم مِّنَ ٱلْعِلْمِ إِلَّا قَلِيلًا ﴿٨٥﴾

صدق ا العلي العظيم

الأسراء-٨٥

Dedication

To my family … with love and respect

To my teachers … with grateful and esteem

To my friends … with joy and delight

Muntasir

List of Symbols

Symbol	Description
T	Translation parameter
S	Scale parameter
$\psi(t)$	The transforming function
$\Psi(t)$	called the mother wavelet
x	Original audio signal
g	Highpass filter
h	Lowpass filter
L	Low frequencies subband
H	High frequencies subband
w_m	The maximum value can be coded using single codeword
w_r	The value of the last codeword
ω	The weight factor(which is a function of the sample value)
δ	The absolute value of sample value minus 128

WT	Wavelet Transform

List of Abbreviations

A/D	Analog to Digital
CMR	Compression Ratio
CWT	Continues Wavelet Transform
D/A	Digital to Analog
DCT	Discrete Cosine Transform
DSSS	Direct Sequence Spread Spectrum
DVD	Digital Versatile Disk
DWT	Discrete Wavelet Transform
FEM	Fixed-length Encoding Method
fmt	format subchunk
FHWT	Forward Haar Wavelet Transform
FT	Fourier Transform
GIF	Graphical Image Format
HAAS	Hiding Audio in Audio System
HAP	Hiding in Audible Parts
HAS	Human Auditory System
HEM	Hybrid Encoding Method
HWT	Haar Wavelet Transform
IHWT	Inverse Haar Wavelet Transform
IWT	Integer Wavelet Transform
JPEG	Joint Photography Expert Group
LSB	Least Significant Bit
LSBRT	Least Significant Bit with Recovery Technique
LSBWTD	Least Significant Bit in Wavelet Transform Domain
LZW	Lampel Ziv Weltch
MMSE	Modified Mean Square Error
MPSNR	Modified Signal to Noise Ratio
MSE	Mean Square Error
MSNR	Modified Signal to Noise Ratio
PCM	Pulse Code Modulation
PHWT	Packet Haar Wavelet Transform
PIWT	Packet Integer Wavelet Transform
PSNR	Peak Signal to Noise Ratio
SEM	S-shift Encoding Method
SNR	Signal to Noise Ratio
SS	Spread Spectrum
STFT	Short Time Fourier Transform
THWT	Tree Haar Wavelet Transform
TIWT	Tree Integer Wavelet Transform
WAV	Windows Audio Visual

Abstract

Steganography is the art of hiding information in ways that prevent its detection. A message in cipher text may arouse suspicion while an invisible message will not. Digital steganography uses a host data or message, known as a “container” or “cover” to hide another data or message called “secret” in it.

An audio in audio steganography system had been proposed in this thesis in order to embed a secret audio data in another cover audio data. In this system, the secret data is first transformed using wavelet transform and then the resultant coefficients have to be coded using one of the three coding methods (fixed length encoding method, S-shift coding method and hybrid coding method).

The next stage in this system is the embedding stage where the output of coding stage (a stream of bits) is embedded in the cover data. Three embedding methods were implemented in the proposed system (least significant bit insertion in wavelet transform domain, two least bits insertion in time domain with recovery technique and hiding in audible parts). A modified fidelity criteria were derived by adding some modifications to the standard fidelity criteria to be more precise for audio and the modified criteria are called (modified Mean Square Error, modified Signal to Noise Ratio and modified Peak Signal to Noise Ratio) and they also used through the system testing stage.

All of the fidelity criteria obtained in the tests have indicates good results for PSNR and its modified version (50 dB). The reconstructed data is exactly the same as secret data if the integer wavelet transform is used before the coding stage while a small unrecognizable error may done when the Haar wavelet transform is used before the coding stage.

Table of Contents

Chapter One
Introduction

1.1 Overview

Digital multimedia communication is of the essence to the Internet. In numerous applications it is desired or required that the communication be private or secure. The two most common methods for secure communications are cryptography and steganography [Ira00]. In cryptography the secret message (of any media format) is encrypted, while in steganography the message or payload is hidden, in an imperceptible manner, in a "carrier" media. Steganography is an alternative to cryptography, because of the ease to develop customized steganographic systems and the appeal that, unlike cryptography, the secure communication is not apparent to any 3rd party [Jia03]. In the study of communications security, cryptography techniques received more attention from the research community and from industry than information hiding, but in the recent years a rapid growth of this discipline was seen. The reason of this growing in interest is due to the availability of multimedia content in digital form. Digital representation of signals brings many advantages when compared to analog representation among these advantages: [Ste99]

1. Lossless recording and copying.
2. Convenient distribution over networks.
3. Easy editing and modification.
4. Easily searchable archival.
5. Durable.

1.2 Information Hiding

The term information hiding refers to both watermarking and steganography. Both steganography and watermarking techniques are used to imperceptibly convey information by embedding it into the cover media. Steganography typically relates to cover point to point communication between two parties. Steganographic methods are usually not robust against modification or have only limited robustness, and protect the embedded information against the technical modifications that may occur during transmission or storage (like format conversion, compression or digital to analog conversion) [Ste99].

1.3 Steganography Definition

Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, here the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present.

Steganography can also be explained literally through its Greek meaning, “Covered Writing”. The art of steganography allows messages to be passed covertly without attracting the attention of a third party. These steganographic techniques range from physically hiding the messages, using invisible ink or microdots, to advanced technological methods of watermarking [Joh00].

1.4 Steganography Advantages and Disadvantages [Deb03]

a. Advantage

The advantage of using steganography is to hide information, such that the transmission of messages is transparent to any given viewer. Messages can be hided in different formats that are undetectable and un-readable to the human eye. Steganographic technologies are very important in Internet privacy today. With the use of steganography and encryption, corporations, governments, and law enforcement agencies can communicate secretly.

Encryption protects data and can be detected; the only thing missing is the secret key for decryption. Steganography is harder to detect under traditional traffic pattern analysis, while steganography enhances the privacy of personal communication. Since encryption can be detected and some governments prohibit the use of encryption, steganography can be used to supplement encryption. Additional layers of security are of benefit to secrecy. If a steganographic message is detected, there still is the need for the encryption key.

The method of encrypting a message and then using steganography is most widely used by steganographers.

b. Disadvantages

One of the biggest disadvantages is that quite frequently the size of a secret message is usually larger than the original cover. There can be color changes, or detectable sound changes, there are evident, especially if well-known images or audios are chosen as the steganographic cover. Another issue to mention, text messages are limited in size for the hiding of data, they need redundant data to replace a secret message. Changing the type of the format or replacing the readable text can alter text messages.

Through the use of the new technology, some Internet firewalls can detect steganographic messages. As this technology evolves detecting steganographic messages, it can be considered as a drawback because an important message may be deleted or quarantined and this message may be the one that will save a country.

1.5 Steganography Uses

The use of steganography undeniably means of dishonest activity, but there is a peaceful aspect to consider. Steganography is used in map making, where cartographers add a nonexistent street or lake in order to detect copyright offenders. Or similarly, fictional names are added to mailing lists to catch unauthorized resellers. Modern techniques use steganography as a watermark to inject encrypted copyright marks and serial numbers into electronic mediums such as books, audio, and video. For example, DVD recorders detect copy protection on DVDs that contain embedded authorizations.

Potential uses of steganography are undoubtedly vast. Companies could advertise public Web pages containing private, hidden text that only internal members could intercept. An attempt to decipher the hidden text would be unwarranted since no encryption (or code) was used. Steganography could also be used to hide the existence of sensitive files on storage media. This would entail a cover folder and an embedded hidden folder [Jam00].

1.6 Aim of Thesis

The present work aims to develop a system for hiding audio signal in another audio signal using Wavelet Transform. The secret audio file data first coded (compressed) then embedded in the wavelet domain of the cover data. The receiver can extract the coded secret data and then decoded (decompressed) it in order to reconstruct the original secret file. The proposed system uses the concept of blind steganography (i.e. the receiver does not need to overhead information in order to perform the extraction stage, only the secret key is required, and it should be the same like that used by the sender).

1.7 Related Work

A lot of research work conducted by several researchers concerned with developing information hiding techniques have been published. These researches tried to insert new additional features focus on increase the system robustness and invisibility, some of these researches are summarized below:

- Xin Li and Hong H. Yu [Xin99], in this research the audio data hiding using spread spectrum (SS) technique in subband domain was introduced. On transparency part, a novel method was proposed to tune the psycho-acoustic models, in order to employed in audio compression to control the audibility of introduced distortion for data hiding purpose. On robustness part, the feature selection and synchronization problem was studies in order to maximize the survivability of the embedded data. The results showed that the mid-band coefficients are appropriate features for data embedding. And it proposes to track /recover synchronization of audio signal before the detection, facilitating the extraction of the embedded data. Experiment results have shown that the data hiding scheme in subband domain can survive a wide range of attacks while providing transparent audio quality and abundant embedding capacity (>20bps).

- Jianyun Xu, Andrew H. Sung, Peipei Shi, Qingzhong Liu [Jia03], proposed a system for "text steganography using wavelet transform". This research implies an algorithm to limit errors in lossy transforms to achieve high capacity text hiding in image files using discrete Haar wavelet transform (DHWT). It also discusses robust text steganography using multiple-level lossless DHWT. Experimental results validated the method for high capacity plain text hiding and demonstrated that lossless recovery of the hidden text from JPEG images with compression rate as high as 67% is possible.

- Yasmeen I. Dieab [Yas03] proposed a system to embed a digital watermark in audio signal while retained perceptual to the listener. The system uses two techniques: Low Bit Encoding in time domain and the human auditory characteristics in frequency domain. In the frequency domain method the Fast Fourier Transform (FFT) with segmentation is used to embed the watermarks. The imperceptuality of the watermarking is measured by using the PSNR metrics, it have been proven that it has a good quality (PSNR=40 dB).

1.8 Thesis Layout

This thesis is organized in five chapters. The contents of these chapters are:

- **Chapter Two: (Theoretical Background)** in this chapter information hiding techniques are presented. Information hiding features and applications are also explained. This chapter also presents the conceptual concepts of the wavelet transform, and the fidelity measures are also explained.
- **Chapter Three: (System Design and Implementation)** it dedicated to introduced the proposed system design steps, and the practical work to implement these steps. Each practical step is discussed with its related algorithms.
- **Chapter Four: (Experimental Results and System Evaluation)** this chapter contains the results of a comprehensive tests performed on the proposed system using different test samples. The results are discussed to assess the overall system efficiency.
- **Chapter Five: (Conclusions and Future Work Suggestions)** this chapter is dedicated to introduce some conducted conclusions that are derived from the test results, also some new ideas that can be added to the suggested system as future work, are given in this chapter.

Chapter Two
Theoretical Considerations

2.1 Introduction

This chapter concerns with the main fundamental concepts needed to understand the ideas applied in the proposed hiding system. In fact two main concepts were covered, they are: Information Hiding Techniques and Wavelet Transform. Data compression and fidelity measures were also explained.

2.2 Information Hiding Techniques

There are several information hiding techniques that should classified according to the media where the information are hidden.

2.2.1 Hiding in Text

Methods such as line-shift coding, word shift coding, and feature coding are the most commonly used methods to hide data in text. When using a text data as a host media, the embedded data is usually a codeword that is hidden within the text by altering its different textual features. The three methods listed above determine what feature is to be changed. To encode the codeword, each bit of the codeword is applied using one of the three methods [Sel99]:-

a) Line-Shift Coding

Line-shift coding is very easy to perform and is considered as the most resistant to degradation due to copying. In line-shift coding, the lines of text are shifted vertically to encode the document. By determining which lines have been shifted, the encoded bits can be discovered. Although this method withstands copying, the human eye and other measurements can easily detect it. It can also be easily defeated through replacing or reformatting of the text [Sel99].

b) Word-Shift Coding

Word-shift coding can also be easily done. In word-shift coding, codewords are coded into a document by shifting the horizontal location of words within the lines of text. In doing so, the appearance of natural spacing must be maintained in order not to arouse suspicion. By determining the location where unnatural spacing has occurred, the encoded bits can be revealed.

c) Feature Coding

Feature coding is another way of embedding data into a text file. In feature coding, certain text features are altered depending on the embedded data. For example, one type of feature coding would be extending the vertical lines of characters such as “l”, “d”, “b” and “h” [Sel99]. In order for this type of feature coding to work, the text must be altered by randomizing the lengths of the vertical lines before applying this algorithm. The randomness will help the text look less suspicious to its readers. In order to decode this algorithm, the text, after the randomization, but before the algorithm application, can be compared with the message containing the embedded data to retrieve the encoded bits. This type of feature coding can be easily defeated if the vertical line length is adjusted to a fixed length before the file is opened [Da00].

2.2.2 Data Hiding in Image

Text files are not the only files that can be used for host data. Images are also another popular source for hidden data. The same techniques can be applied to these images to hide information. These techniques include least significant bit insertion and the use of algorithms and transformations [Sel99].

a) Least Significant Bit Insertion

Least significant bit insertion, or LSB, is one of the most common techniques used to hide information in images. When working with 24-bit pixel images, three bits can be encoded into each pixel. Because the least significant bits are the ones being altered, the change is difficult to determine by the viewer. However, when working with 8-bit pixel images, this method becomes harder to implement because a change of a bit may result in a change of an entirely different color. Although this technique is popular due to its simplicity, it is also one of the easiest methods to accidentally alter. When transforming images to different formats, such as from GIF to JPEG, lossy compression occurs. Lossy compression, which is standard for JPEG images, uses high compression [Sel99]. However, this high compression may cause some change from the original image. Although it is almost an exact replica of the original image, the bits from the original image cannot be guaranteed.

b) Algorithms and Transformations

Other compression algorithms and transformations are also used when dealing with images and their usage in hiding data. Some of the most popular methods are the Patchwork method, the discrete cosine transform (or DCT), and the Fourier transform. The Patchwork method takes the advantage of the fact that the human eye cannot easily detect varying amounts of light [Kat00]. The Patchwork method gets its name by "using redundant pattern encoding to repeatedly scatter hidden information throughout the cover image, like patchwork" [Sel99]. One advantage of this technique is that it can hide a small message many times throughout an image. Because of this, even when an image is cropped or rotated, the chances of one instance of the encoded message still being intact are very high.

However, in other compression algorithms, message retention is not always guaranteed. As stated earlier, when images such as JPEG images are compressed, data may be lost. There are many different compression algorithms that perform this type of lossy compression such as the discrete cosine transform, the wavelet transform, and the Fourier transform.

The discrete cosine transform (DCT) is an algorithm that finds a set of coefficients that allow a small set of cosine functions to approximate a portion of the image [Yan01]. For example, the JPEG algorithm uses 8x8 blocks of pixels and fits them with a set of cosine functions that can approximate a section of the image. The DCT finds a different coefficient for each function so that the weighted sum of the functions adds up to recreate the original 8x8 block of pixels [Yan01].

The wavelet transform and Fourier transform methods use complicated mathematical formulas in order to find the coefficients, in which, to map a signal into the frequency domain [Kat00]. More information on the different mathematical formulas can be found in [Kat00] and [Yan01].

2.2.3 Data-Hiding in Audio

Audio files can also be used to hide information. Steganography is often used to copyright audio files to protect the rights of music artists. Techniques such as least significant bit insertion, phase coding, spread spectrum coding, and echo hiding can be used to protect the content of audio files. The biggest challenge face all these methods is the sensitivity of the human auditory system or HAS [Kat00]. Because the HAS is so sensitive, people can often pick up randomly added noise making it hard to successfully hide data within audio data. To fully understand the different techniques of hiding information in audio data, transmission of audio signals must first be understood. When working in audio the transmission medium must always be considered.

The transmission medium of an audio signal refers to the environment in which a signal might go through to reach its destination. Bender and his colleagues categorize the possible transmission environments into the four following groups [Sel99]:

1. Digital end-to-end environment, where the sound files are copied directly from one machine to another.
2. Increased/decreased resampling environment, where the signal is resampled to a higher or lower sampling rate.
3. Analog transmission and resampling, where a signal is converted to an analog state, played on a clean analog line, and resampled.
4. "Over the air" environment, where the signal is played into the air, passed through a microphone.

By understanding the different mediums in which audio signals may travel, the appropriate technique for embedding data in audio files can be determined.

The most commonly used methods for hiding data in audio media are the following methods:

a) Least Significant Bit Insertion

Like image files, the least significant bit insertion method can also be used to store data in the least significant bit of audio files. However, like image files, by using this method, the hidden data can be easily destroyed and detected. Resampling and channel noise may alter the hidden data, while changing the least significant bit may introduce audible noise [Sel99]. Information may also be destroyed through compression, cropping, or A/D, D/A conversion [Yan01]. Although this technique is simple to perform, its lack of dependability makes other methods more appealing.

b) Phase Coding

It is another technique used to hide data in audio files. This is done by substituting of the phase of an initial audio segment with a reference phase that represents the data. The phase of the following segments is adjusted accordingly to preserve the relative phase between segments [Yan01]. The steps to phase coding are as follows [Kat00]:

1. The original sound sequence is broken into a series of N short segments.
2. A discrete Fourier transform is applied to each segment.
3. The phase difference between each adjacent segment is calculated.
4. For segment S_0, the first segment, an artificial absolute phase P_0 is created.
5. For all other segments, new phase frames are created.
6. The new phase and original magnitude are combined to get a new segment, S_n.
7. The new segments are concatenated to create the encoded output.

To enable the receiver to decode the hidden data, one must know the length of the segments, the discrete Fourier transform points, and the intervals in which the data are hidden. Phase coding is one of the most effective schemes in terms of the signal-to-perceived noise ratio because listeners often do not hear a difference in the altered audio file when the phase shift is smooth [Yan01].

c) Spread Spectrum Coding

Spread spectrum coding can also be used to hide data in audio files. Usually when audio files travel through communication channels, the channels try to concentrate audio data through narrow regions of the frequency spectrum in order to conserve bandwidth and power [Sel99]. However, this technique requires the embedded data to be spread across the frequency spectrum as much as possible. Unlike the LSB insertion, spread spectrum coding uses the entire spectrum of the file to embed data [Fer98]. Many methods can be used to spread the embedded data over the frequency spectrum. Direct sequence spread spectrum (DSSS) encoding spreads the signal by multiplying it by a certain maximal length pseudorandom sequence called chip [Sel99]. Unfortunately, like the LSB method, DSSS may add random noise that the listener can detect. For frequency hopped spread spectrum encoding, the original audio signal is divided into small pieces and each piece is carried by a unique frequency [Yan01]. The main advantage of using spread spectrum coding is its resistance to modification. Because the embedded data is spread throughout the cover data, it would be difficult to modify the embedded data without causing noticeable harm to the cover data.

d) Echo Data Hiding

Echo data hiding hides data in a host signal by introducing an echo [Kat00]. The embedded data is hidden by varying three parameters of the echo: initial amplitude, decay rate, and delay [Kat00]. As the timing between the original signal and echo decreases, the two signals may blend, making it hard for the human ear to distinguish between the two signals. The value of the hidden data corresponds to the time delay of the echo and its amplitude. By using different time delays between the original signal and the echo to represent binary one or zero, data can be embedded into the audio file. To embed more than one bit, the original signal is divided into smaller segments and each segment can then be echoed to embed the desired bit. The final cover data consists of the recombination of all the independently encoded segments [Kat00]. Echo hiding works particularly well with high quality audio files. Audio files with no additional degradation and no gaps of silence are preferred when using this technique [Sel99].

2.3 Information Hiding Features and Applications

a) Features

Data-hiding techniques should be capable of embedding data in a host signal with the following restrictions and features:

1. The host signal should be nonobjectionally degraded and the embedded data should be minimally perceptible. (The goal is for the data to remain *hidden*. We will use the words *hidden*, *inaudible*, *imperceivable*, and *invisible* to mean that an observer does not notice the presence of the data, even if they are perceptible.)
2. The embedded data should be directly encoded into the media, rather than into a header or wrapper, so that the data remain intact across varying data file formats.
3. The embedded data should be immune to modifications ranging from intentional and intelligent attempts at removal to anticipated manipulations, e.g., channel noise, filtering, resampling, cropping, encoding, lossy compressing, printing and scanning, digital-to-analog (D/A) conversion, and analog- to-digital (A/D) conversion.
4. Asymmetrical coding of the embedded data is desirable, since the purpose of data hiding is to keep the data in the host signal, but not necessarily to make the data difficult to access.
5. Error correction coding may be used to ensure data integrity. It is inevitable that there will be some degradation to the embedded data when the host signal is modified.
6. The embedded data should be self-clocking or arbitrarily re-entrant. This ensures that the embedded data can be recovered when only fragments of the host signal are available, e.g., if a sound bite is

extracted from an interview, data embedded in the audio segment can be recovered. This feature also facilitates automatic decoding of the hidden data, since there is no need to refer to the original host signal (this feature is vital in watermarking) [Ben96].

b) Applications

Trade-off exists between the quantity of embedded data and the degree of immunity to host signal modification. By constraining the degree of host signal degradation, a data-hiding method can operate with either high-embedded data rate, or high resistance to modification, but not both. As one increases, the other must decrease. While this can be shown mathematically for some data-hiding systems such as a spread spectrum, it seems to hold true for all data-hiding systems. In any system, you can trade bandwidth for robustness by exploiting redundancy.

The quantity of embedded data and the degree of host signal modification vary from application to application. Consequently, different techniques are employed for different applications. Several prospective applications of data hiding are discussed in this section.

An application that requires a minimal amount of embedded data is the placement of digital watermark. The embedded data are used to place an indication of ownership in the host signal, serving the same purpose as an author's signature or a company logo.

A second application for data hiding is tamper-proofing, It is used to indicate that the host signal has been modified from its authored state. Modification to the embedded data indicates that the host signal has been changed in some way.

A third application, feature location, requires more data to be embedded. In this application, the embedded data are hidden in specific locations within an image. It enables one to identify individual content features, e.g., the name of the person on the left versus the right side of an image. Typically, feature location data are not subject to intentional removal. However, it is expected that the host signal might be subjected to a certain degree of modification, e.g., images are routinely modified by scaling, cropping, and tone scale enhancement. As a result, feature location data hiding techniques must be immune to geometrical and nongeometrical modifications of a host signal [Ben96].

2.4 Digital Sound Representation

When developing a data hiding method on sound waves, like speech or music, one of the first considerations is how does sound is represented digitally. Audio refers to the sound within the human hearing range (20 Hz to 20 KHz). An audio signal in nature is analog, analog sounds are waves detected by human ears. These waves are continues in both time and amplitude. Amplitude represents the height or (volumes), of the sound [Kie98, Dec99]. The analog signal should be converted to digital form to be stored and processed by computers and transmitted through computer networks. An A/D (analog to digital) conversion consists of two steps: sampling and quantization.

1. **Sampling**: Sampling or approximating involves periodically measuring the analog signal and use these measurements (samples) instead of the original signal; a sampled wave is shown in figure (2.1).

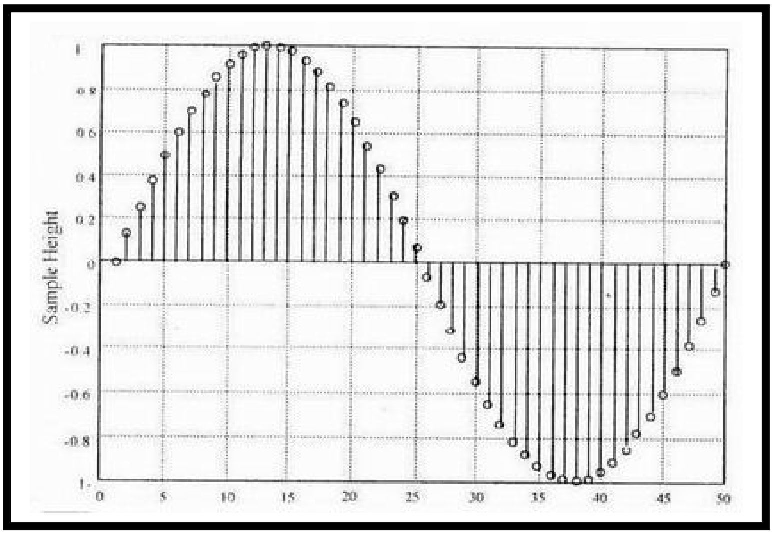


Fig (2.1) Sampled wave

Usually samples are stored as binary numbers, but they can be stored in other ways. A very well known way is to represent each sample with a series of pulses that represent it's binary code, such representation called Pulse Code Modulation (PCM).

There are various modulation types, but PCM is the widely used in digital audio. For a programmer a various modulation techniques are irrelevant. In a computer's memory, the successive binary values are simply stored as numbers. For most programmers PCM can be thought of as that shown in figure (2.2) [Kie98, Qu96].

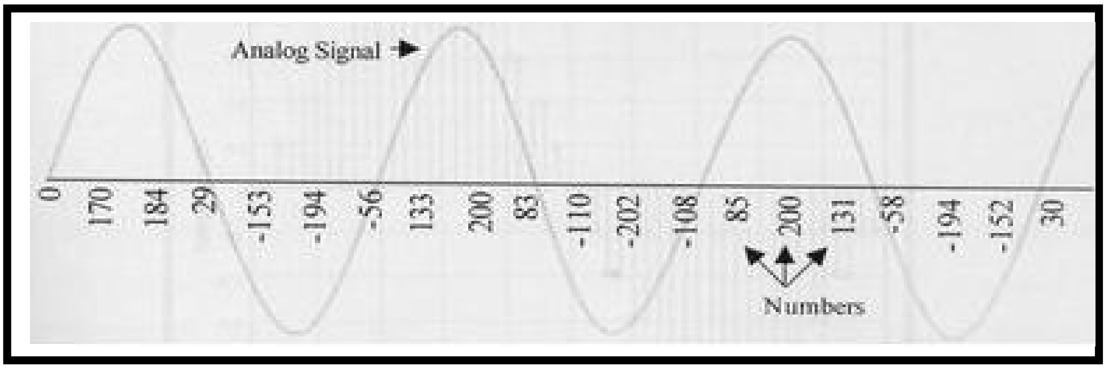


Fig (2.2) PCM for the computer programmer

2. **Quantization:** to quantize a signal means to determine the signal's value to some degree of accuracy. Because the finiteness of computer ability, the digital representation is also finite. For example if an 8-bit or 16-bit integers were used, either 256 or 65,536 discrete integer sample value can be obtained, but the original samples are not integers. The process of rounding the exact sample value to less-precise value is referred to as quantization [Aud03].

2.5 Data Compression

Data compression is a type of data encoding, one that is used to reduce the size of data. Other type of data encoding include encryption (Cryptography) and data transmission. There are many known method for data compression. They are based on different ideas, that are suitable for different types of data, and produce different results, but they are all based on the same principle, namely they compress data by removing redundancy from the original data in the source file. Compression uses means of encoding to eliminate the redundancy, thereby effectively reducing the size of the data traveling over the communications link or being stored in the repository. This leads to several applications that address the needs of today's networking applications [Tib03].

In order to be useful, a compression algorithm has a corresponding decompression algorithm that reproduces the original data from the compressed data. Many types of compression algorithms were developed, these algorithms fall into two broad types: Lossless algorithms and Lossy algorithms [Aud03].

a) 2.5.1 Lossless Compression [Sal98]

Lossless techniques are capable to recover the original representation perfectly. However the lossy compression provides higher compression ratios (ratio of uncompressed data to compressed data) by losing some information. When the compressed stream is decomposed, the result is not identical to the original data stream. These methods make sense especially in compressing images, moving pictures or sounds. If the loss of data is small, we may not be able to tell the difference. Constructing text files (especially files containing computer programs) may become worthless if even one bit gets modified. Such files should be compressed using lossless compression methods.

The most common lossless compression techniques are:

b) **Huffman Coding.**

c) **Arithmetic Coding.**

d) **Lampel_Ziv_Weltch (LZW).**

e) **Run Length.**

f) **S-Shift Coding.**

2.5.2 Lossy Compression [Ibr04]

Lossy compression method is based on compromising the accuracy of the reconstructed signal in order to increase compression. If the resulting distortion (which may or may not be subjectively detectable), can be tolerated, the increase in compression can be significant. The main features of this type of compression are as follows:

1. Allows a loss in the actual signal.
2. Original signal can not be recovered exactly from compressed data.

Lossy compression is useful for broadcast television, video conferencing, and facsimile transmission.

2.6 Wavelet Transform

Fourier transform is based on spectral analysis; it is the dominant analytical tool for frequency domain analysis. However, Fourier transform cannot provide any information of the spectrum changes with respect to time. Fourier transform assumes the signal is stationary, but real signals are always non-stationary. To overcome this deficiency, a modified method (called short time Fourier transform) allows to represent the signal in both time and frequency domain through time windowing function [Ibr04].

The window length determines a constant time and frequency resolution. Thus, a shorter time windowing is used in order to capture the transient behavior of a signal; we sacrifice the frequency resolution. The nature of the real signals is nonperiodic and transient (such as sound, image and video signals), such signals cannot easily be analyzed by conventional transforms. So, an alternative mathematical tool- wavelet transform must be selected to extract the relevant time-amplitude information from a signal [Wah02]. A continuous-time wavelet transform of *x(t)* is defined as:

$$cwt_x^{\psi}(T,S) = \Psi_S^{\psi}(T,S) = \frac{1}{\sqrt{|S|}} \int x(t)\psi^*\left(\frac{t-T}{S}\right)dt \quad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.1$$

Where

T represents the translation parameter.

S represents the scale parameter.

$\psi(t)$ represents the transforming function.

$\Psi(t)$ is also called the mother wavelet. The term wavelet gets its name due to two important properties of the wavelet analyses as will explained below.

The term wavelet means a small wave. The smallness refers to the condition that this (window) function is of finite length. The wave refers to the condition that this function is oscillatory. The term mother implies that the functions with different region of support that used in the transformation process are derived from one main function, which is called the mother wavelet. [Pol98]

2.6.1 Why Wavelet Analysis Effective

Wavelet transforms have proven to be very efficient and effective in analyzing a very wide class of signals and phenomena. The properties that give the effectiveness are:

a. The wavelet expansion allows a more accurate local description and separation of signal characteristics. A Fourier coefficient represents components that last for all time and, therefore, temporary events must be described by the phase characteristics that allow cancellation and reinforcement over large time periods. Wavelet expansion coefficients represent a component that itself local and easier to interpret. The wavelet expansion may allow a separation of components of a signal that overlaps in both time and frequency.

b. Wavelets are adjustable and adaptable. Because there is not just one wavelet, they can be designed to fit individual systems that adjust themselves to suit the signal.

c. The generation of the wavelets coefficients is well matched to the digital computers. There are no derivatives or integrals, just multiplications and additions operations that are basic to the digital computer.[Bur98]

2.6.2 Discrete Wavelet Transform (DWT)

A time-scale representation of a digital signal is obtained using digital filtering techniques. The heart of the DWT implies two filters h and g, low pass and highpass respectively. The block diagram of one level DWT is shown in figure (2.3). The one dimensional signal, x, is convolved with high pass filter to analyze the high frequencies, and it is convolved with low pass filter to analyze the low frequencies, and each result is down sampled by two, yielding the transformed signal $\mathbf{x_g}$ and $\mathbf{x_h}$. A DWT is obtained by further decomposing the low pass output signal $\mathbf{x_h}$ by means of a second identical pair of analysis filters. This process my be repeated, and the number of such stages defines the level of the transform.

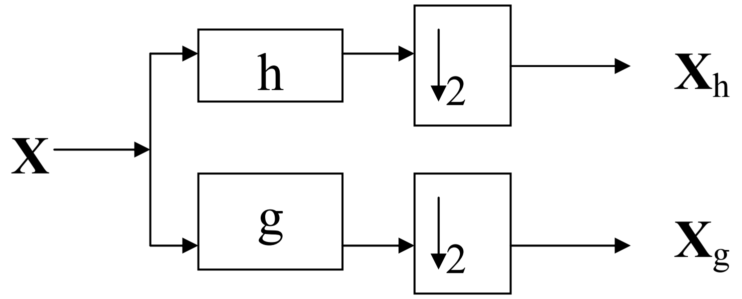


Figure (2.3) one level wavelet decomposition

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by the up-sampling and down sampling operations (sub-sampling). Sub-sampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. Up-sampling a signal correspond to increasing the sampling rate of a signal by adding new samples to the signals.

A half band low pass filter removes all frequencies that are above half of the highest frequency in the signal. Note that the low pass filtering removes the high frequency information, but leaves the scale unchanged. Only the sub-sampling process changes the scale. Resolution on the other hand is related to the amount of information in the signal, and therefore it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies [Bur98].

The DWT analyze the signal at different frequency bands with different resolutions by decomposing the signal into coarse approximation and detail information. DWT employs two sets of functions, called the scaling function and wavelet function, which are associated with lowpass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive lowpass and highpass filtering of the time domain signal. The original signal x(n) is first passed through a half band highpass filter g(n) and lowpass filter h(n). After filtering half of the samples can be eliminate. The signal can therefore be subsampled by two. These constitute one level of decomposition and can mathematically be expressed as follows:

$$\left.\begin{aligned} y_{high}(k) &= \sum_n x(n).g(2k-n) \\ y_{low}(k) &= \sum_n x(n).h(2k-n) \end{aligned}\right\} \quad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.2$$

Where $y_{high}(k)$ and $y_{low}(k)$ are the output of the highpass and lowpass filters respectively, after subsampling by 2.

The above procedure, which also known as subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of samples (and hence half the time resolution). Figure (x) illustrate this procedure, where x(n) is the original signal to be decomposed, and h(n) and g(n) are lowpass and highpass filters respectively [mus01].

2.6.3 Haar Wavelet Transform (HWT)

The oldest and most basic of the wavelet systems has constructed from the Haar basis function. The equations for forward Haar wavelet transform and inverse Haar wavelet transform will shown in the next two sections.

2.6.3.1 Forward Haar Wavelet Transform (FHWT) [Jia03]

Given an input sequence (x_i) i=0…N-1, it is FHWT produce (L_i) i=0…N/2-1 and (H_i) i=0…N/2-1 by using the following transform equations:

(a) If N is even

$$L(i) = \frac{x(2i) + x(2i+1)}{\sqrt{2}}$$

$$H(i) = \frac{x(2i) - x(2i+1)}{\sqrt{2}}$$

(b) If N is odd

$$\left.\begin{aligned} L(i) &= \frac{x(2i) + x(2i+1)}{\sqrt{2}} \\ H(i) &= \frac{x(2i) - x(2i+1)}{\sqrt{2}} \end{aligned}\quad \text{For } i = 0\ldots(n-1)/2 \right\} ,\ldots\ldots\ldots2.4$$

$$\left.\begin{array}{l} L(\frac{N+1}{2}) = x(N-1)\sqrt{2} \\ H(\frac{N+1}{2}) = 0 \end{array}\right\} ,\ldots\ldots 2.5$$

2.6.3.2 Inverse Haar Wavelet Transform (IHWT) [Jia03]

The inverse one-dimensional HWT is simply the inverse to those applied in the FHWT; the IHWT equation are:

(a) If N is even

$$\left.\begin{array}{l} x(2i) = \dfrac{L(i) + H(i)}{\sqrt{2}} \\ x(2i+1) = \dfrac{L(i) - H(i)}{\sqrt{2}} \end{array} \quad \text{For } i = 0...n/2-1 \right\} ,\ldots\ldots 2.6$$

(b) If N is odd

$$\left.\begin{array}{l} x(2i) = \dfrac{L(i) + H(i)}{\sqrt{2}} \\ x(2i+1) = \dfrac{L(i) - H(i)}{\sqrt{2}} \end{array} \quad \text{For } i = 0...\ (n-1)/2 \right\} ,\ldots\ldots 2.7$$

$$x(N-1) = L(\frac{N+1}{2})\sqrt{2} \quad ,\ldots\ldots 2.8$$

Where

N is the number of data samples.

L is the low frequencies subband.

H is the high frequencies subband.

2.6.4 Integer Wavelet Transform (IWT) [Eri97]

One level of IWT decomposes the signal into a low frequency part and high frequency part, both at lower resolutions. The low part can be used with the high part to reconstruct the original signal. This decomposition represents one level of the IWT.

Typically several levels of forward IWT can be computed, by iterating the procedure just described upon the low-frequency part (in a tree scheme), or reapply the procedure upon both low-frequency and high frequency parts (in a packet scheme).

In the next two paragraphs, we described one level of forward IWT and inverse IWT.

2.6.4.1 One-Dimensional Forward IWT

Given an input sequence (x_i) i=0…N-1, its forward IWT (y_i), i=0…N-1, will also be an integer sequence, its computation depends on the length N of the integer sequence weather its even or odd.

If the signal length N is even (i.e. n=2k), then the integer transform sequence is computed by implementing the following two steps:

(a) Determine the odd coefficients

$$\left.\begin{aligned} y_{2i+1} &= x_{2i+1} - \lfloor (x_{2i} + x_{2i+2})/2 \rfloor && \text{For } i = 0 \ldots \text{k-2} \\ y_{2i+1} &= x_{2i+1} - x_{2i} && \text{For i= k-1} \end{aligned}\right\} ,\ldots\ldots\ldots 2.9$$

(b) Determine the even coefficients

$$\left.\begin{aligned} y_{2i} &= x_{2i} + \lfloor y_{2i+1}/2 \rfloor && \text{For } i\text{= 0} \\ y_{2i} &= x_{2i} + \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor && \text{For } i\text{= 1…k-1} \end{aligned}\right\} ,\ldots\ldots\ldots 2.10$$

If the signal length N is odd (i.e. N = 2k+1), then the integer transform is computed in the following two steps:

(a) Determine the odd coefficients

$$y_{2i+1} = x_{2i+1} - \lfloor (x_{2i} + x_{2i+2})/2 \rfloor \quad \text{For } i = 0\ldots\text{k-1},\ldots\ldots\ldots 2.11$$

(b) Determine the even coefficients

$$\left.\begin{array}{ll} y_{2i} = x_{2i} + \lfloor y_{2i+1}/2 \rfloor & \text{For } i = 0 \\ y_{2i} = x_{2i} + \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & \text{For } i = 1 \ldots k-1 \\ y_{2i} = x_{2i} + \lfloor y_{2i-1}/2 \rfloor & \text{For } i = k \end{array}\right\} ,\ldots\ldots\ldots\ldots 2.12$$

2.6.4.2 One-Dimensional Inverse IWT

The inverse one-dimensional IWT is simply the inverse procedure from that applied when computing the forward one-dimensional IWT.

If the length N of the transformed signal y is even (i.e. N =2k), then the integer transform sequence x is computed in the following two steps: (a) Determine the even coefficients

$$\left.\begin{array}{ll} x_{2i} = y_{2i} - \lfloor y_{2i+1}/2 \rfloor & \text{For } i = 0 \\ x_{2i} = y_{2i} - \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & \text{For } i = 1 \ldots k-1 \end{array}\right\} ,\ldots\ldots\ldots\ldots 2.13$$

(b) Determine the odd coefficients

$$\left.\begin{array}{ll} x_{2i+1} = y_{2i+1} + \lfloor (y_{2i} + x_{2i+2})/2 \rfloor & \text{For } i = 0 \ldots k-2 \\ x_{2i+1} = y_{2i+1} + x_{2i} & \text{For } i = k-1 \end{array}\right\} ,\ldots\ldots\ldots\ldots 2.14$$

If the length N of transform signal y is odd (i.e. N=2k+1), then the inverse integer transform x is computed in the following two steps:

(a) Determine the even coefficients

$$\left.\begin{array}{ll} x_{2i} = y_{2i} - \lfloor y_{2i+1}/2 \rfloor & \text{For } i = 0 \\ x_{2i} = y_{2i} - \lfloor (y_{2i-1} + y_{2i+1})/4 \rfloor & \text{For } i = 1 \ldots k-1 \\ x_{2i} = y_{2i} - \lfloor y_{2i-1}/2 \rfloor & \text{For } i = k \end{array}\right\} ,\ldots\ldots\ldots\ldots 2.15$$

(b) Determine the odd coefficients

$$x_{2i+1} = y_{2i+1} + \lfloor (x_{2i} + x_{2i+2})/2 \rfloor \quad \text{For } i = 0 \ldots k-1, \ldots\ldots\ldots\ldots 2.16$$

2.7 Fidelity Measures [Sco98]

Fidelity measures can be divided into two classes:

a. Objective Fidelity Criteria.

b. Subjective Fidelity Criteria.

The objective fidelity criteria are borrowed from digital signal processing and information theory and provide us with equations that can be used to measure the amount of error in the reconstructed signal (image, sound or video).

Subjective fidelity criteria require the definition of a qualitative scale to asses signal quality. This scale can then be used by human test subjects to determine signal fidelity.

The commonly used objective measures are the Mean Square Error (MSE), Signal to Noise Ratio (SNR) and Peak Signal to Noise Ratio (PSNR).

The MSE is found by taking the summation of the square of the difference between the original and the reconstructed signal and finally divide it by the total number of samples as shown below:

$$MSE = \frac{1}{size}\sum_{i=0}^{size-1}(R_i - O_i)^2 \quad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.17$$

Where

R= Reconstructed signal.

O= Original signal.

Size= number of signal samples.

The smaller value of MSE mean the better the reconstructed signal represents the original signal.

The SNR metrics consider the reconstructed signal to be the "signal" and the error to be "noise". The SNR can be defined as:

$$SNR = 10\log_{10}\left(\frac{\frac{1}{Size}\sum_{i=0}^{Size-1} O_i^{\,2}}{MSE}\right) \quad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.18$$

Where

$Size$ = number of signal samples.

O = Original signal.

A large value of SNR implies a better reconstructed signal.

The PSNR metrics consider the "maximum peak value" and the error to be "noise". The PSNR can be defined as:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.19$$

PSNR is like SNR where the large value means a better-reconstructed signal represents the original signal.

Chapter Three
System Development

3.1 Introduction

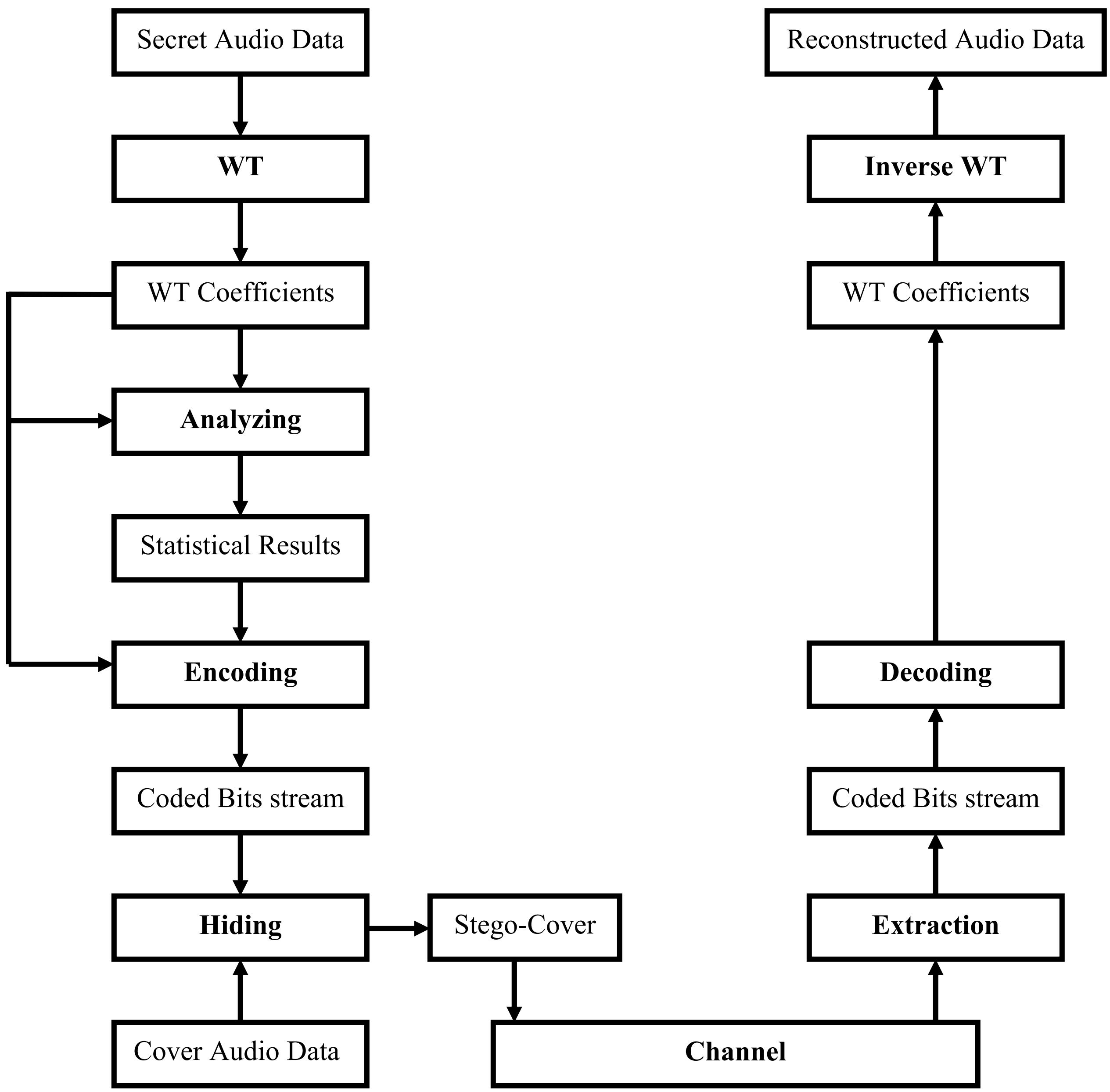
Multimedia data hiding techniques have been developed to be as strong basis for steganography area, its number of applications scenarios (like digital rights managements, covert communication, hiding executables for access control) was gradually extended, multimedia steganography techniques have to satisfy two basic requirements in order to make cover object and stego-cover object perceptually indiscernible. The first one is perceptual transparency and the other is high embedding capacity for additional data.

3.2 The Overall System Model

The block diagram of the proposed system is shown in figure (3.1), it can be broken into seven main parts as follows:

- **Input Audio Files (secret and cover).**
- **Analyzing.**
- **Encoding.**
- **Hiding.**
- **Extraction.**
- **Decoding.**
- **Inverse Wavelet Transform.**

Each part of the system will be discussed separately in the next subsections.



Figure(3.1) The Overall System Model

3.2.1 Input Audio Files

The secret audio signal and the cover audio signal are both considered files of type WAV (Windows Audio Visual). The WAV file starts out with a header followed by a sequence of data chunks. A WAV file often contains single “WAVE” chunk, which consists of two subchunks, a format (fmt), subchunk specify the file format and the (data), subchunk containing the actual data samples. A detailed description of the WAV file format is presented in appendix (A).

In the proposed system the secret and cover files are both considered of type (8 bits). The block diagram (figure 3.2)is the reading process of WAV file.

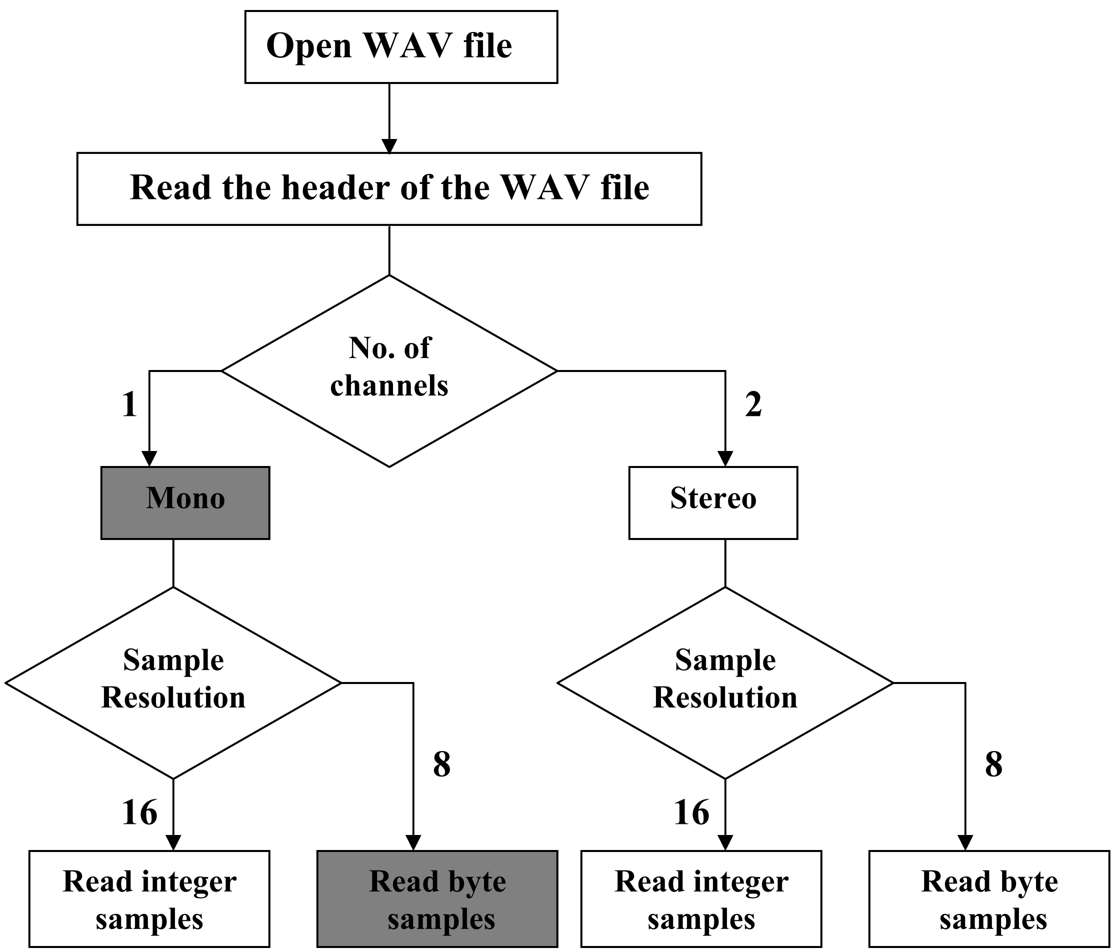


Figure 3.2 Reading Audio File

Note: The shaded cases were implemented in the proposed system

3.2.2 Analyzing

The proposed system uses three encoding methods they are (fixed encoding method, S-shift encoding method and hybrid encoding method), these methods will discussed in section (3.2.3). Because of the behavior of wavelet transform, the nature of its coefficients for each subband is different, so that different encoding methods are used to encode the coefficients of each subband. Analyzing step will determine:

1. The best number of wavelet transform passes.
2. The best encoding method for each subband.

The word "best" here means the case uses the minimum number of bits to encode the wavelet transform coefficients. Algorithm (3.1), presents the steps of the implemented "Analysis" unit. Algorithm (3.2) and (3.3) presents the implemented steps of forward Haar and Integer 3/5 wavelet transform, respectively.

Algorithm (3.1) Analyzing

```
INPUT
        X ( )    // Array of one dimension represent the secret wave data
        N        // number of wave samples
OUTPUT
        Best_Encoding_Methods   // Best method for each subband

      Select a WT algorithm (algorithm (3.2) or algorithm (3.3)).
      Set Min_No_of_Bits = N*8
      Set Total_Bits = 0
      Loop for No_of_Passes // No_of_Passes= Int(Log(N) / Log(2)) - 2
         Apply the selected WT algorithm
         Loop for each subband
                Call Find_Best_Encoder that output (No_of_Bits)// algorithm (3.4)
                Set Total_Bits =Total_Bits+No_of_Bits
        End Loop
```

To be Continue

```
    If Total_Bits < Min_No_of_Bits Then
        Set Min_No_of_Bits = Total_Bits
        Record the encoding methods as Best_Encoding_Methods
    End If
End Loop
```

Algorithm (3.2) Haar Wavelet Transform

INPUT

    X ( )   // Array of wave data
    N   // Number of wave samples

OUTPUT

    L ( )   // Low subband (approximation) coefficients
    H ( )   // High subband (detailed) coefficients

If N is even Then

Set K = N/2 -1

Loop For i = 0 … K

Set $L(i) = \left[X(2i) + X(2i+1)\right] / \sqrt{2}$

Set $H(i) = \left[X(2i) - X(2i+1)\right] / \sqrt{2}$

End Loop

Else

Set K = (N-2)/ 2

Loop For i = 0 … K

Set $L(i) = \left[X(2i) + X(2i+1)\right] / \sqrt{2}$

Set $H(i) = \left[X(2i) - X(2i+1)\right] / \sqrt{2}$

End Loop

Set $L(\frac{N+1}{2}) = X(N-1) * \sqrt{2}$

Set $H(\frac{N+1}{2}) = 0$

End If

Algorithm (3.3) Integer Wavelet Transform

```
INPUT
    X ( )          //array of wave data
    N              // Number of wave samples
OUTPUT
    L ( )          // Low subband (approximation) coefficients
    H ( )          // High subband (detailed) coefficients

    If N is even Then
        Set K=N /2
        Compute vector Y ( ) as follows
            Loop For I = 0 … K-2
                Set Y(2i) = X(2i+1) − (X(2i) + X(2i+2)) div2
            End Loop
            Set Y(2K−1) = X(2K−1) − X(2K−2)
            Set Y(0) = (X(0) + Y(0)) div2
            Loop For I = 1 … K-1
                Set Y(2i) = X(2i) + (Y(2i−1) + Y(2i+1)) div4
            End Loop
    Else
        Set K = (N-1) /2
        Compute vector Y ( ) as follows
            Loop For I = 0 … K-1
                Set Y(2i) = X(2i+1) − (X(2i) + X(2i+2)) div2
            End Loop
            Set Y(0) = X(0) + Y(0) div2
            Loop For i = 1 … K-1
                Set Y(2i) = X(2i) + (Y(2i−1) + Y(2i+1)) div4
            End Loop
            Set Y(2K) = X(2K) + Y(2K−1) div2
    End If
    Split the vector Y ( ) into L ( ) and H ( ), where L ( ) = Even Y's, and H ( ) = Odd
    Y's
```

Note: It is possible to reapply algorithms (3.2, 3.3) on the vector L only according to the number of wavelet transform passes, this kind of implementation is called tree scheme, or reapply them to both L and H, this kind of implementation is called packet scheme. Both schemes were implemented in the proposed system.

Algorithm (3.4) Find_Best_Encoder

INPUT

TX // A vector represent the WT coefficients

OUTPUT

No_of_Bits // bits required to encode all the transform coefficients

Index_Best_Encoding_Method

Encoding_information

Find Max // Maximum transform coefficient

Compute His () // histogram for the transform coefficients

Set $N = \lceil \log_2 (\text{Max}) \rceil$

Set $TB1 = (N + 1)\sum_{i=0}^{Max} His(i)$ // No. of bits for Fixed Encoding Method

Loop for B=1…N // is the number of coefficients in the entire subband

Set $U=2^B - 2$

Set $Ob = \lceil \log_2 (\text{Max - U}) \rceil$

Set $TB2 = B\sum_{i=0}^{U} His(i) + \sum_{i=U+1}^{Max} His(i)[round(i/U)B]$ // No. of bits for S-Shift

Set $TB3 = B\sum_{i=0}^{U} His(i) + Ob\sum_{i=U+1}^{Max} His(i)$ // No. of bits For Hybrid

End Loop

Set M = Min (TB1,TB2, TB3)

If M = TB1 then

Set Index_Best_Encoding_Method = Fixed Encoding Method

Set No_of_Bits = TB1; output N as Encoding_information

End If

If M = MTb2 then

Set Index_Best_Encoding_Method = S-Shift Encoding Method

Set No_of_Bits = MTb2, output recorded B as Encoding_information

End If

To be Continue

Else

Set Index_Best_Encoding_Method= Hybrid Encoding Method

Output B and Ob as Encoding_information

End If

3.2.3 Encoding

The second part of the proposed system is encoding, the purpose of this part is to encode the wavelet transform coefficients in a way that use as minimum as possible number of bits (i.e. to increase the hiding ratio), in addition to that the variety of encoding may provide additional security. To accomplish these purposes three encoding methods were used they are:

3.2.3.1 Fixed Length Encoding Method

It is the simplest encoding method, in this method all of the coefficients will be coded by using the same number of bits needed to encode the maximum one, so the total number of bits needed to encode the entire coefficients stream can be computed as follows:

$$TotalBits = (n+1)\sum_{i=1}^{N} X_i \quad ,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.1)$$

Where

n is the number of bits needed to encode the maximum coefficient.

N is the number of transform coefficients.

X is a vector represents the wavelet transform coefficients.

Since the input to the encoding step is wavelets transform coefficients, and these coefficients may be negative, so additional bit is added to represent the signs as shown in algorithm (3.5).

Algorithm (3.5) Fixed Encoding Method

```
INPUT
        X( )    //A vector represent the wavelet transform coefficients
        N       // number of WT coefficients.
OUTPUT
        Coded bit stream
Steps:
        Find max        // maximum wavelet transform coefficients
        Set  n = number of bits needed to encode max
        Loop For I =1…N
                If X(I) >= 0 then output 0 using 1 bit
                Else output 1 using 1 bit
                End If
                Output (abs (X(I))) using n bits
        End Loop
```

Note: if $\log_2$ (max) is not an integer number then round max to the smallest integer number greater than max that satisfy this condition.

3.2.3.2 S-Shift Encoding Method

The idea of this method is to encode the sequence of numbers by codewords whose bit length is less than the bit length required to represent the maximum wavelet transform coefficient value, when it is coded using the fixed encoding method. The coefficients whose values are large may splitted into a sequence of codewords, using the formula:

$$X = nw_m + w_r \;, \qquad (3.2)$$

Where:

X is the number to be coded.

n a number of the shift number multiples.

w_m is the maximum value which can be coded using single codeword.

w_r is the value of the last codeword used to encode X, as shown in algorithm(3.6).

Algorithm (3.6) S-Shift Encoding Method

```
INPUT
        X( )    // A vector represent the wavelet transform coefficients
        B       // number of bits needed to represent the shift number
OUTPUT
        Coded bit stream

        Set U = 2^B – 2
        Loop For I = 1…N
                If X(I)>0 then Output 0 using 1 bit
                Else Output 1 using 1 bit
                End If
                Set X(I) = abs(X(I))
                Loop while X(I)>=U
                        Output U
                        Set X(I)=X(I)-U
                End Loop
                Output X(I)
        End Loop
```

3.2.3.3 Hybrid Encoding Method

This method is based on both fixed length encoding and S-shift encoding. In this method the number greater than the shift number is not encoded as a multiple of shift number but using fixed length method. The coefficients whose values are large is coded as follows:

$X = w_m + (X - w_m)$,……………………………………………..(3.3)

Where:

X is the number to be coded.

w_m is the maximum value which can be coded using single codeword.(i.e., instead of X two values will be stored which are the two terms of the right hand side of equation(3.3)), as shown in algorithm (3.7)

Algorithm (3.7) Hybrid Encoding Method

```
INPUT
        X( )    // A vector represent the WT coefficients
        B       // number of bits needed to represent the shift number
OUTPUT
        Coded bit stream
Steps:
        Set U = 2^B – 2
        Loop For I = 1…N
                If X(I)>0 then Output (0) using 1 bit
                Else Output (1) using 1 bit
                Set X(I) = abs(X(I))
                If X(I) < U then Output X(I) using B bits
                Else Output X(I)-U using B bits
                End If
        End Loop
```

In addition to the previous three encoding methods, the overhead information is encoded using fixed number of bits, as follows:

- Secret data size, (secret file length): 32 bits.
- Secret file sampling rate: 2 bits.
- Encoding methods: 2 bits.
- Wavelet transforms type: 2 bits.
- Number of passes of the wavelet transform: 4 bits.
- Step factor (the distance between the samples, that used to hide secret data):5 bits.

3.2.4 Hiding

It is the main part of the system. In this part, the output of the encoding part (i.e., coded bits stream) is embedded in the cover audio data to produce the stego cover. The first step in the hiding process is to hide the overhead information. The overhead information are hided in a fixed way regardless of the other secret information. Three hiding methods are suggested to hide the secret encoded information, which are: Two Least Significant Bits Insertion with Recovery Technique (LSBRT), Hiding in Audible Parts (HAP) and LSB Insertion in WT Domain (LSBWTD).

3.2.4.1 Hiding the Overhead Information

The overhead information were hided in the first 47 bytes of the cover audio data using standard LSB insertion in the fifth bit with error reducing methods as shown in algorithm (3.8).

Algorithm (3.8) Hiding the Overhead Information

```
INPUT
        Ov( )   //a vector represent the overhead information
        C( )    //  a vector represent cover file data
OUTPUT
        C( )    // cover data after hiding the overhead information
Loop for I = 0 to 46
        Set X0 = C(I)
        Set X = X0 and 239
        If X(I) =1 then X = X OR 16
        Set E0 = abs(X0-X)
        Loop for J = 1 to 8
                Set Xn = (X0+J) mod 256
                Set Xn2 = Xn AND 239
                If X(I) = 1 then Xn2 = Xn2 OR 16
                E = abs (Xn – Xn2)
                If E0< E then
                        Set E0 = E
                        Set X = Xn2
                End If
        End Loop
        Loop for J = 1 to 8
                Set Xn =abs(X0-J) mod 256
                Set Xn2 = Xn AND 239
                If X(I) = 1 then Xn2 = Xn2 OR 16
                E = abs (Xn – Xn2)
                If E0< E then
                        Set E0 = E
                        Set X = Xn2
                End If
        End Loop
        C(I) = X
End Loop
```

3.2.4.2 Two Least Significant Bits Insertion with Recovery Technique (LSBRT)

The most common and simplest steganographic method is the least significant bit (LSB) insertion method that embeds message in the least significant bits of the host audio. For increasing the embedding capacity, two LSBs in each sample can be used to hide additional data. At the same time, the risk of making the embedded message statistically detectable increases and the perceptual transparency of stego cover decreases. Therefore, there is a limit for the number of bits of each sample of the cover audio that can be used to embed the message.

The proposed system recovers this risk using a two steps approach. Figure (3.3) illustrates the proposed embedding method. In the first step, algorithm embeds two LSBs using standard embedding method. In the second step of the algorithm a simple method is used to search for the level of audio closest to the original audio level.

Let:

a(n) be the original level of audio.

s(n) the level obtained by embedding two LSBs directly.

s'(n) the level of audio obtained by flipping the value of the third LSB of s(n).

The minimum error level must be s(n) or s'(n). Let e(n) be the difference between a(n) and s(n) and e'(n) be error between a(n) and s'(n). So if $e(n)<e'(n)$ then s(n) will replace a(n), otherwise s'(n) is selected. This method is called minimum error replacement (MER). This hiding algorithm is illustrated in algorithm(3.9).

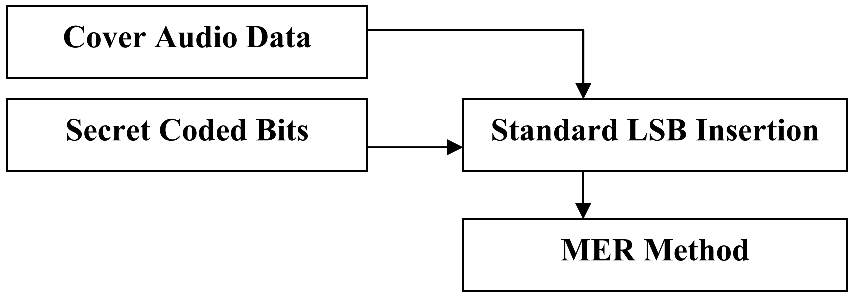


Figure (3.3) LSBRT Hiding Method

Algorithm (3.9) LSBRT Hiding Method

```
INPUT
        X( )    // Secret coded bits vector
        N       // Secret data size
        C( )    // Cover audio data vector
        Csiz    // Cover data size
OUTPUT
        Stego cover file

        Set Y = C
        Set I = 1; Set J = 47
        Set StpFctr = (Csiz*2) div N  // distance between used audio samples in
                                         hiding process
        Loop While I <= N
                Set Y(J) = Y(J) AND 252      // mask the two LSBs
                Set Y(J) = Y(J) OR X(I)
                If X(I+1) = 1 then   Set Y(J) = Y(J) OR 2
                Set e1 = abs(Y(J) – C(J))
                Set T = Y(J)
                Set T = T  AND 251   // mask the third bit
                Set e2 = abs(T – C(J))
                If e1 < e2 then  Output Y(J) to stego cover file
                Else   Output T to stego cover file
                Set I=I + 2; J=J+ StpFctr
        End Loop
```

3.2.4.3 Hiding in Audible Parts (HAP)

It is obvious that the Human Auditory System (HAS) is very sensitive to any change in the sound, but this sensitivity is not uniform.

In this method th WT coefficients with large values are selected to hold the secret data as shown in algorithm (3.10).

Algorithm (3.10) HAP Hiding Method

```
INPUT
        X( )    // Secret coded bits vector
        N       // secret data size
        C( )    // a vector represent the cover audio data
        Stp     // quantization step
OUTPUT
        Stego cover file

        Set I = 1; Set J = 47
        If N is odd then
                N=N+1
                X(N+1)=0
        End If
        Call integer WT for input C   // algorithm (3.3)
        Let TC be the transformed C  // A vector represent the WT coefficients
        Loop while I <= N
                T = abs(TC(j))
                S = sign (TC(j))
                If  (T) >= 3 then
                        V = X(I) +2 * X(I+1)
                        T = Stp* trunc(T/Stp)
                        T = T + V
                        TC(j) = T*S
                        J = J + 1
                        I = I + 2
                End If
        End Loop
        Call inverse integer WT for input TC  // algorithm (3.11)
        Output  the result of algorithm (3.11) in to stego cover file.
```

Algorithm (3.11) Inverse Integer Wavelet Transform

```
INPUT
        Y( )    // a vector represent the transform coefficients
        N       // transformed data size
OUTPUT
        X( )    // a vector represent the reconstructed data

        If N is even Then
                Set  K=N /2
                Set  X(0) = Y(0) − (Y(1) div 2)
                Loop For i = 1 … K-1
                        Set  X(2i) = Y(2i) − ((Y(2i−1) + Y(2i+1)) div 4)
                End Loop
                Loop For i = 0…K-2
                        Set  X(2i+1) = Y(2i+1) + ((Y(2i) + X(2i+1)) div 2)
                End Loop
                Set  X(2K−1) = Y(2K−1) + X(2K−2)
        Else
                Set  K = (N-1) /2
                Set  X(0) = Y(0) − (Y(1) div 2)
                Loop For i = 1…K-1
                        Set  X(2i) = Y(2i) − ((Y(2i−1) + Y(2i+1)) div 4)
                End Loop
                Set  X(2K) = Y(2K) − (Y(2K−1) div 2)
                Loop For i = 0 …K -1
                        X(2i+1) = Y(2i+1) + (X(2i) + X(2i+2) div 2)
                End Loop
        End If
```

3.2.4.4 LSB Insertion in WT Domain (LSBWTD)

This method is based on using the time domain algorithm (standard LSB insertion) but in WT domain. Since the integer WT is a lossless transform, so its coefficients can be used to carry the secret data as illustrated in algorithm (3.12).

Algorithm (3.12) LSBWTD Hiding Method

```
INPUT
        X( )     // a vector represent the secret coded bits
        N        // secret data size
        C( )     // a vector represent the cover file data
        Csiz     // Cover data size
OUTPUT
        Stego cover file

        Set StpFctr = (Csiz) div N      // distance between used audio samples in hiding
                                           process
        Call integer WT for the input C // algorithm (3.3)
        Let TC be the transform coefficients
        Set I = 47; J=1
        Loop  while  I <= N
                Set S = sign of TC(I)
                TC(I) = abs (TC(I)) and 254
                TC(I) = TC(I) OR X(J)
                TC(I) = S * TC(I)
                Set I = I+ StpFctr; J=J+1
        End Loop
        Call inverse integer WT for input TC //  algorithm (3.10)
        Output the result of algorithm (3.10) into stego cover
```

3.2.5 Extraction

It is the reverse process of hiding. The first step is to extract the overhead information, these information are necessary to the extraction and decoding steps. The overhead information are simply extracted from the first 47 bits of the stego cover file as shown in algorithm (3.13). In the proposed system the extraction of the other secret information is depends on which hiding method was used. The hiding method is identified by using a password (stego key) that will converted into a number using a hash function, this number represent a position in the cover file where the two bits will be hided. Algorithm(3.14) shows the process of computing the position in the cover file.

Algorithm(3.13) Overhead Information Extraction

```
INPUT
        Sc( )  // a vector represents the stego cover file information
OUTPUT
        Ov( )  // a vector represents the overhead information

        Loop for I = 0 to 46
                If (Sc(I) AND 16) >= 1 then  Ov(I) =  1
                Else  Ov(I) = 0
        End Loop
```

Note: the two bits represent the hiding method are hided and extracted by the same method used to hide the overhead information.

Algorithm (3.14) compute position in file

```
INPUT
        P       // a string represent the password
        S       // cover file size
OUTPUT
        Pos     // Position in the cover file

        Set Acm = 0
        Loop For I = 1 … length(P)
                Acm = Acm + ASCII(P(I))/255
                Acm = Acm/0.0523
        End Loop
        Acm = Acm – Int(Acm)
        Pos = trunc(Acm *S)
```

In the proposed system, there are three extraction algorithms each for a embedding method, these are algorithms (3.15, 3.16, and 3.17). Note that the system avoids hiding in and extract from the computed position in algorithm (3.14).

Algorithm(3.15) Extract Two LSBs

```
INPUT
        X       // a vector represent a stego cover file data
        N       // stego cover data size
        StpFctr         // distance between used audio samples in hiding process

OUTPUT
        B       // a vector represent the secret coded bits
Set  I = 1 ; Set  J = 47
Loop While I <= N
        Set B(I) = X(J) AND 1
        If (X(J) AND 2) = 2 then   Set B(I+1) = 1
        Else     Set B(I+1) = 0
        Set I = I + 2;    J = J + StpFctr
End Loop
```

Algorithm(3.16) Extract HAP

```
INPUT
        X       // a vector represent the stego cover data
        N       // stego cover data size
        Stp     // quantization step
OUTPUT
        B       // a vector represent the secret coded bits
Set  I = 1;  Set  J = 1
Loop While I <= N
        If X(J) >= 3 then
                T = Stp * trunc(X(J)/Stp)
                V = X(J) – T
                B(I) = V and 1
                B(I+1) = V >> 1
                I = I + 2
        End If
        J = J + 1
End Loop
```

Algorithm(3.17) Extract LSB from WT Domain

```
INPUT
        X       // a vector represent a stego cover file data
        N       // stego cover data size
        StpFctr         // distance between used audio samples in hiding process
OUTPUT
        B       // a vector represent the secret coded bits
Let TX be the output of applying algorithm(3.3) on the input X
Set I = 47; J = 1
Loop while I <= N
        Set B(J) = abs(TX(I)) AND 1
        Set I = I + StpFctr; J=J+1
End Loop
```

3.2.6 Decoding

Given the extracted coded bits, the first job to be done is to get the overhead information required to reconstruct the hidden information. The coefficients of each WT subband are decoded according to the used encoding method. For each encoding algorithm there is a corresponding decoding algorithm which are illustrated by algorithms (3.18 – 3.20).

Algorithm(3.18) Fixed Length Decoding

```
INPUT
          B        // a vector represent the coded bits stream
          Bt       // bits length for each coefficients
OUTPUT
          X        // a vector represent the WT coefficients

          Let I be the start index of the WT subband
          Let N be the end index of the WT subband
          Set J = 1
          Loop While I <= N
                   If B(J) = 1 then        Set  S = -1
                   Else     Set S = 1
                   Loop For K = 1 … Bt
                            T(K) = B(J+K)
                   End Loop
                   Set X(I) = S * Convert to decimal (T())
                   Set I = I + 1;    Set J = J + Bt
          End Loop
```

Algorithm(3.19) S-Shift Decoding

```
INPUT
        B       // a vector represent the coded bits stream
        Bt      // bits length for each coefficients
OUTPUT
        X       // a vector represent the WT coefficients


Let  I be the start index of the WT subband
Let  N be the end index of the WT subband
Set J = 1
Set U = 2^Bt – 1
Loop While I <= N
        If B(J) = 1 then        Set  S = -1
        Else    Set  S = 1
        Set C = 0
        Loop While C >= 0
                Loop For K = 1 … Bt
                        Set  T(K) = B(J+K)    // T is a temporary vector
                End Loop
                Set V = convert to decimal (T())
                Set C = C + V;  Set J = J + Bt
        End Loop
        Loop For K = 1 … Bt
                Set  T(K) = B(J+K)
        End Loop
        Set R = convert to decimal (T)
        Set  C = C + R
        Set  X(I) = S * C
        Set  I = I + 1
End Loop
```

Algorithm(3.20) Hybrid Decoding

```
INPUT
        B       // a vector represent the coded bits stream
        Bt      // bits length for each coefficients
        Ob      // bits length to represent Max – Shift number
OUTPUT
        X       // a vector represent the WT coefficients

Let  I be the start index of the WT subband
Let  N be the end index of the WT subband
Set  J = 1;  Set U = 2^Bt – 1
Loop While I <= N
        If B(J) = 1 then        Set S = -1
        Else    Set S = 1
        Loop For K = 1 … Bt
                Set  T(K) = B(J+K)
        End Loop
        Set  V = convert to decimal (T)
        Set J = J + Bt
        Loop For K = 1 … Ob
                Set  T(K) = B(J+K)
        End Loop
        Set  R = convert to decimal (T)
        Set  X(I) = S * (V + R)
        Set  I = I + 1
End Loop
```

3.2.6 Inverse Wavelet Transform

The output of the decoding step is a WT coefficients, so the final step to reconstruct the secret data is the applying of the inverse WT. In the system the user can select one of the two WT types (Integer WT or Haar WT), and its type index is recorded as an overhead information and according to that the system have two algorithms for inverse WT which are algorithm (3.11) for integer type and algorithm (3.21) for Haar type.

Algorithm(3.21) Haar inverse WT

```
INPUT
        L       // a vector represent the low frequency coefficients
        H       // a vector represent the high frequency coefficients
        N       // transformed data size
OUTPUT
        X( )    // a vector represent the original data
If N is even Then
        Set  K=N /2
        Loop For I = 0… K
                Set X(2I) = (L(I) + H(I)) / √2
                Set X(2I) = (L(I) − H(I)) / √2
        End Loop
Else
        Set K=N /2 -1
        Loop For I = 0… K
                Set X(2I) = (L(I) + H(I)) / √2
                Set X(2I) = (L(I) − H(I)) / √2
        End Loop
        Set X(N − 1) = L(N − 1) / √2
End If
```

Conclusions and Future Work

5.1 Introduction

In this chapter, a list of remarks derived from the investigation of the test results shown in chapter four will presented. Also, some suggestions for a future work, that may enhance the system efficiency, are presented.

5.2 Conclusions

From the tests results conducted on the proposed system, the following remarks were derived:

1. The use of Wavelet Transform (WT) add some power to the encoding reslts, because it divides the audio signal into different wavelet parts (i.e., subbands) according to the number of WT passes, each part can be processed independently so that good compression can be gained.
2. The Integer Wavelet Transform (IWT) is better than the Haar Wavelet Transform (HWT) in both encoding and hiding processes because:
   a) The IWT produces more compact coefficients than HWT, so it can provide high compression ratio.
   b) IWT is error free transform, so it is better for lossless compression. For the same reason IWT is better for hiding since no additional hidden data will be lost in transformation process.
3. The Fixed Encoding Method (FEM) is the worse among the three encoding methods, since it was not adapted by the system selector to encode any subband.
4. The use of speech audio cover is better than music audio cover, since the first causes greatest MPSNR, MSNR and the smallest MMSE. As another reason is that most of the expert musicians can detect the noise

caused by the hiding process just by listening to the Stego_cover file, keep in mind that some of the music files themselves are standard like famous concertos.

5.3 Future Work

During the development of the proposed system, many suggestions for future work was emerged to increase the system efficiency, among these suggestions are the following:

1. Using more encoding methods in addition to these used in the proposed system may lead to gain more compression performance.
2. Develop a system for hiding audio in image or image in image using the same techniques used in the proposed system.
3. Develop a system that uses other hiding methods like (phase coding, spread spectrum and echo data hiding) techniques.
4. Develop a system to use other audio file formats like (MP3 and ADPCM wav file).

Appendix A
The WAV File Format

File offset (bytes)	Field name	Field size (bytes)	
0	ChunkID	4	The "RIFF" chunk descriptor The format of concern here is WAV, which requires two subchunks "fmt" and "data"
4	ChunkSize	4	
8	Format	4	
12	Subchunk1ID	4	The "fmt" subchunk Describes the format of the sound information in the data subchunk
16	Subchunk1size	4	
20	Audioformat	2	
22	NumChannels	2	
24	Samplerate	4	
28	Byterate	4	
32	Blockalign	2	
34	BitsperSample	2	
36	Subchunk2ID	4	The data subchunk Indicates the size of the sound information and contains the raw sound data
40	Subchunk2size	4	
44	Data	Subchunk 2size	

The WAV format starts with the RIFF header:

Offset	Size	Name	Description
0	4	**chunkID**	Contains the letters "RIFF" in ASCII form
4	4	**Chunksize**	This is the size of the entire file in bytes minus 8 bytes for the two fields not included in this count: chunkID and chunksize
8	4	**Format**	Contains the letters "WAVE"

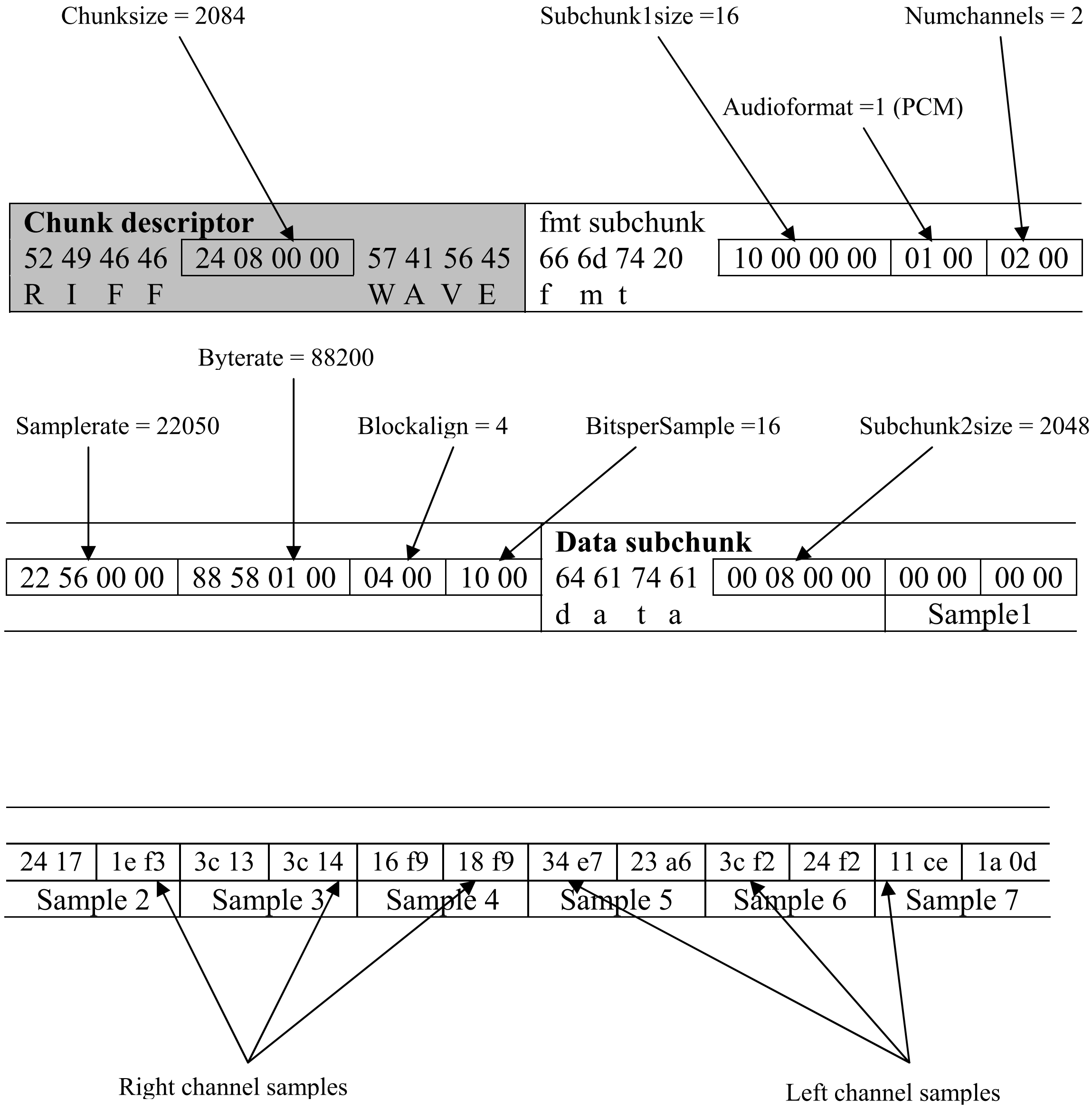
The WAV format consists of two subchunks: "fmt" and "data":
The "fmt" subchunk describes the sound data format:

Offset	Size	Name	Description
12	4	Subchunk1ID	Contains the letters "fmt"
16	4	Subchunk1size	16 for PCM. This is the size of the rest of the subchunk which follows this number.
20	2	Audioformat	PCM = 1 (i.e., linear quantization) values other than 1 indicates some form of compression.
22	2	Numchannels	Mono = 1, Stereo = 2
24	4	Samplerate	8000, 44100, etc.
28	4	Byterate	= Samplerate * Numchannels * BitsperSample/8
32	2	Blockalign	= Numchannels * BitsperSample/8 The number of bytes for one sample including all channels
34	2	BitsperSample	8 bits = 8, 16 bits = 16

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

36	4	Subchunk2ID	Contains the letters "data"
40	4	Subchunk2size	= Numsamples * Numchannels * BitsperSample/8 This is the number of bytes in data
44	*	Data	The actual sound data

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



References

- [Aud03] Auday A. Al-Dulaimy, “Fractal Image Compression with Fasting Approach”, Msc. Thesis, Dept. Of computer science, Al-Nahrain University, Iraq, 2003.
- [Ben96] W. Bender, D. Cruhl, N. Morimoto and A. Lu, “Techniques for Data Hiding”, IBM systems journal, Vol. 36, Nos. 3&4, 1996.
- [Bur98] C. Burrus, R. Gopinath and H. Gao, “Introduction to wavelets and wavelet transform”, Prentice Hall Inc., 1998.
- [Dav00] P. Davern and M. Scott, “Steganography: it is history and it is application to computer based data files”, School of computer applications, 2000.
- [Deb03] Deborah A. Whitiak, “The art of steganography”, GSEC practical, SANS institute, 2003.
- [Dec99] L. Decamro, “New technologies for audio copy protection”, Electronic Media Article, 1999.
- [Eri97] Eric E. Majani, “Algorithm for lossy/progressive/lossless image compression”, National Astronautics and Space Administration (NASA), NASA Tech Brief Vol. 21, N0. 12, Item #146, Dec. 1997.
- [Fer98] Ferrill, Elizabeth, Moyer and Mathew, “A survey of digital watermarking”, 1998.
- [Ibr04] Ibraheem N. Ibraheem, “Image compression using wavelet transform”, Msc. Thesis, Dept. of computer science, Baghdad University, Iraq, 2004.

- [Ira00] Ira S. Moskowitz, Garth E. Longdon and Liwu Chang, "A new paradigm hidden in steganography", Naval research laboratories, Washington, DC 20375, 2000.
- [Jia03] X. Jianyun, Andrew H. Sung, Peipei Shi and Qingzhong Lin, "Text steganography using wavelet transform ", Dept. of computer science, New Maxico Tech, Socorro, NM87801, USA, 2003.
- [Jam00] James Cadwell, "Steganography", US Air Force, 2000.
- [Kat00] Katezbeisser, Stefan Petitcolas and /A. Fabian, "Information hiding techniques for steganography and digital watermarking", Artech House Inc., Norwood, 2000.
- [Kie98] Kientzle Tim, "Programmers Guide to sound", Addison Wesley Developers Press, 1998.
- [Mik01] Mike Amundsen, "Tech yourself Visual Basic programming in 21 days", Design Synergy Corporation, USA, 2001.
- [Mus01] Mustafa Dhiaa T. Al-Hassani, "Design of a Fingerprint Recognition System Using Wavelet Transform", Msc. Thesis, Dept. of computer science, Al-Nahrain University, Iraq, 2003.
- [Pol98] R. Polikar, "The wavelet tutorial part 3: Multiresolution analysis", Dept. of electrical and computer engineering, Rowan University, Oct. 1998.
- [Qu96] Qu Shaohong, "Beyond the basics: real time audio and video", Prentice Hall Inc., 1996.
- [Sal98] Salomon D., "Data Compression the Complete Reference", Springer-Verlag, 1998.

- [Sco98] Scott E. Umborgh, “Computer Vision and Image Processing”, Prentice Hall PTR, Upper Saddle River, 1998.
- [Sel99] Sellars Duncan, “Introduction to steganography”, 1999.
- [Ste99] C. Stefano and F. Stefano, “Steganography and Copyright Marking”, Dept. of computer science, University of Bologna, Italy, 1999.
- [Tib03] Tiba Z. A., “Advanced compression techniques for multimedia applications”, Msc. Thesis, Dept. of computer science, Al-Nahrain University, Iraq, 2003.
- [Val99] C. Valens, “A really friendly guide to wavelets”, 1999.
- [Xin99] Xin Li and Hang H. Lu, “Transparent and Robust Audio Data Hiding in Subband Domain ”, Dept. of Electrical Engineering, Prinzeton University, Prinzeton, 1999
- [Yas03] Yasmeen I. Dieab, "Audio Watermarking", Msc. Thesis, Dept. Of computer science, Al-Nahrain University, Iraq, 2003.
- [Yan01] Yang Yang and faculty of computer science, Dalhousie University, “Digital watermarking technologies”, 2001.

ا لاصة

الكتابة ا فية (Steganography) هو فن اخفاء ا علومات بطرق معينة يث يصعب اكتشافها. الرسالة ا شفرة تؤدي ا الشك بينما الرسالة الغ مرئية لا تؤدي ا الشك. فن الكتابة ا فية الرقمية تستخدم رسالة او بيانات معينة تعرف با اوية (Container) او الغطاء (Cover) لاخفاء بيانات او رسائل اخرى تسمى بالسرية (Secret) داخله.

النظام ا قترح هذه الاطروحة هو نظام اخفاء صوت داخل صوت. النظام ا قترح البيانات السرية يتم ويلها اولا باستخدام التحويل ا و ي (Wavelet Transform) و ا عاملات النا ة يتم ترميزها باستخدام احدى طرق الترميز الثلاثة (طريقة ترميز الطول الثابت، طريقة الترميز بالتزحيف وطريقة الترميز ا جينة) ا نفذة النظام ا قترح.

ا رحلة التالية النظام ا قترح هي مرحلة الاخفاء يث يتم اخفاء رجات مرحلة الترميز داخل بيانات الغطاء.

هناك ثلاث طرق للاخفاء تنفيذها النظام ا عتمد (الاخفاء بالبت الاقل تأث ا، الاخفاء بالبت الاقل تاث ا مع تقنية التحس والاخفاء الاجزاء ا سموعة).

النظام ا قترح اختباره باستخدام ثلاث مقاييس معولية قياسية هي متوسط ا طأ ا ربع (MSE) ونسبة الاشارة ا الضوضاء (MSE) ونسبة الاشارة العليا ا الضوضاء (PSNR).

كما اشتقاق مقاييس معولية جديدة بادخال تعديلات على ا قاييس ا عولية القياسية هي متوسط ا طأ ا ربع ا عدل (MMSE) ونسبة الاشارة ا الضوضاء ا عدلة (MMSE) ونسبة الاشارة العليا ا الضوضاء ا عدلة (MPSNR) لتكون مناسبة اكثر لاختبارات الصوت.

كل ا قاييس ا عولية ال استخدمت اختبار النظام ا قترح اظهرت قيم جيدة لـ(PSNR) والنسخة ا عدلة منه حيث كانت (50 dB) كمعدل.

اما البيانات ا سترجعة فكانت بالضبط هي نفسها البيانات السرية ال اخفاءها اذا كان التحويل ا و ي الصحيح (Integer Wavelet Transform) هو ا ستخدم مرحلة الترميز بينما يظهر بعض الاختلاف الغ سوس اذا كان نظام التحويل ا و ي هار (Haar Wavelet Transform) هو ا ستخدم خلال مرحلة الترميز.

الجمهورية العراقية
وزارة التعليم العالي و البحث العلمي
جامعة النهرين

اخفاء الصوت ضمن مجال التحويل المويجي

رسالة
مقدمة الى قسم علوم الحاسبات في جامعة النهرين
كجزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل
منتصر جابر جواد
(بكالوريوس جامعة النهرين ٢٠٠٢)

أشراف

د. لؤي أدور جورج | د. بان نديم الكلاك

كانون الثاني ٢٠٠٥ | ذي الحجة ١٤٢٥